Foundations of Physical Computing Teacher Guide

Foundations of Physical Computing

Introduction

Computer science is very important to the success of our students in a digital, interconnected world. Computer science is a discrete academic discipline, but it is also embedded in virtually every area of academic study, since computer literacy, data analysis, and digital citizenship have become essential skills in most disciplines. Students need to be met where they are developmentally and this course allows students to enter and be successful, regardless of any prior computer science knowledge. Students who have had the opportunity to practice computational thinking, block coding, or other online computer course work, will find that they can grow with the open-ended challenges as well as the ability to write code more than one way.

This course covers computer science skills while it applies science, mathematics, and language arts. The lessons integrate the subjects seamlessly and include unplugged activities, 21st Century skills, and a focus on careers. Foundations of Physical Computing offers a progression for students to find instant success then be pushed to their maximum potential in open-ended challenges in every unit. Reflecting on their skills, students create a plan for entering into a career field after they have explored all 16 career pathways. Ending the course is a culminating experience which includes researching and solving a real-world problem and then presenting the solution to the class.

In this **computer science progression**, students will:

- Learn about computer science topics while applying science, language arts, math and engineering concepts
- Develop computational thinking skills
- Solve computational problems using prototypes
- Critically engage in discussions on computer science topics
- Decompose problems and prioritize criteria for solutions
- Better understand the role of computing in the world around them
- Learn, perform, and express themselves in multiple subjects and interests
- Experience real world issues and solve real problems

In the **careers lessons**, students will

- Explore all 16 Career pathways
- Complete a career connection in every unit
- Determine the link between skills and lessons to careers and career pathways
- Compare career fields and jobs
- Determine skills needed in most careers

- Determine careers that impact a given situation in a culminating career activity
- Self-assess their interests, skills, and goals
- Compare their interests, skills, and goals at the start to those at course completion
- Reflect and create a career plan

Learning Promise

By the end of this unit, students create artifacts and build models that use motors, sensors, wheels, lights, and sounds. They utilize various programming techniques including writing pseudocode and comments, using conditional statements, loops, Boolean expressions, and compound conditionals as well as linear and computational thinking and debugging. Students apply what they learn to various guided and open-ended projects which culminate in a presentation of a solution to a real-world problem in one of the career pathways. Students also explore all 16 Career pathways, reflect on the skills required, reflect on their own skills and interests, complete a culminating career activity and create individualized plans for success in a career.

Knowledge and Skills Applied

This course is built around the K-12 Framework and the CSTA standards. A matrix of the lessons and standards, framework areas covered is available at the end of this document. Students will start with the basics and have experiences that reinforce and apply skills that they are learning or may have learned previously.

At the end of this unit, students will know or be able to do the following:

- Design, iteratively develop and program a prototype of a robot to solve a problem
- Work collaboratively, give and receive feedback, and incorporate suggestions
- Debug and troubleshoot both hardware and software problems
- Use algorithms, data, compound conditionals, sensors, loops and Boolean logic
- Document programs, feedback, testing, and debugging
- Articulate flowcharts, pseudocode, and comments to address complex problems
- Decompose problems and subproblems into parts
- Discuss issues of bias and accessibility
- Communicate the solution to a problem, including model and programming
- Collect and graph information for analysis
- Analyze data and apply the information to improve a solution
- Determine the societal effects of a given solution
- Kinesthetically work through examples of how computers and robots work together

Lesson Outline (Lesson times vary; most lessons are 45 or 90 minutes)

Unit 1 What is a Robot?

- Intro to Robotics
- What is a Robot?
- Sound and Light
- Motors and Sensors
- Make it Move
- Connecting to Careers – Human Services and Education & Training

Unit 2 Improving a Design

- Iteration and Perseverance
- Pseudocode
- Create a New Product
- Share Your Ideas
- Mini-Challenge: Creativity in Business
- Connections to Careers: Marketing and Arts, A/V Technology & Communications

Unit 3 Troubleshooting and Debugging

- Model Debugging
- Software Debugging
- Dancer Break Down
- Dance to the Color
- Mini-Challenge: Design a Route
- Connections to Careers: Information Technology and Law, Public Safety, Corrections & Security

Unit 4 Testing and Evaluating Solutions

- Testing Prototypes
- Human vs. Robot
- Comparing Robotic Grabbers
- Repetitive Tasks
- Turns, Speed, and Accuracy
- Uphill Climb
- Mini-Challenge: Design for Someone
- Connecting to Careers: Health Science and Agriculture, Food, & Natural Resources

Unit 5 Sensors

- Sensors Trigger Reactions
- Sensors and Data
- Dance to Debug
- Maze
- Factory Robot
- Parking Lot
- Mini-Challenge: Parking Lot
- Connecting to Careers: Manufacturing and Transportation, Distribution & Logistics

Unit 6 Variables

- Drone Pitch and Roll
- Drone Movements
- Variables
- Graphing, Speed, and Distance
- Mini-Challenge: Distance Game
- Connecting to Careers: Finance and Business Management

Unit 7 Arrays, Boolean Expressions, and Conditionals

- Intro to Arrays (Lists) and Conditionals
- Comparing Arrays
- Conditionals and Simplifying Code
- Conditionals and Boolean Expressions
- Mini-Challenge: Security Alarm Using Operators
- Connecting to Careers: Hospitality, Tourism & Government

Unit 8 Compound Conditionals

- Game with Variables
- Compound Conditionals
- Compounding Conditionals
- Mini-Challenge: Break Out Room
- Connecting to Careers Architecture & Construction and Science, Technology, Engineering and Mathematics

Unit 9 – Careers Culminating Project

- Culminating Activity: Careers
- My Careers Interests
- Culminating Activity: My Career Reflection and Plan

Unit 10 – Physical Computing Culminating Project

- Culminating Activity in the areas of Natural Resources, Transportation, Health Science and STEM, Manufacturing, Government and Public Administration and Law, Public Safety, Corrections & Security, and Agriculture

Teacher Support for Students

For students to be successful in this unit, the teacher may need to provide support in one or more areas - research, evaluation, identification and selection of problems and solutions, and time management. Teachers may need to provide additional scaffolding for some students with special needs or those who do not have the prerequisite skills in areas such reading, math, social and emotional skills. We suggest teachers have professional development on programming with SPIKE Prime and some experience with building LEGO® models prior to starting this course.

Parts should not be shared between sets. Every set contains one bag of spare parts. We recommend you collect those when the sets are opened and use as needed. Tell students when they do inventory to let you know if a part is missing. You may or may not have a spare part to cover the loss. If students take inventory of one segment of a tray daily, parts are rarely lost. Students will make a journal entry daily about the materials management of their sets.

Throughout the process, the teacher should be supporting students with guidance, rather than provide answers. We recommend you do not touch their models nor their programming, but rather ask questions and suggest students work together in their teams or across teams.

We suggest the following for helping students find success throughout the unit:

- Provide teams with recommended websites for research into the topics

- Limit the time for brainstorming ideas and research
    - Give students a definite amount of time they can use
    - Teams that need a push can be given time checks with progress points

- Provide strategies brainstorming techniques
    - Mind mapping
    - Slip writing
    - Round robin

- Provide strategies for narrowing topics
    - Write topics in one sentence
    - Write practical solutions for each topic in one sentence
    - Rank the topics
    - Discuss possible ethical ramifications for topics/solutions

- Provide teams with ideas for collaboration
    - Use pair programming
    - Take turns building, building subcomponents
    - Take turns programming, programming subcomponents
    - Division of tasks – all members involved
    - Students will make a journal entry about their teamwork daily.

- Discussion with equity for all members
    - Everyone gets a turn to talk
    - Pros and cons for each person's ideas
    - Have a team meeting about all current issues
    - Give everyone a chance to think and respond

- Provide teams with ideas for problem-solving and debugging
    - Creating pseudocode before programming
    - Tracing their code
    - Checking each subcomponent
    - Asking all team members for ideas
    - Asking all team members to be part of discussions on issues
    - Asking another team to help troubleshoot
    - Discuss pair programming and roles

    (A set of tips for debugging is in the student guide.)

- Walking through the construction of a model as part of the solution
    - Checking to see if the model within the solution has a programming or hardware issue
    - Removing only the section of the model that isn't working
    - Asking all team members to look at the issue and give feedback
    - Asking another team to help troubleshoot

Support for Teachers

LEGO Education also provides teacher support through online professional development. SPIKE™ Prime Classroom - PD | LEGO® Education

Teachers can learn about STEAM Concepts, 21st Century Skills, Pedagogy, and Classroom Management in short 10-minute Bursts or longer 30-45 minute Quests. These are video-based lessons with teachers in real classrooms, both face-to-face and virtual.

Additionally, teachers can learn from Global Master Trainers and Certified LEGO Education Trainers how to prepare the SPIKE Prime sets and start programming.

Another support is the LEGO Education Community. Welcome to the LEGO® Education Community - LEGO Education The community is group of educators from around world that share their ideas, experiences, and lessons. The community is a great place to ask a question and get responses from other teachers working in the classroom with LEGO Education products. We encourage you to become part of the community and share your ideas and experiences.

Schools and Districts can also purchase face-to-face or virtual training and coaching. This allows for ongoing support starting before the class begins and continuing throughout the school year.

Culminating Project

There are multiple challenges from which students choose only one for their culminating project. They can work in groups of two students or in a small group of two teams (four students). The rubric for the Culminating Project is provided.

The Culminating Project consists of four phases, which allow for teacher review and support and peer review and support. It provides a smooth transition from lessons to application and completion of a solution.

At the conclusion of the Culminating Project, students will reflect and complete a self-assessment and a team assessment. These are provided at the end of the unit.

Students should take apart the models and put the elements into the correct compartments. Then, each team should take a **complete inventory** of the set.

Culminating Project – The Phases

Challenge Phase 1

Teacher check and approval as soon as team is ready.

- Research several topics
- Identify two or three possible problems to solve within the challenge
- Select one problem to solve within the challenge
- Brainstorm solutions for the problem chosen
- Consider human/computer interactions – impact on society – ethical; physical haptics/design of their proposed solution
- Determine the solution is doable within time frame allowed
- Write a clear statement of the problem, proposed idea for a solution, and the impact on society in their journals.
- **Teacher check and approval prior to going forward**

Challenge Phase 2

Peer review should be done as soon as a team is ready for it.

- Model plan including where using sensors and motors
- Model plan including where to use subcomponents
- Model plan including how and when to use sounds and lights
- Sketches of where to use components and overall design
- Description of solution and model concept
- Pseudocode for the solution
- **Peer review - based on rubric**
- **Teacher review**

Note: Students should write feedback received into their journals as well as take notes in their journals on feedback they gave.

Challenge Phase 3

Create the model and the program.

While working on solution, the teacher should ask to see:

- Previous/similar design/lessons where they have used
- Models and initial programming
- Pseudocode for possible program (determine software/hardware interactivity)
- Considerations of human/computer interactions – impact on society – ethical; physical haptics design
- Iterations of design and programing

Challenge Phase 4

Presentations & Assessment

Presentation of Completed Solution to the Class

- Present completed solution to class
- Teacher and team check against rubric
- Peer review
- Team self-assessment
- Individual self-assessment
- Clean up including all models taken apart and a full inventory of all sets

Culminating Project – The Challenges

Challenge 1

Natural Resources - Environmental Clean Up

Students research problems in the environment such as, but not limited to, recycling, environmental cleanup, or reducing the use of limited resources. Students choose an issue and a location for an environmental problem to solve. They create a model and present how it works to solve the problem. Students determine how well their solution works and the social impact of their solution.

We want each student to choose his/her own specific challenge and the solution. However, if students are really stuck, you may use this example:

> You are scientists and engineers who form a team to take white plastic items from the ocean. The size containers you target are soda and water bottles and plastic bags. When the sensor sees something white in the range of a grabber, the model makes a sound to alert the operator. The lights on the hub keep a count of the materials that have been captured.

Challenge 2

Transportation

Students choose A, B, C, or D in this area.

A. Transportation – Delivering to Specific Locations

Students research problems of transportation in a specific location – could be inside a building or outside in a limited area. Students choose an issue and a location for their transportation problem to solve. They create a model and present how it works to solve the problem. Students determine how well their solution works and the social impact of their solution.

B. Transportation of Materials On the Moon

Students are part of a team devising new transportation vehicles for moving materials on the moon. The vehicle will need to be able to move in, out, and around craters, be seen from towers, and carry a load of either people or materials. Students determine how well their solution works and the environmental impact of their solution.

C. Transportation of Materials In the Ocean

The ocean, a place for exploration. The students work for a company that has won the bid to create a vehicle that can handle the pressure of the depths, go into shark infested waters, or navigate sensitive ecosystems. The vehicle will also have to grab items to bring to the surface for study, show video real-time, and not bump into anything, especially coral reefs, other equipment in the ocean, or the bottom of the ocean. Students determine how well their solution works and the environmental impact of their solution.

D. Autonomous Transportation of Humans on Earth

Students have landed a job on the autonomous vehicle team! This team is creating an autonomous vehicle that can safely carry people from place to place, obeying all traffic laws, parking safely, and letting people know when the battery needs to be charged. Remember you must discuss and take into consideration people getting in and out of the vehicle, decisions that could involve evaluating choices for crashes, and visibility on the road. Students determine how well their solution works and the environmental impact of their solution.

We want each student to choose his/her own specific challenge and the solution. However, if students are really stuck, you may use this example:

> You are engineers and planners for a hospital. You must ensure that the autonomous vehicles deliver the specific materials to the correct locations. Using sensors on a wheeled vehicle, the model must take vials of blood to the laboratory, x-rays to reviewing doctors, escort visually impaired persons to the surgery center, and family members to the gift shop. Your machine needs to transport all materials safely, not run into objects or people, and show the location on the brick of where it is headed.

Challenge 3

Health Science and STEM: Game for Mental Agility

Students research problems of mental agility by helping people keep the brain active and learning new things. Students choose ways to help people stay mentally agile and create a game that incorporates multiple tasks. The game can be any type, but must meet the requirements in the rubric. They create a model and present how it works to solve the problem. Students determine how well their solution works and the social impact of their solution.

We want each student to choose his/her own specific challenge and the solution. However, if students are really stuck, you may use this example:

> Students are game designers and scientists who are working on an escape room puzzle challenge that people can do from home. Players have a set of robotic materials and download the instructions for a challenge. Players build the model and then begin the challenge. A timer will indicate the amount of time remaining. The clues allow the players to unlock the box in the time allowed – if they can figure out the steps. A minimum of 6 steps is required.

Challenge 4

Manufacturing

A novel item is being created by the company for which the students work. They will need to create an item, be able to gather and stack the items, place a stack in each location for warehouse/storage, then take from storage to shipping.

We want each student to choose his/her own specific challenge and the solution. However, if students are really stuck, you may use this example:

> Students build models from LEGO® bricks – first they gather the materials using a robot to move materials from Raw Materials area to a Production area of the company. The students can automate production or make by hand stacks of bricks. Then, students automate the system to recognize specific products and take them to the warehouse/storage area and, as needed, from the storage area to a shipping area. A customer order could be for a stack of red bricks or blue bricks or yellow bricks which must be produced and have all available in the warehouse. Then when an order arrives, only the correct product moves to shipping.

Challenge 5

Government and Public Administration and Law, Public Safety, Corrections & Security - Helping People in Times of Natural Disaster

What disasters occur in your area - earthquakes, tornadoes, hurricanes, large scale fires, blizzards? Students choose one type of disaster and determine how to create an alert system to help keep people out of harm's way and a system to find people that are trapped or to clean up areas affected by the disaster.

We want each student to choose his/her own specific challenge and the solution. However, if students are really stuck, you may use this example:

> A fire has been started by lightning and due to high, shifting winds and very dry conditions, the local populations are endangered. Students create a system to help keep people out of harms' way and a vehicle to find and rescue trapped individuals.

Challenge 6

Agriculture

There are lots of tasks that require manual labor in agriculture. Think about one of these tasks and create a way to automate it. Each student should choose his/her own specific challenge and the solution.

We want each student to choose his/her own specific challenge and the solution. However, if students are really stuck, you may use this example:

> After hay bales have been created, farmers take forklifts and trackers to move the bales onto trucks or to locations near barns. Could you create an autonomous vehicle to handle this job?

Culminating Project: Constraints and Presentation Requirements

Constraints for all challenges:

- Clear explanation of the issue and the solution
- Model must contain and use one or more sensors
- Model must contain and use one or more motors
- Model must use lights on the hub for communication
- Model must use sound for communication
- Use subcomponents within the programming

Presentation requirements:

- Explain the process used to determine the chosen solution
- Explain societal implications of the solution
- Explain how easy the solution is to use – what are any limitations (physical)
- Present the solution model and how it works
- Present how well the solution works and what improvements you would foresee in the future
- Recommendations from peer review incorporated into the solution
- Credit given to others for any original IP that was utilized to get the solution

Peer Review of Culminating Project

Students will complete a peer review after all teams have determined a problem and a possible solution. The same peer review can be done at the end of the project as well – to show growth and incorporation of feedback. A form to help with peer review is provided at the end of the unit.

After the solution ideas have been determined, students should have a peer review of all projects. Each project should receive at least two responses from classmates and incorporate the feedback into their final project.

The class may need some help with giving feedback. Consider asking questions like:

- What is something that is missing, does not seem to work, or could be improved?
- What is something that is confusing or could be done differently?
- What is something that works well or that you really like about the solution?

Review the following questions to help teams when thinking about the solution:

- Did you understand what the solution is supposed to do? (Clarify)
- What features does the solution have? Does the project seem as if it will work as expected? (Features)
- How engaging is the solution? Is it interactive, original, fun, interesting? Who would use it/interact with it? (Appeal)

Note: Students should write feedback received into their journals as well as take notes in their journals on feedback they gave.

Student Self-Assessment

The class may wish to use colors/symbols to help with feedback:

- **3 –** I gave and accepted graciously feedback to my partner and other teams. I allowed my partner to have time to build, program, explain, and document.
- **2 –** I gave feedback graciously to others but had a bit of difficulty accepting feedback from others. I didn't always give my partner equal time with building, programming, explaining, or documenting.
- **1 –** I gave little feedback to others; I accepted little feedback from others.

Team Assessment

The class may wish to use colors/symbols to help with feedback:

- **3 –** Our team worked like a well together. We gave and accepted graciously feedback from each other and from other teams. We made sure each person played a significant role in solving the problem and creating the solution.
- **2 –** Our team worked fairly well together. We had a bit of trouble now and then giving and receiving feedback from each other and from other teams. We generally made sure people were involved or engaged.
- **1 –** Our team struggled. We had a tough time giving and receiving feedback, and the work was not shared equally. Some member(s) monopolized the time together and insisted their ideas were best.

Culminating Project Rubric

Both the teacher and the team should complete the rubric, assessing the solution and presentation.

Planning	3	2	1
Developed a Plan and Diligently Worked Toward a Successful Solution. (Fostering an Inclusive Computing Culture)	Created a well-documented plan and the team worked effectively toward a solution. Documentation contains all the following: initial ideas from brainstorming, chosen topic, notes from research, initial idea for model and how it will work, daily notes on progress, issues that need to be resolved during process.	Created a plan and worked toward the solution. Documentation contains most of the following: initial ideas from brainstorming, chosen topic, notes from research, initial idea for model and how it will work, daily notes on progress, issues that need to be resolved during process.	Did not have a complete plan and team worked occasionally toward a solution. Documentation contains a little of the following: initial ideas from brainstorming, chosen topic, notes from research, initial idea for model and how it will work, daily notes on progress, issues that need to be resolved during process.
Programming (Testing and Refining Computational Artifacts)	**3**	**2**	**1**
Includes purposeful use of lights.	Uses lights in an original way.	Uses lights directly as done in a lesson.	Does not use lights.
Includes purposeful use of sounds.	Uses sound in an original way.	Uses sound directly as done in a lesson.	Does not use sound.
Written efficiently Includes subcomponents and My Blocks.	Uses several subcomponents and My Blocks.	Uses one My Block as subcomponent.	Does not use subcomponents or My Blocks.
Uses sequences and loops.	Uses multiple sequences and nested loops.	Uses a sequence and a loop.	Does not use a loop.
Uses conditional statements.	Uses multiple conditional statements.	Uses one conditional statement.	Does not use a conditional statement.
Can justify the appropriate tools and techniques.	Clear explanation of why programming was chosen and used.	Explanation is satisfactory but needs refinement.	Explanation is difficult to follow and shows a lack of understanding.
Can explain the program and its functions.	Clear explanation of how programming was chosen and used.	Explanation is satisfactory but needs refinement.	Explanation is difficult to follow and shows a lack of understanding.

Engineering Design (Testing and Refining Computational Artifacts)	**3**	**2**	**1**
Model is fully functional with purposeful use of motors.	Purposeful use of motors.	Motor used.	No motor used.
Model is fully functional with purposeful use of sensors.	Purposeful use of multiple sensors.	Sensor used.	No sensor used.
Can explain how the team iterated possible solutions as they used the design process.	Clear explanation of how the model idea was chosen and the iterations needed to get to the final version.	Some explanation of how the model idea was chosen, but only a few examples of iterations between initial idea and final version.	No clear explanation of how model idea was chosen and no iteration notes between idea and final model.
Can explain how feedback was incorporated into the design.	Clear explanation of how final model was tested with feedback noted and how feedback was used to improve.	Some explanation of how final model was tested with feedback notes, but vague information on how feedback was incorporated.	Little to no explanation of feedback received or how it was incorporated.
Documentation (Creating Computational Artifacts)	**3**	**2**	**1**
Followed the process for solving a problem.	Identified the problem; brainstormed several possible solutions; tested models; iterated model design and programming; analyzed the designs and programming; reviewed feedback; chose the best solution based on criteria.	Identified the problem, thought of a solution, built a model, and made changes, programmed, and debugged the solution.	Identified a problem, built, and programmed a model.
Created computational artifacts.	Chose appropriate tools; decomposed the problem; coded with the design in mind; debugged the program; improved on the original design.	Choose appropriate tools, built, and coded the model, debugged the program, made the model work.	Made a model and programmed it.
Selected a variety of formats to communicate ideas.	Used screen shots, video, photos, journaling, and models to communicate throughout the project.	Used photos and journaling to communicate during but mostly at the end of the project.	Made some journal entries and took a few photos at the end of the project.
Collaboration and Feedback (Collaborating Around Computing)	**3**	**2**	**1**
Used the feedback form for sharing ideas with at least 2 other groups.	Useful feedback, both positive and inspiring were provided to other teams.	Feedback was given, but minimal.	No feedback was given.

Gave an example of useful feedback shared with another team.	Feedback shows thought and was helpful to another team and could have led to improvement.	Feedback was given but was minimal.	No feedback was given.
Completed the B/Y/V feedback form for self-assessment.	Feedback was honest and thoughtful, providing both positive comments and areas where improvement could be made.	Feedback was minimal with little thought being made for what could be done in the future and what was done well.	No feedback was given.
Completed the B/Y/V feedback form for team assessment.	Feedback was honest and thoughtful, providing both positive comments and areas where improvement could be made.	Feedback was minimal with little thought being made for what could be done in the future and what was done well.	No feedback was given.
Presentation (Communicating About Computing, Collaborating Around Computing, Recognizing and Defining Computational Problems)	**3**	**2**	**1**
All team members participated. (Collaborating Around Computing)	All members participated fully.	All or most participated but dominated by one person.	Only one person presented or completed a majority of the work or no presentation occurred.
All team members answered questions. (Collaborating Around Computing)	All members participated fully.	All or most participated but dominated by one person.	Only one person presented, or no presentation occurred.
The problem was clearly identified. (Recognizing and Defining Computational Problems)	The problem was clearly identified and explanation of how the problem was determined through brainstorming and narrowing of options.	The problem was identified.	The problem was never clearly identified.
The solution solved the problem. (Collaborating Around Computing)	The solution solved the problem in an original manner through a process of iteration that was explained.	The final solution solved the problem.	The solution did not solve the problem.
The physical model was programmed and demonstrated as part of the solution.	The model demonstrated the solution and an explanation of how each part of the solution, including the model and the programming, work together to create the final solution.	The model demonstrated the solution.	The model did not demonstrate a solution.

The audience was taken into consideration when presenting.	The audience was taken into consideration and explanations were clear and thoughtful, giving necessary background, processes, and explanations.	The audience was generally taken into consideration, but some aspects such as adequate background were missing.	The audience was tangential to the presentation. Clear explanations for someone not familiar with the issue were missing.

Overview of Course with Standards

Unit 1 What is a Robot?			
Lesson	**Focus**	**Student Objectives**	**Standards**
01 Introduction to Robotics 45 minutes	• Define teamwork criteria • Outline final project • Intro to hardware -SPIKE Prime set	• Describe characteristics of working on a team • Identify hardware components • Describe the function of hardware **Vocabulary** Hardware	**CSTA** 2-AP-18 2-CS-02 **K12 Framework Practices** 2. Collaborating around computing 5. Creating computational artifacts **Concepts** Computing Systems **Sub-concepts** Hardware and software **Crosscutting concepts** Human-computer interaction System relationship Communication and collaboration

Unit 1 What is a Robot?			
Lesson	**Focus**	**Student Objectives**	**Standards**
02 What is a robot? 45 minutes	• Define robot • Introduce Hardware/Software • Getting Started Hub Lesson	• Define a robot Identify ways we use robotics to solve problems • Describe the function of hardware and software • Identify hardware components (LEGO Elements)	**CSTA** 2-AP-18 2-CS-02 **K12 CS Framework Practices** 2. Collaborating around computing 3. Recognizing and Defining Computational Problems 5. Creating Computational Artifacts **Concepts** Computing Systems **Sub-concepts** Hardware and Software Devices **Crosscutting Concepts** Human-computer interaction System relationship Privacy and security Communication and coordination

Unit 1 What is a Robot?			
Lesson	**Focus**	**Student Objectives**	**Standards**
03 Sound and Light 45 minutes	• Introduce Light and Sound blocks • Create an inspirational message using sound and lights • Use pseudocode to plan prior to writing code	• Explore light and sound blocks • Write pseudocode before programming • Describe how software and hardware work together • Explain how sound and light can be used to share a message • Program independently light and sound using the hub **Vocabulary** Ascending Descending Parameter	**CSTA** 2-AP-10 2-CS-02 3A-AP-13 **K12 CS Framework Practices** 2. Collaborating around computing 3. Recognizing and Defining Computational Problems 4. Developing and using abstractions **Concepts** Computing Systems Algorithms and Programming **Sub-concepts** Hardware and Software Program Development **Crosscutting Concepts** Human-computer interaction System relationship Communication and coordination

Unit 1 What is a Robot?			
Lesson	**Focus**	**Student Objectives**	**Standards**
04 Motors and Sensors 45 minutes	• Guided and unguided programming motors and sensors • Explanation of how motors and sensors are used in the real world	• Describe the function of motors and sensors • Program independently motors and sensors. • Describe how hardware and software work together **Vocabulary** Ultrasonic or Distance sensor Gyro sensor Force sensor	**CSTA** 2-AP-18 2-CS-02 3A-AP-13 **K12 CS Framework Practices** 2. Collaborating around computing 3. Recognizing and Defining Computational Problems 4. Developing and using abstractions **Concepts** Computing Systems Algorithms and Programming **Sub-concepts** Hardware and Software Program Development **Crosscutting Concepts** Human-computer interaction System relationship Communication and coordination

Unit 1 What is a Robot?			
Lesson	**Focus**	**Student Objectives**	**Standards**
05 Make It Move 90 minutes	• Learning to use pseudocode • Creating a prototype • Using iteration and documenting changes	• Define prototype • Investigate how changes in software can alter the performance of a robot • Develop a system for documenting changes made to a design solution • Make a claim if the Spike Prime hopper is a robot **Vocabulary** Prototype	**CSTA** 2-AP-12 2-AP-19 2-CS-02 **K-12 CS Framework Practices** 5. Creating Computational Artifacts 7. Communicating about Computing **Concepts** Computing Systems Algorithms and Programming **Sub-concepts** Hardware and Software Algorithms Program Development **Crosscutting Concepts** Human Computer interaction Abstractions System Relationship Communication and Coordination
06 Connecting to Careers Human Services and Education & Training 240 minutes	• Learn about the career pathways focusing on Human Services and Education & Training • Complete a self-reflection on individual interests and skills	• Articulate their personal interests and goals. • Relate their personal interests and goals into possible career pathways. • Explore various careers in career pathways.	**Career Ready Practice** 10 – Plan education and career path aligned to personal goals **(CCTC)**

Unit 2 Improving a Design			
Lesson	**Focus**	**Student Objectives**	**Standards**
U2-01 Iteration and Perseverance 90 minutes	• Create a prototype Iterate the prototype design • Give and receive feedback Use feedback to modify the design	• Use an iterative process to improve the performance of a robot. • Identify characteristics of specific feedback. • Describe how feedback can be used to improve a design. • Give and use specific feedback in the design process. **Vocabulary** Iteration Feedback Rapid prototyping	**CSTA** 2-CS-03 2-AP-12 2-AP-15 2-AP-17 3A-AP-16 3A-AP-21 **K12 CS Framework Practices** 2. Collaborating around computing 5. Creating Computational Artifacts 6. Testing and Refining Computational Artifacts **Concepts** Algorithms Troubleshooting Program Development Hardware and Software **Sub-concepts** Computing Systems Algorithms and Programming Program Development **Crosscutting concepts** System relationships Feedback

Unit 2 Improving a Design			
Lesson	**Focus**	**Student Objectives**	**Standards**
U2-02 Pseudocode 45 minutes	Read, write, compare pseudocode with coding	• Write pseudocode before programming. • Read pseudocode and program to match. • Compare pseudocode to a program. **Vocabulary** Pseudocode Clarify	**CSTA** 2-AP-10 **K-12 CS Framework Practices** 2. Collaborating Around Computing 5. Creating Computational Artifacts 6. Testing and Refining Artifacts 7. Communicating About Computing **Concepts** Algorithms **Sub-Concepts** Hardware and Software Troubleshooting Algorithms **Crosscutting Concepts** System Relationships Communication and Coordination

Unit 2 Improving a Design			
Lesson	**Focus**	**Student Objectives**	**Standards**
U2-03 Create a New Product 45 minutes	Design a product and present it to the class	• Be creative. • Design a new product and present it to the class in a commercial. **Vocabulary** Creativity	**CSTA** 2-CS-02 2-AP-15 **K-12 CS Framework Practices** 2. Collaborating Around Computing 5. Creating Computational Artifacts 7. Communicating About Computing **Concepts** **Sub-Concepts** Hardware and Software **Crosscutting Concepts** Communication and Coordination
U2-04 Share Your Ideas 45 minutes	• Determine if a task is suitable for automation Communicate your ideas to others	• Determine if a task can be automated. • Communicate to others your ideas. **Vocabulary** Automate	

Unit 2 Improving a Design			
Lesson	**Focus**	**Student Objectives**	**Standards**
U2-05 Mini-Challenge: Creativity in Business 45-90 minutes	• Create a new idea and present it to the class. Unguided building and programming.	• Create a new robot. • Present your ideas persuasively to others. • Integrate effective presentation skills into a business pitch. **Vocabulary** Entrepreneur	**CSTA** 2-CS-02 2-CS-03 2-AP-15 **K-12 CS FRAMEWORK Practices** 2. Collaborating Around Computing 5. Creating Computational Artifacts 7. Communicating About Computing **Concepts** Computing Systems **Sub-Concepts** Hardware and Software **Crosscutting Concepts** Communication and Coordination
U2-06 Connecting to Careers: Marketing, Arts, A/V Technology & Communications 90 minutes	• Relate their personal interests and goals into possible career pathways. • Explore various careers in 2 career pathways.	• Articulate their personal interests and goals. • Relate their personal interests and goals into possible career pathways. • Explore various careers in career pathways.	Career Ready Practice 10- Plan education and career path aligned to personal goals. **(CCTC)**

Unit 3 Troubleshooting and Debugging			
Lesson	• **Focus**	• **Student Objectives**	**Standards**
U3 - 01 Model Debugging 90 minutes	• Troubleshoot hardware	• Identify and repair a hardware problem in a physical model. **Vocabulary** Debugging CNC	**CSTA** 2-AP-13 3A-CS-03 3A-AP-16 3A-AP-17 **K12 CS Framework** 2. Collaborating around computing 3. Recognizing and Defining Computational Problems 6. Testing and Refining Computational Artifacts **Concepts** Computer Systems **Sub-concepts** Troubleshooting **Crosscutting Concepts** Abstraction System Relationships Communication and Coordination

Unit 3 Troubleshooting and Debugging			
Lesson	• **Focus**	• **Student Objectives**	**Standards**
U3-02 Software Debugging 90 minutes	• Learn tips and tricks for debugging software • Debug a program	• Debug a software problem. • Learn and follow a debugging strategy involving multiple steps. **Vocabulary** Debugging Pseudocode	**CSTA** 2-AP-10 2-AP-13 2-AP-17 3A-CS-03 3A-AP-16 3A-AP-17 **K12 CS Framework Practices** 2. Collaborating around computing 3. Recognizing and Defining Computational Problems 6. Testing and Refining Computational Artifacts **Concepts** Computer Systems **Sub-concepts** Troubleshooting **Crosscutting Concepts** Abstraction System Relationships Communication and Coordination

Unit 3 Troubleshooting and Debugging			
Lesson	• **Focus**	• **Student Objectives**	**Standards**
U3 – 03 Dancer Break Down 45 minutes	• Identify problems and debug the program. • Utilize pseudocode to assist with debugging.	• Identify a problem and debug the program. • Compare pseudocode to code to help with debugging. • Write pseudocode from code to help with debugging. **Vocabulary** Debugging	**CSTA** 2-AP-10 2-AP-13 2-AP-17 2-AP-19 **K12 CS Framework Practices** 2. Collaborating around computing 3. Recognizing and Defining Computational Problems 6. Testing and Refining Computational Artifacts **Concepts** Computer Systems **Sub-concepts** Troubleshooting **Crosscutting Concepts** Abstraction System Relationships Communication and Coordination

Unit 3 Troubleshooting and Debugging			
Lesson	• **Focus**	• **Student Objectives**	**Standards**
U3-04 Dance to the Color 45 minutes	• Identify a problem and debug the program. • Find and fix issues with hardware, software, or interface between the two.	• Identify a problem and debug the program. • Identify and fix problems that are hardware or software or the interface of hardware and software. **Vocabulary** Debugging	**CSTA** 2-AP-10 2-AP-13 2-AP-17 2-AP-19 **K12 CS Framework Practices** 2. Collaborating around computing 3. Recognizing and Defining Computational Problems 6. Testing and Refining Computational Artifacts **Concepts** Computer Systems **Sub-concepts** Troubleshooting **Crosscutting Concepts** Abstraction System Relationships Communication and Coordination

Unit 3 Troubleshooting and Debugging			
Lesson	• **Focus**	• **Student Objectives**	**Standards**
U3-05 Mini-Challenge: Design a Route 90 minutes	• Iterate a solution, program the robot and debug the program.	• Program a robot that can autonomously move from one location to another following a prescribed route. • Iterate a solution and debug a software problem. **Vocabulary** Debugging	**CSTA** 2-AP-10 2-AP-13 2-AP-17 2-AP-19 **K12 CS Framework Practices** 2. Collaborating around computing 3. Recognizing and Defining Computational Problems 6. Testing and Refining Computational Artifacts **Concepts** Computer Systems **Sub-concepts** Troubleshooting **Crosscutting Concepts** Abstraction System Relationships Communication and Coordination
U3 – 06 Connecting to Careers: Information Technology and Law, Public Safety, Corrections & Security 90 minutes	• Relate their personal interests and goals into possible career pathways. • Explore various careers in 2 career pathways.	• Articulate their personal interests and goals. • Relate their personal interests and goals into possible career pathways. • Explore various careers in career pathways.	Career Ready Practice 10- Plan education and career path aligned to personal goals. (CCTC)

Unit 4 Testing and Evaluating Solutions			
Lesson	**Focus**	**Student Objectives**	**Standards**
U4 – 01 Testing Prototypes 90 minutes	• Brainstorm and create a solution for a bridge. • Build and program a model, code it for movement, and debug the program.	• Brainstorm ideas and develop solutions to a problem involving a bridge with movement. • Build a working model and write code. • Test the model and debug the program. • Use the design engineering process. **Vocabulary** Engineering Design Process Constraint Prototype	**CSTA** 2-AP-18 2-AP-12 2-CS-01 2-CS-02 **K12 CS Framework Practices** 2. Collaborating Around Computing 3. Recognizing and Defining Computational Problems 5. Creating Computational Artifacts 7. Communicating about Computing **Concepts** Computing Systems Data and Analysis Impacts of Computing **Sub-concepts** Devices Hardware and Software Troubleshooting Visualization and Transformation Social Interaction **Crosscutting Concepts** System Relationships Human-Computer Interaction

Unit 4 Testing and Evaluating Solutions			
Lesson	• **Focus**	• **Student Objectives**	**Standards**
U4 - 02 Human vs. Robot 90 minutes	• Compare how well a human hand and a robotic hand can pass bricks. • Collect and analyze data on the effectiveness of passing bricks. • Determine tasks that are difficult for robots to solve.	• Identify problems that are difficult for a robot to solve. • Use data to support a claim as to why a robot would be preferred over a human for a task. • Use data to compare a task completed by a human and a robot. • Provide data to show a time when a human is more effective than a robot. **Vocabulary** Accessibility Manufacturing Categorize	**CSTA** 2-AP-18 2-AP-12 2-CS-01 CS-2-02 **K12 CS Framework Practices** 2. Collaborating Around Computing 3. Recognizing and Defining Computational Problems 5. Creating Computational Artifacts 7. Communicating about Computing **Concepts** Computing Systems Data and Analysis Impacts of Computing **Sub-concepts** Devices Hardware and Software Troubleshooting Visualization and Transformation Social Interaction **Crosscutting Concepts** System Relationships Human-Computer Interaction

Unit 4 Testing and Evaluating Solutions			
Lesson	**Focus**	**Student Objectives**	**Standards**
U4-03 Comparing Robotic Grabbers 90 minutes	• Introduce idea of societal impact • Use data to support a claim • Use data to compare tasks completed • Identify positive and negative consequences	• Identify problems that are difficult for a robot to solve. • Use data to support a claim as to why a robot would be preferred over a human for a task. • Use data to compare a task completed by a human and a robot. • Identify positive and negative consequences for automating a human task. **Vocabulary** Effective	**CSTA** 2-AP-18 2-AP-12 2-CS-01 2-CS-02 **K12 CS Framework Practices** 2. Collaborating Around Computing 5. Creating Computational Artifacts 6. Testing and Refining Computational Artifacts 7. Communicating about Computing **Concepts** Computing Systems Data and Analysis Impacts of Computing **Sub-concepts** Troubleshooting Visualization and Transformation Social Interactions **Crosscutting Concepts** System Relationships Human-Computer Interaction

Unit 4 Testing and Evaluating Solutions			
Lesson	• **Focus**	• **Student Objectives**	**Standards**
U4-04 Repetitive Tasks 90 minutes	• Explain why robots instead of humans are used for repetitive tasks • Explain how speed affects accuracy	• Collect data around the accuracy of a robot at various speeds • Identify the benefits of using robots for repetitive tasks. • Use data to support the claim that for some tasks a robot is more effective than a human. **Vocabulary** Repetitive Clockwise Counterclockwise	**CSTA** 2-AP-18 2-AP-12 2-AP-16 **K-12 CS Framework Practices** 2. Collaborating Around Computing 5. Creating Computational Artifacts 6. Testing and Refining Computational Artifacts 7. Communicating about Computing **Concepts** Computing Systems Data and Analysis Algorithms Impacts of Computing **Sub-concepts** Hardware and Software Troubleshooting Algorithms Control Visualization and Transformation **Crosscutting Concepts** Communication and Coordination System Relationships

Unit 4 Testing and Evaluating Solutions			
Lesson	**Focus**	**Student Objectives**	**Standards**
U4-05 Turns, Speed, & Accuracy 90 minutes	• Moving a robot with 2 wheels • Making three types of turns • Comparing speed and accuracy	• Identify the three types of turns – point, pivot, and arc • Identify how speed affects accuracy. • Determine how to give credit to others for their intellectual property. • Share programming ideas and adjust programs based on feedback **Vocabulary** Arc turn Pivot turn Point turn Clockwise Counterclockwise	**CSTA** 2-AP-18 2-AP-12, 2-CS-01 **K-12 CS Framework Practices** 2. Collaborating Around Computing 4. Developing and Using Abstractions 5. Creating Computational Artifacts 6. Testing and Refining Computational Artifacts 7. Communicating about Computing **Concepts** Computing Systems Data and Analysis Algorithms and Programming **Sub-concepts** Troubleshooting Visualization and Transformation Algorithms **Crosscutting Concepts** Abstraction System Relationships Communication and Coordination

Unit 4 Testing and Evaluating Solutions			
Lesson	• **Focus**	• **Student Objectives**	**Standards**
U4-06 Uphill Climb 90 minutes	• Graph and analyze created data around energy.	• Build and program a robot to move uphill autonomously. • Graph and analyze data. • Create data around energy. **Vocabulary** Incline Potential energy	**CSTA** 2-CS-02 2-DA-09 **K-12 CS Framework Practices** 2. Collaborating Around Computing 6. Testing and Refining Computational Artifacts 7. Communicating about Computing **Concepts** Computing Systems Data and Analysis **Sub-concepts** Troubleshooting Visualization and Transformation Algorithms **Crosscutting Concepts** Abstraction System Relationships Communication and Coordination

Unit 4 Testing and Evaluating Solutions			
Lesson	• **Focus**	• **Student Objectives**	**Standards**
U4-07 Mini-Challenge: Design for Someone 90-135 minutes	• Design and program a prosthetic hand	• Design and program a prosthetic hand. • Sketch your idea and use pseudocode to assist with writing code. • Present your model and ideas to the class. **Vocabulary** Entrepreneur Prosthesis	**CSTA** 2-CS-01 2-CS-02 2-CS-03 2-AP-17 2-AP-18 **K-12 CS FRAMEWORK Practices** 2. Collaborating Around Computing 5. Creating Computational Artifacts 7. Communicating About Computing **Concepts** Computing Systems **Sub-Concepts** Hardware and Software **Crosscutting Concepts** Communication and Coordination
U4 – 08 Connecting to Careers: Health Science and Agriculture, Food, & Natural Resources 90 minutes	• Relate their personal interests and goals into possible career pathways. • Explore various careers in 2 career pathways. •	• Articulate their personal interests and goals. • Relate their personal interests and goals into possible career pathways. • Explore various careers in career pathways. •	Career Ready Practice 10- Plan education and career path aligned to personal goals. (CCTC)

Unit 5 Sensors			
Lesson	• **Focus**	• **Student Objectives**	**Standards**
U5-01 Sensors Trigger Reactions 90 minutes	• Using sensors and motors together • Move until ultrasonic sensor/color sensor trigger a stop	• Write pseudocode to design a program • Use comments to explain code • Utilize conditional statements in a program using sensors and motors **Vocabulary** Conditional	**CSTA** 2-AP-10 2-AP-12 2-AP-13 2-AP-18 2-CS-02 **K-12 CS Framework Practices** 2. Collaborating Around Computing 5. Creating Computational Artifacts 6. Testing and Refining Computational Artifacts **Concepts** Computing Systems Algorithms and Programming **Sub-concepts** Troubleshooting Algorithms Control **Crosscutting Concepts** System Relationships

Unit 5 Sensors			
Lesson	• **Focus**	• **Student Objectives**	**Standards**
U5-02 Sensors and Data 90 minutes	• Use a distance sensor to stop the robot a given distance from the wall at different speeds. • Collect data to determine how speed affects accuracy.	• Write pseudocode before programming. • Use conditional statements in a program using sensors and motors. • Collect, graph and analyze data. **Vocabulary** Conditional	**CSTA** 2-AP-10 2-AP-12 2-AP-13 2-AP-18 2-CS-02 **K-12 CS Framework Practices** 2. Collaborating Around Computing 5. Creating Computational Artifacts 6. Testing and Refining Computational Artifacts **Concepts** Computing Systems Algorithms and Programming **Sub-concepts** Troubleshooting Algorithms Control **Crosscutting Concepts** System Relationships

Unit 5 Sensors			
Lesson	• **Focus**	• **Student Objectives**	**Standards**
U5-03 Dance to Debug 45 minutes	• Identify a problem and debug the program.	• Program a robot to react to color. • Program movement that is rhythmic. • Identify a problem and debug the program. **Vocabulary** Debugging	**CSTA** 2-AP-13 3A-AP-17 2-AP-18 2-CS-03 3A-CS-03 **K-12 CS Framework Practices** 2. Collaborating Around Computing 5. Creating Computational Artifacts 6. Testing and Refining Computational Artifacts **Concepts** Computing Systems Algorithms and Programming **Sub-concepts** Troubleshooting Algorithms Control **Crosscutting Concepts** System Relationships

Unit 5 Sensors			
Lesson	• **Focus**	• **Student Objectives**	**Standards**
U5-04 Maze 90-135 minutes	• Program accurate movement using sensors	• Create a maze and program a robot to autonomously navigate it. • Program a model to move safely and accurately using sensors and motors. • Use iteration and debugging skills. **Vocabulary** Traverse Autonomous	**CSTA** 2-AP-10 2-AP-13 2-AP-16 2-AP-17 2-AP-19 2-CS-02 **K-12 CS Framework Practices** 2. Collaborating Around Computing 5. Creating Computational Artifacts 6. Testing and Refining Computational Artifacts **Concepts** Algorithms and Programming **Sub-concepts** Algorithms Control **Crosscutting Concepts** Communication and Coordination

Unit 5 Sensors			
Lesson	• **Focus**	• **Student Objectives**	**Standards**
U5-05 Factory Robot 90-135 minutes	• Design, build, and program a robot that can identify and move materials.	• Design, build, and program a factory robot that can identify and move materials autonomously. • Use pseudocode and comments to assist in writing code and debugging. **Vocabulary** Constraint	**CSTA** 2-AP-10 2-AP-12 2-AP-13 2-AP-17 2-AP-19 2-CS-02 **K-12 CS Framework Practices** 2. Collaborating Around Computing 5. Creating Computational Artifacts 6. Testing and Refining Computational Artifacts 7. Communicating About Computing **Concepts** Algorithms and Programming **Sub-concepts** Hardware and Software Troubleshooting Algorithms Control **Crosscutting Concepts** System Relationships Communication and Coordination

Unit 5 Sensors			
Lesson	• **Focus**	• **Student Objectives**	**Standards**
U5-06 Parking Lot 90-135 minutes	• Use computational thinking in a real-life scenario.	• Coordinate programming with another team • Apply computational thinking to real-life problems. • Coordinate robots' movements within constraints to obtain a goal. **Vocabulary** Constraint	**CSTA** 2-AP-10 2-AP-12 2-AP-13 2-AP-15 2-AP-18 2-AP-19 2-CS-02 **K-12 CS Framework Practices** 2. Collaborating Around Computing 5. Creating Computational Artifacts 6. Testing and Refining Computational Artifacts 7. Communicating About Computing **Concepts** Algorithms and Programming **Sub-concepts** Hardware and Software Troubleshooting Algorithms Control **Crosscutting Concepts** Communication and Coordination

Unit 5 Sensors			
Lesson	• **Focus**	• **Student Objectives**	**Standards**
U5-07 Mini-Challenge: Parking Lot 90-135 minutes	• Use sensors in a real-life scenario.	• Use conditional statements in a program using sensors and motors. • Apply computational thinking and sensors to real-life problems. • Coordinate robots' movements within constraints to obtain a goal. • Coordinate programming with another team. • Write pseudocode and give credit through comments for code borrowed from others. **Vocabulary** Constraint	**CSTA** 2-AP-10 2-AP-12 2-AP-13 2-AP-15 2-AP-18 2-AP-19 2-CS-02 **K-12 CS Framework Practices** 2. Collaborating Around Computing 5. Creating Computational Artifacts 6. Testing and Refining Computational Artifacts 7. Communicating About Computing **Concepts** Algorithms and Programming **Sub-concepts** Hardware and Software Troubleshooting Algorithms Control **Crosscutting Concepts** Communication and Coordination

LEGO education

Unit 5 Sensors			
Lesson	• **Focus**	• **Student Objectives**	**Standards**
U5-08 Connecting to Careers: Manufacturing and Transportation, Distribution & Logistics 90 minutes	• Relate their personal interests and goals into possible career pathways. • Explore various careers in 2 career pathways.	• Articulate their personal interests and goals. • Relate their personal interests and goals into possible career pathways. • Explore various careers in career pathways.	Career Ready Practice 10- Plan education and career path aligned to personal goals. (CCTC)

Unit 6 Variables			
Lesson	**Focus**	**Student Objectives**	**Standards**
U6-01 Drone Pitch & Roll 90 minutes	• Use conditional statements in a program using sensors • Analyze data from sensors	• Use conditional statements in a program using sensors. • Apply sensors to real-life problems by analyzing sensor data. • Use pitch and roll to move like a drone around obstacles and to create data for analysis.	**CSTA** 2-CS-02 2-AP-12 2-AP-13 2-AP-17 2-DA-08 **K12 CS FRAMEWORK Practices** 2. Collaborating Around Computing 5. Creating Computational Artifacts 6. Testing and Refining Computational Artifacts **Concepts** Computing Systems Algorithms and Programming **Sub-concepts** Troubleshooting Algorithms Control **Crosscutting Concepts** System Relationships

Unit 6 Variables			
Lesson	**Focus**	**Student Objectives**	**Standards**
U6-02 Drone Movement 90 minutes	• Use conditional statements in a program using sensors • Analyze data from sensors	• Use yaw, pitch, and roll to move like a drone around obstacles and to create data for analysis. • Apply sensors to real-life problems. • Analyze graphs to determine movement. **Vocabulary** Yaw Pitch Roll	**CSTA** 2-CS-02 2-AP-12 2-AP-13 2-AP-17 2-DA-08 **K12 CS FRAMEWORK Practices** 2. Collaborating Around Computing 5. Creating Computational Artifacts 6. Testing and Refining Computational Artifacts **Concepts** Computing Systems Algorithms and Programming **Sub-concepts** Troubleshooting Algorithms Control **Crosscutting Concepts** System Relationships

Unit 6 Variables			
Lesson	**Focus**	**Student Objectives**	**Standards**
U6-03 Variables 90 minutes	Create variables and write a program using the variables to keep score.	• Describe the function of a variable • Create clearly named variables • Program using variables after planning the program using pseudocode. **Vocabulary** Variable	**CSTA** 2-AP-11 2-AP-18 2-CS-02 **K-12 CS Framework Practices** 5. Creating Computational Artifacts **Concepts** Algorithms and Programming **Sub-concepts** Troubleshooting Algorithms Variables **Crosscutting Concepts** Abstraction

Unit 6 Variables			
Lesson	**Focus**	**Student Objectives**	**Standards**
U6-04 Graphing, Speed, and Distance 90 minutes	Graph and analyze data	• Use variables to determine speed. • Graph and analyze data • Create data around energy **Vocabulary** Rotations Circumference Distance	**CSTA** 2-AP-13 2-AP-17 2-DA-08 3A-CS-03 3A-AP-16 3A-AP-17 **K12 CS FRAMEWORK Practices** 2. Collaborating Around Computing 3. Recognizing and Defining Computational Problems 6. Testing and Refining Computational Artifacts **Concepts** Computer Systems **Sub-concepts** Troubleshooting **Crosscutting concepts** Abstraction System Relationships Communication and Coordination

Unit 6 Variables			
Lesson	**Focus**	**Student Objectives**	**Standards**
U6-05 Mini-Challenge: Distance Game 90 minutes	Use data to win a game	• Use data analysis as strategy for winning a game. • Create and program multiple variables for gathering data. • Create a program to determine initial kinetic energy. **Vocabulary** Kinetic energy Variable Mass Velocity	**CSTA** 2-AP-11 2-AP-13 2-DA-08 3A-AP-13 3A-AP-16 **K12 CS FRAMEWORK Practices** 2. Collaborating Around Computing 3. Recognizing and Defining Computational Problems 6. Testing and Refining Computational Artifacts **Concepts** Computer Systems **Sub-concepts** Troubleshooting **Crosscutting concepts** Abstraction System Relationships Communication and Coordination

Unit 6 Variables			
Lesson	**Focus**	**Student Objectives**	**Standards**
U6-06 Connecting to Careers: Finance and Business Management 90 minutes	• Relate their personal interests and goals into possible career pathways. • Explore various careers in 2 career pathways.	• Articulate their personal interests and goals. • Relate their personal interests and goals into possible career pathways. • Explore various careers in career pathways.	Career Ready Practice 10- Plan education and career path aligned to personal goals. (CCTC)

Unit 7 Arrays, Boolean Expressions, and Conditionals			
Lesson	• **Focus**	• **Student Objectives**	**Standards**
U7-01 Intro to Arrays (Lists) & Compound Conditionals 90 minutes	• Create and utilize arrays (lists) in a program with compound conditionals.	• Create and utilize arrays (lists). • Code with compound conditionals using lists (arrays). • Read pseudocode and compare with a program. **Vocabulary** Array List	**CSTA** 2-CS-01 2-CS-02 2-DA-09 2-AP-10 2-AP-12 2-AP-13 2-AP-16 2-AP-18 **K12 CS FRAMEWORK** **Practices** 2. Collaborating Around Computing 4. Developing and Using Abstractions 5. Creating Computational Artifacts 6. Testing and Refining Computational Artifacts 7. Communicating About Computing **Concepts** Computing Systems Algorithms **Sub-concepts** Troubleshooting Algorithms Control **Crosscutting Concepts** Abstraction System Relationships

Unit 7 Arrays, Boolean Expressions, and Conditionals			
Lesson	**Focus**	**Student Objectives**	**Standards**
U7-02 Comparing Arrays 135 minutes	• Create and compare arrays	• Create and compare arrays (lists). • Use conditionals with multiple operators. • Use Boolean operators while coding. **Vocabulary** Array Conditional	**CSTA** 2-CS-01 2-CS-02 2-DA-09 2-AP-10 2-AP-12 2-AP-13 2-AP-16 2-AP-18 **K-12 CS Framework Practices** 2. Collaborating Around Computing 4. Developing and Using Abstractions 5. Creating Computational Artifacts 6. Testing and Refining Computational Artifacts 7. Communicating About Computing **Concepts** Computing Systems Algorithms **Sub-components** Troubleshooting Algorithms Control **Crosscutting Concepts** Abstraction System Relationships

Unit 7 Arrays, Boolean Expressions, and Conditionals			
Lesson	**Focus**	**Student Objectives**	**Standards**
U7-03 Conditionals and Simplifying Code 90 minutes	• Use conditionals to follow a line • Create a My Blocks to create subprograms	• Use conditionals to make a robot follow a line. • Debug programs using sub-components. • Utilize My Blocks when creating subprograms within a comprehensive program. **Vocabulary** Subcomponent My Block	**CSTA** 2-AP-10 2-AP-12 2-AP-13 2-AP-14 2-AP-16 2-AP-18 3A-AP-16 3A-AP-17 **K-12 CS Framework Practices** 2. Collaborating Around Computing 4. Developing and Using Abstractions 5. Creating Computational Artifacts 6. Testing and Refining Computational Artifacts 7. Communicating About Computing **Concepts** Computing Systems Algorithms **Sub-components** Troubleshooting Algorithms Control **Crosscutting Concepts** Abstraction System Relationships Communication and Coordination

Unit 7 Arrays, Boolean Expressions, and Conditionals			
Lesson	• **Focus**	• **Student Objectives**	**Standards**
07-04 Conditionals and Boolean Expressions 90 minutes	• Create new My Blocks to incorporate with previously created subcomponents. • Evaluate Boolean expressions and create conditional statements	• Use Boolean expressions and My Blocks when writing code. • Evaluate and write Boolean expressions with multiple conditions • Debug programs containing subcomponents. **Vocabulary** Boolean expression My Block Subcomponent Evaluate	**CSTA** 2-AP-10 2-AP-12 2-AP-13 2-AP-14 2-AP-16 2-AP-17 2-AP-19 3A-AP-17 **K-12 CS Framework Practices** 2. Collaborating Around Computing 4. Developing and Using Abstractions 5. Creating Computational Artifacts 6. Testing and Refining Computational Artifacts 7. Communicating About Computing **Concepts** Computing Systems Algorithms **Sub-components** Troubleshooting Algorithms Control **Crosscutting Concepts** Abstraction System Relationships Communication and Coordination

Unit 7 Arrays, Boolean Expressions, and Conditionals			
Lesson	• **Focus**	• **Student Objectives**	**Standards**
07-05 Mini-Challenge: Security with Operators 90 minutes	• Program using Boolean operators	• Program an alarm using Boolean operators. • Use pseudocode to help write code that must meet multiple requirements. • Iterate and debug coding to create a working security device. **Vocabulary** Boolean operator	**CSTA** 2-AP-10 2-AP-12 2-AP-13 2-AP-14 2-AP-16 2-AP-17 2-AP-19 3A-AP-17 **K-12 CS Framework Practices** 2. Collaborating Around Computing 4. Developing and Using Abstractions 5. Creating Computational Artifacts 6. Testing and Refining Computational Artifacts 7. Communicating About Computing **Concepts** Computing Systems Algorithms **Sub-components** Troubleshooting Algorithms Control **Crosscutting Concepts** Abstraction System Relationships Communication and Coordination

Unit 7 Arrays, Boolean Expressions, and Conditionals			
Lesson	**Focus**	**Student Objectives**	**Standards**
U7-06 Connecting to Careers: Hospitality, Tourism, & Government 90 minutes	• Relate their personal interests and goals into possible career pathways. • Explore various careers in 2 career pathways.	• Articulate their personal interests and goals. • Relate their personal interests and goals into possible career pathways. • Explore various careers in career pathways.	Career Ready Practice 10- Plan education and career path aligned to personal goals. (CCTC)

Unit 8 Compound Conditionals			
Lesson	**Focus**	**Student Objectives**	**Standards**
08-01 Game with Variables 90 minutes	Create a game that uses variables.	• Create a game for two players. • Keep score by using variables. **Vocabulary** Variable	**CSTA** 2-CS-02 2-CS-03 2-AP-11 2-AP-15 2-AP-17 3A-AP-13 3A-AP-16 **K-12 CS Framework Practices** 2. Collaborating Around Computing 4. Developing and Using Abstractions 5. Creating Computational Artifacts 6. Testing and Refining Computational Artifacts 7. Communicating About Computing **Concepts** Computing Systems Algorithms **Sub-components** Troubleshooting Algorithms Social Interactions **Crosscutting Concepts** System Relationships Human-Computer Interaction Communication and Coordination

Unit 8 Compound Conditionals			
Lesson	**Focus**	**Student Objectives**	**Standards**
08-02 Compound Conditionals 90 minutes	Use compound conditionals for security	• Identify ways to create a more secure entrance into a device. • Use conditional statements and compound conditionals to mimic passwords that unlock a box. **Vocabulary** Compound conditional Complexity	**CSTA** 2-CS-01, 2-CS-02 2-CS-03 2-AP-10 2-AP-12 2-AP-13 2-AP-16 2-AP-17 2-AP-19 3A-AP-13 3A-AP-21 2-IC-20 2-DA-09 **K-12 CS Framework Practices** 2. Collaborating Around Computing 3. Recognizing and Defining Computational Problems 4. Developing and Using Abstractions 5. Creating Computational Artifacts 6. Testing and Refining Computational Artifacts 7. Communicating About Computing **Concepts** Computing Devices Algorithms and Programming **Sub-concepts** Troubleshooting Algorithms Control **Crosscutting Concepts** Algorithms and Programming

Unit 8 Compound Conditionals			
Lesson	**Focus**	**Student Objectives**	**Standards**
08-03 Compounding Conditionals 90 minutes	Write programs including compound conditionals and Boolean Expressions	• Identify ways to create a more secure entrance into a device. • Build and program a device to keep an object very secure. • Use compound conditionals to mimic multi-factor authentication.	**CSTA** 2-CS-01, 2-CS-02 2-CS-03 2-AP-10 2-AP-12 2-AP-13 2-AP-16 2-AP-17 2-AP-19 3A-AP-13 3A-AP-21 2-IC-20 2-DA-09 **K-12 CS Framework Practices** 2. Collaborating Around Computing 3. Recognizing and Defining Computational Problems 4. Developing and Using Abstractions 5. Creating Computational Artifacts 6. Testing and Refining Computational Artifacts 7. Communicating About Computing **Concepts** Computing Devices Algorithms and Programming **Sub-concepts** Troubleshooting Algorithms Control **Crosscutting Concepts** Algorithms and Programming

Unit 8 Compound Conditionals			
Lesson	**Focus**	**Student Objectives**	**Standards**
08-04 Mini-Challenge: Break Out Room 90 minutes	Program using Boolean operators	• Write code that includes conditions that must be met in a game format. • Create a game that requires a series of robot responses in sequence. **Vocabulary** Constraints	**CSTA** 2-AP-10 2-AP-12 2-AP-13 2-AP-14 2-AP-16 2-AP-17 2-AP-19 2-DA-09 3A-AP-16 3A-AP-17 **K-12 CS Framework Practices** 2. Collaborating Around Computing 4. Developing and Using Abstractions 5. Creating Computational Artifacts 6. Testing and Refining Computational Artifacts 7. Communicating About Computing **Concepts** Computing Systems Algorithms **Sub-components** Troubleshooting Algorithms Control **Crosscutting Concepts** Abstraction System Relationships Communication and Coordination

Unit 8 Compound Conditionals			
Lesson	**Focus**	**Student Objectives**	**Standards**
U8-05 Connecting to Careers: Architecture & Construction and Science, Technology, Engineering and Mathematics 90 minutes	• Relate their personal interests and goals into possible career pathways. • Explore various careers in 2 career pathways.	• Articulate their personal interests and goals. • Relate their personal interests and goals into possible career pathways. • Explore various careers in career pathways.	Career Ready Practice 10- Plan education and career path aligned to personal goals. (CCTC)

Unit 9 Careers Culminating Activity			
Lesson	**Focus**	**Student Objectives**	**Standards**
U9-01 Culminating Activity: Careers 90 minutes	• Determine the careers that impact a real-world situation around attendees at a convention.	• Determine the careers that impact a real-world situation. • Articulate their personal interests and goals. • Relate their personal interests about careers. • Explore various careers in career pathways.	Career Ready Practice 10- Plan education and career path aligned to personal goals. (CCTC)

Unit 9 Careers Culminating Activity			
Lesson	**Focus**	**Student Objectives**	**Standards**
U9-02 My Career Interests 120 minutes	• Relate their personal interests and goals into possible career pathways. • Compare their responses to career interests at the beginning of the course with their responses at the end of the course	• Articulate their personal interests and goals. • Relate their personal interests and goals into possible career pathways. • Explore various careers in career pathways. • Compare their responses to career interests at the beginning of the course with their responses at the end of the course.	Career Ready Practice 10- Plan education and career path aligned to personal goals. (CCTC)
U9-03 My Career Reflection and Plan 120 minutes	• Relate their personal interests and goals into possible career pathways. • Reflect on their individual skills and interests • Create a plan for moving into a specific career	• Create a personal plan for moving into a specific career. • Articulate their personal interests and goals. • Relate their personal interests and goals into possible career pathways. • Explore various careers in career pathways.	Career Ready Practice 10- Plan education and career path aligned to personal goals. (CCTC)

Unit 10 Physical Computing Culminating Activity			
Lesson	**Focus**	**Student Objectives**	**Standards**
U10 Culminating Activity 500 minutes	Culminating Activity: Choose Your Own Project that relates to one of the career pathways.	• Solve a real-world problem and design, build, and program a model as part of the solution • Iterate design ideas and debug program prior the final presentation • Present the solution, answering questions, and showcase the model • Evaluate your solution, your team and yourself.	**CSTA** 2-CS-01, 2-CS-02, 2-CS-03 2-AP-10, 2-AP-11, 2- AP-12 2-AP-13, 2-AP-15, 2-AP-16, 2-AP-17, 2-AP-18 2-AP-19 2-DA-09 3A-AP-13, 3A-AP-16 3A-AP-17, 3A-AP-21 2-IC-20, 2-IC-21, 2-IC-22 **K-12 CS Framework Practices** 1, 2, 3, 4, 5, 6, 7 **Concepts** Computing Systems, Data and Analysis Algorithms and Programming Impacts of Computing **Sub-Components** Hardware and Software, Troubleshooting Visualization and Transformation Algorithms, Control, Modularity, Law and Ethics Program Development, Culture, Safety **Crosscutting Concepts** Abstraction, System Relationships Human-Computer Interaction Privacy and Security Communication and Coordination

Foundations of Physical Computing Student Guide

Student Guide

Introduction

Welcome to Foundations of Physical Computing. Computer Science and robots are fully integrated into our lives. We use them at home, at work, in schools, in the field of medicine, engineering, science, and so much more. In this course, you will discuss what robots are, how robots affects human life, and the computational skills needed when programming robots. You will complete physical activities, computational thinking activities, and of course, activities to design and program robots. You will also learn the fundamentals of computer science – algorithms, debugging, conditional statements, and more.

This student guide includes a lot of information that you might not understand at first – and that is okay! That is why you are taking this course.... Do not worry about that now; it will make more sense as you complete the lessons. You will be referring to this guide throughout the course. You can make notes on it or in your journal to remember things as you get ideas or things are clarified.

Knowledge and Skills Applied

You will have experiences to reinforce and apply many of the skills learned, such as working together to solve problems, using block code, persevering through failure, and enjoying the thrill of seeing your ideas come to life. Be prepared to collaborate, communicate, create, and use computational thinking every day while you design, build, program, debug, and iterate to solve problems.

Helpful Tips for Debugging

- Say aloud what the problem is. For example, “The right motor is not turning.”
- Say aloud what you expected it to do. For example, both motors should turn on at the same time.
- Have you checked the physical model to be sure all the connections are solid? If so, then test your program a couple of times to be sure it always happens.
- Check the ports to be sure they match with the wiring to the hub.
- Find the most recent code you added. Did it work BEFORE you added the code? If so, read the new code carefully.
- If you started with one large program, break it into parts, verifying a small amount of code at a time. Keep adding code until you find where it breaks. Then, debug that section.
- Explain the code to your partner. Have your partner verify the code.
- Look for pieces of code that are directly related to the issue.
- Make one slight change and test the code.
- Continue to make minor changes until you get the intended behavior.
- Once you have debugged the program, take a minute to document what you have learned. What tips can you give yourself for the next time you need to debug?

Pair Programming

- Each team should have a driver and a navigator. The students will switch roles several times during each class.
- A “driver” is thinking about the programming needed and using the computer to type in the code.
- A “navigator” helps direct the driver, making sure all the code is correctly written – the ports are correct, the operators are precise, and so forth.
- Think of it as the driver holds and moves the steering wheel on the car, while the navigator has the map and tells the driver when and where to turn.
- Switch roles several times within a class period.

Feedback Model

- Feedback is not doing something for someone else.
- You should not rebuild a model for someone else.
- You should not type into someone’s program.
- You should ask questions of each other.
- You should share your ideas and show your own programming, explaining why and how you did something.
- You should be encouraging and helpful to others and not provide negative or mean comments.

You may use the following procedure, depending on your teacher’s preference.

- Team A provides feedback while Team B takes notes in their journal. Then teams can switch roles.
- You should start with something you really like.
- Then, if someone sees something confusing or that could be improved, let the other team/person know what is confusing or could be improved in a kind way.
- Ask the group to show their models working. Provide feedback to the team as needed if the model is not working correctly.
    - For example, if some models turn in circles or veer to one side, suggest how to modify the model and/or the program.
    - For example, provide ideas to the group about ways to speed up or slow down the model.

Culminating Project: Choose Your Own Adventure

There are several challenges from which you will choose only one for the culminating project. You can work in groups of two or in a small group of two teams (four students). The rubric for the Culminating Project is provided later in this guide. Refer to it often as you prepare the final project.

The Culminating Project consists of four phases, which allow for teacher review and support and peer review and support. It provides you with feedback so your solution will be top-notch.

At the conclusion of the Culminating Project, you will reflect and complete a self-assessment and a team assessment. These are provided later in this guide.

Culminating Project: The Four Phases

Challenge Phase 1

Teacher check and approval as soon as team is ready.

- Research several topics
- Identify two or three possible problems to solve within the challenge.
- Select one problem to solve within the challenge.
- Brainstorm solutions for the problem chosen.
- Consider human/computer interactions – impact on society – ethical; physical haptics/design of their proposed solution.
- Determine the solution is doable within time frame allowed.
- Write a clear statement of the problem, proposed idea for a solution, and the impact on society in their journals.

Teacher check and approval prior to going forward

Challenge Phase 2

Peer review should be done as soon as a team is ready for it.

- Model plan including where using sensors and motors
- Model plan including where to use subcomponents
- Model plan including how and when to use sounds and lights
- Sketches of where to use components and overall design
- Description of solution and model concept
- Pseudocode for the solution
- **Peer review - based on rubric**

Challenge Phase 3

Create the model and the program.

- While working on solution, your teacher should ask to see:
- Previous/similar design/lessons where they have used
- Models and initial programming
- Pseudocode for possible program (determine software/hardware interactivity)
- Considerations of human/computer interactions – impact on society – ethical; physical haptics design
- Iterations of design and programing

Challenge Phase 4

Presentations & Assessment

- Presentation of Completed Solution to the Class
- Present completed solution to class
- Teacher and team check against rubric
- Peer review
- Team self-assessment
- Individual self-assessment
- Clean up including all models taken apart and a full inventory of all sets

Culminating Project – The Challenges

Challenge 1

Natural Resources - Environmental Clean Up

You will research problems in the environment such as, but not limited to, recycling, environmental cleanup, or reducing the use of limited resources. You choose an issue and a location for an environmental problem to solve. You create a model and present how it works to solve the problem. You determine how well their solution works and the social impact of their solution.

Challenge 2

Transportation

Students choose A, B, C, or D in this area.

A. Transportation – Delivering to Specific Locations

You will research problems of transportation in a specific location – could be inside a building or outside in a limited area. You choose an issue and a location for their transportation problem to solve. You create a model and present how it works to solve the problem. You determine how well their solution works and the social impact of their solution.

B. Transportation of Materials on the Moon

You are part of a team devising new transportation vehicles for moving materials on the moon. The vehicle will need to be able to move in, out, and around craters, be seen from towers, and carry a load of either people or materials. You determine how well their solution works and the environmental impact of their solution.

C. Transportation of Materials in the Ocean

The ocean, a place for exploration. You work for a company that has won the bid to create a vehicle that can handle the pressure of the depths, go into shark infested waters, or navigate sensitive ecosystems. The vehicle will also have to grab items to bring to the surface for study, show video real-time, and not bump into anything, especially coral reefs, other equipment in the ocean, or the bottom of the ocean. You determine how well their solution works and the environmental impact of their solution.

D. Autonomous Transportation of Humans on Earth

You have landed a job on the autonomous vehicle team! This team is creating an autonomous vehicle that can safely carry people from place to place, obeying all traffic laws, parking safely, and letting people know when the battery needs to be charged. Remember you must discuss and take into consideration people getting in and out of the vehicle, decisions that could involve evaluating choices for crashes, and visibility on the road. You determine how well their solution works and the environmental impact of their solution.

Challenge 3

Health Science and STEM: Game for Mental Agility

You research problems of mental agility by helping people keep the brain active and learning new things. You choose ways to help people stay mentally agile and create a game that incorporates multiple tasks. The game can be any type but must meet the requirements in the rubric. You create a model and present how it works to solve the problem. You determine how well their solution works and the social impact of their solution.

Challenge 4

Manufacturing

A novel item is being created by the company for which the you work. You will need to create an item, be able to gather and stack the items, place a stack in each location for warehouse/storage, then take from storage to shipping.

Challenge 5

Government and Public Administration and Law, Public Safety, Corrections & Security - Helping People in Times of Natural Disaster

What disasters occur in your area - earthquakes, tornadoes, hurricanes, large scale fires, blizzards? You choose one type of disaster and determine how to create an alert system to help keep people out of harm's way and a system to find people that are trapped or to clean up areas affected by the disaster.

Challenge 6

Agriculture

There are lots of tasks that require manual labor in agriculture. Think about one of these tasks and create a way to automate it. Each student should choose his/her own specific challenge and the solution.

Culminating Project: Requirements

Constraints for all challenges:

- Clear explanation of the issue and the solution
- Model must contain and use one or more sensors.
- Model must contain and use one or more motors.
- Model uses lights on the hub for communication.
- Model uses sound for communication.
- Use subcomponents within the programming.

Presentation requirements:

- Explain the process used to determine the chosen solution.
- Explain societal implications of the solution.

- Explain how easy the solution is to use – what are any limitations (physical)
- Present the solution model and how it works.
- Present how well the solution works and what improvements you would foresee in the future.
- Recommendations from peer review incorporated into the solution.
- Credit given to others for any original IP that was utilized to get the solution.

Culminating Project: Assessment

Peer Review
Complete a peer review after all teams have determined a problem and possible solution. The same peer review will be completed at the end of the project as well – to show growth and incorporation of feedback. A form to help with peer review is provided later in the guide.

After the solution ideas have been determined, complete a peer review of all projects. Each project should receive at least two responses from classmates and students should incorporate the feedback into their final project.

Review the following questions to help other teams when thinking about the solution:

- Did you understand what the solution is supposed to do? (Clarify)
- What features does the solution have? Does the project seem as if it will work as expected? (Features)
- How engaging is the solution? Is it interactive, original, fun, interesting? Who would use it/interact with it? (Appeal)

Self-Assessment
Your class may choose to use numbers to help with assessment.
3 – I gave and accepted graciously feedback to my partner and other teams. I allowed my partner to have time to build, program, explain, and document.
2 – I gave feedback graciously to others but had a bit of difficulty accepting feedback from others. I did not always give my partner equal time with building, programming, explaining, or documenting.
1 – I gave little feedback to others; I accepted little feedback from others.

Team Assessment
Your class may choose to numbers to help with assessment.
3 – Our team worked like a well together. We gave and accepted graciously feedback from each other and from other teams. We made sure each person played a significant role in solving the problem and creating the solution.
2 – Our team worked fairly well together. We had a bit of trouble now and then giving and receiving feedback from each other and from other teams. We generally made sure people were involved or engaged.
1 – Our team struggled. We had a hard time giving and receiving feedback, and the work was not shared equally. Some member(s) monopolized the time together and insisted their ideas were best.

Culminating Project Rubric

Both the teacher and the team should complete the rubric, assessing the solution and presentation.

Culminating Project Rubric

Both the teacher and the team should complete the rubric, assessing the solution and presentation.

Planning	**3**	**2**	**1**
Developed a Plan and Diligently Worked Toward a Successful Solution (Fostering an Inclusive Computing Culture)	Created a well-documented plan and the team worked effectively toward a solution. Documentation contains all the following: initial ideas from brainstorming, chosen topic, notes from research, initial idea for model and how it will work, daily notes on progress, issues that need to be resolved during process.	Created a plan and worked toward the solution. Documentation contains most of the following: initial ideas from brainstorming, chosen topic, notes from research, initial idea for model and how it will work, daily notes on progress, issues that need to be resolved during process.	Did not have a complete plan and team worked occasionally toward a solution. Documentation contains a little of the following: initial ideas from brainstorming, chosen topic, notes from research, initial idea for model and how it will work, daily notes on progress, issues that need to be resolved during process.
Programming (Testing and Refining Computational Artifacts)	**3**	**2**	**1**
Includes purposeful use of lights.	Uses lights in an original way.	Uses lights directly as done in a lesson.	Does not use lights.
Includes purposeful use of sounds.	Uses sound in an original way.	Uses sound directly as done in a lesson.	Does not use sound.
Written efficiently Includes subcomponents and My Blocks.	Uses several subcomponents and My Blocks.	Uses one My Block as subcomponent	Does not use subcomponents or My Blocks.
Uses sequences and loops.	Uses multiple sequences and nested loops.	Uses a sequence and a loop.	Does not use a loop.

Uses conditional statements.	Uses multiple conditional statements.	Uses one conditional statement.	Does not use a conditional statement.
Can justify the appropriate tools and techniques.	Clear explanation of why programming was chosen and used.	Explanation is satisfactory but needs refinement.	Explanation is difficult to follow and shows a lack of understanding.
Can explain the program and its functions.	Clear explanation of how programming was chosen and used.	Explanation is satisfactory but needs refinement.	Explanation is difficult to follow and shows a lack of understanding.
Engineering Design (Testing and Refining Computational Artifacts)	**3**	**2**	**1**
Model is fully functional with purposeful use of motors.	Purposeful use of motors	Motor used.	No motor used.
Model is fully functional with purposeful use of sensors.	Purposeful use of multiple sensors	Sensor used.	No sensor used.
Can explain how the team iterated possible solutions as they used the design process.	Clear explanation of how the model idea was chosen and the iterations needed to get to the final version.	Some explanation of how the model idea was chosen, but only a few examples of iterations between initial idea and final version.	No clear explanation of how model idea was chosen and no iteration notes between idea and final model.
Can explain how feedback was incorporated into the design.	Clear explanation of how final model was tested with feedback noted and how feedback was used to improve.	Some explanation of how final model was tested with feedback notes, but vague information on how feedback was incorporated.	Little to no explanation of feedback received or how it was incorporated.
Documentation (Creating Computational Artifacts)	**3**	**2**	**1**
Followed the process for solving a problem.	Identified the problem; brainstormed several possible solutions; tested models; iterated model design and programming;	Identified the problem, thought of a solution, built a model, and made changes,	Identified a problem, built and programmed a model.

	analyzed the designs and programming; reviewed feedback; chose the best solution based on criteria.	programmed, and debugged the solution.	
Created computational artifacts.	Chose appropriate tools; decomposed the problem; coded with the design in mind; debugged the program; improved on the original design.	Choose appropriate tools, built and coded the model, debugged the program, made the model work.	Made a model and programmed it.
Selected a variety of formats to communicate ideas.	Used screen shots, video, photos, journaling, and models to communicate throughout the project.	Used photos and journaling to communicate during but mostly at the end of the project.	Made some journal entries and took a few photos at the end of the project.
Collaboration and Feedback (Collaborating Around Computing)	**3**	**2**	**1**
Used the feedback form for sharing ideas with at least 2 other groups.	Useful feedback, both positive and inspiring were provided to other teams.	Feedback was given, but minimal.	No feedback was given.
Gave an example of useful feedback shared with another team.	Feedback shows thought and was helpful to another team and could have led to improvement.	Feedback was given but was minimal.	No feedback was given.
Completed the B/Y/V feedback form for self-assessment.	Feedback was honest and thoughtful, providing both positive comments and areas where improvement could be made.	Feedback was minimal with little thought being made for what could be done in the future and what was done well.	No feedback was given.
Completed the B/Y/V feedback form for team assessment.	Feedback was honest and thoughtful, providing both positive comments and areas where improvement could be made.	Feedback was minimal with little thought being made for what could be done in the future and what was done well.	No feedback was given.

Presentation (Communicating About Computing, Collaborating Around Computing, Recognizing and Defining Computational Problems)	**3**	**2**	**1**
All team members participated. (Collaborating Around Computing)	All members participated fully.	All or most participated but dominated by one person.	Only one person presented or completed a majority of the work or no presentation occurred.
All team members answered questions. (Collaborating Around Computing)	All members participated fully.	All or most participated but dominated by one person.	Only one person presented, or no presentation occurred.
The problem was clearly identified. (Recognizing and Defining Computational Problems)	The problem was clearly identified and explanation of how the problem was determined through brainstorming and narrowing of options.	The problem was identified.	The problem was never clearly identified.
The solution solved the problem. (Collaborating Around Computing)	The solution solved the problem in an original manner through a process of iteration that was explained.	The final solution solved the problem.	The solution did not solve the problem.
The physical model was programmed and demonstrated as part of the solution.	The model demonstrated the solution and an explanation of how each part of the solution, including the model and the programming, work together to create the final solution.	The model demonstrated the solution.	The model did not demonstrate a solution.

The audience was taken into consideration when presenting.	The audience was taken into consideration and explanations were clear and thoughtful, giving necessary background, processes, and explanations.	The audience was generally taken into consideration, but some aspects such as adequate background were missing.	The audience was tangential to the presentation. Clear explanations for someone not familiar with the issue were missing.

Foundations of Physical Computing Glossary

Glossary

A

Accessibility – the design of products that take in account the ability of a variety of users who may have physical or technological limits

Array – an ordered series of items, especially numbers; In Scratch programming an array is called a list

Arc – part of the circumference of a circle

Arc turn – a turn made by both wheels turning in the same direction, but one wheel turning faster than the other

Ascending – going upward

Automate – having a machine function to reduce human work

Autonomous – able to control its movements (a robot that is programmed to run without human intervention)

B

Boolean logic – a form of algebra where all values are true or false, 0 or 1

Boolean expression – an expression that can be evaluated as true or false

C

Categorize – sort into groups based on similar characteristics

Circumference – the outer edge of a circle or wheel

Clarify – to make clear

Clockwise – moving from the top to the right, then down, then to the left, then up to the top (start)

CNC – Computer Numerical Control – controlling a machine via a computer

Complexity – state of being complicated

Compound conditional – nested *if* statements within a program requiring an expression to be evaluated as true to move to the next *if* statement.

Conditional – requirement that must be met to make a true statement

Constraint – a restriction or limitation

Counterclockwise – moving from the top to the left, then down, then to the right, then up to the top (start)

Creativity – the use of imagination or new ideas

D

Debugging – fixing the errors in an algorithm, program or physical model

Descending – going downward

Distance – the amount of space from one point to another

Distance sensor – a sensor using sound to determine the distance to an object. Also called an ultrasonic sensor

E

Effective – how well something works

Efficiency – how well something works with the least waste of energy/effort/extra code

Engineering Design Process – a process to create and build a new model; Steps include define a problem, research the problem, brainstorm possible solutions, select the most promising solution, construct a prototype, test and evaluate the prototype, communicate, redesign

Entrepreneur – a person who creates and runs a new business

Evaluate – to determine the amount, number, of value of something; For example, to determine if a statement is true or false

F

Feedback – information given or received about a product or performance used to help make improvements

Force sensor – a sensor that can measure the amount of energy used to press it

Frame – a LEGO part that has a rectangular edge and is hollow in the center

Function – a piece of code you can call (use) over and over again

G

Gyro sensor – a sensor that can determine the direction/location in space of the hub – facing up, facing down, turned right, turned left, and so forth

H

Hardware – the physical parts of a computer or physical computing components

Hub – the SPIKE Prime brain that controls the inputs and outputs of the program

I - L

Incline – slope, rise, fall, pitch generally of a surface like a ramp

Iteration – to try something over and over again, often making small changes between trials; a repetitive command made with programming loops

Kinetic Energy – the energy something has because it is in motion

List – an ordered series of items, especially numbers; In Scratch programming an array is called a list

M

Manufacturing – the making of things using machinery

Mass – a measure of the amount of matter in an object

My Block – In Scratch, a block that can represent many lines of code so the one block will replace many, making a program much more efficient

P

Parameter – a factor that sets a condition

Pitch – the up and down movement For example the nose of an aircraft going up or down

Pivot – to move around a point that is on one side of an object – example, in basketball, to move one foot while the other remains stationary

Pivot turn – a turn made by moving one wheel on a robot while the other wheel does not move

Point – a fixed location

Point turn – a turn made by moving one wheel forward while the other wheel moves backward

Potential Energy – the energy something has because of its position

Prosthesis – an artificial body part

Prototype – a first model from which changes will be made to produce other models

Pseudocode – writing in words what you want a coded program to do

R

Rapid prototyping – quickly and continuously making small changes to models to improve performance

Repetitive – doing the same thing over and over

Robot – a machine that can follow commands programmed by a human

Roll – turning over and over on an axis For example, the wings of an aircraft moving down to the right or left so that the plane tilts and could rotate in a circle if the tilting continues

Rotation – the number of times a wheel moves in a complete circle equivalent to the circumference of the outer edge

S

Sensor – a device that detects or measures physical properties such as color, distance, location, and so forth

Software – programs or coding used by a computer

Subcomponent – smaller parts of a larger unit

T - Y

Traverse – travel across or through

Ultrasonic sensor – a sensor using sound to determine the distance to an object

USB - Universal Serial Bus – most common type of computer port which can connect keyboards, mice, robot hubs, etc., to computers

Variable - an element that can change; a placeholder for something that can change

Velocity – the speed of something in a given direction (vector)

Yaw – moving around a vertical axis For example, the nose of an aircraft moving from side to side

Introduction to Robotics

Grade 6-8 | **45 minutes** | **Beginner**

Introduction to Robotics

Students will define the criteria for teamwork, get introduced to the SPIKE Prime hardware, and learn about the final project.

Questions to Investigate

- How can technology help us solve problems?
- How do engineers plan for the use of hardware in a solution to a problem?
- In what ways can I be an effective team member to finish a project on time?

Materials Needed

- SPIKE Prime sets ready for student use. Prior to the first lesson, please visit the following website for help with set up, kit organization and SPIKE App. https://education.lego.com/en-us/start/spike-prime/intro
- Student journal

Prepare

- Determine how students will be grouped (2 students work with one SPIKE Prime set).
- Determine your student expectations for teamwork.

1. Engage

Introduce the Unit

Engage students in a conversation:

- What do you know about robotics?
- What about robotics interests you?
- What would you like to learn about robotics?

KEY OBJECTIVES

Students will:

- Describe characteristics of working on a team
- Describe the function of hardware
- Identify hardware components

STANDARDS

CSTA

2-AP-18 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.

2-CS-02 Design projects that combine hardware and software components to collect and exchange data.

VOCABULARY

Hardware

- Explain to students for this unit, they will be learning about different ways we can use technology to help us solve problems through the lens of robotics.

Culminating activity for the unit

Explain that over the course of the next several weeks, students will be working in teams to explore hardware and software relationships, design processes, and documentation ideas through building and programming prototypes.

At the end of the unit, their team will select one of these projects to design, build, test, and present a possible solution using robotics.

Teamwork

Explain to students that many of the activities in this unit require teamwork.

Ask students:

- What are some benefits of working as a team? (i.e., real-world practice, efficiency, sharing of ideas, distribution of tasks to finish on time).

Share with students your criteria for what effective teamwork looks like and sounds like. You may decide to ask students to contribute ideas as well. Students will reflect often as to how they contributed as an effective team member throughout the course.

Explain to students that much of the work over the next few weeks will be project-based. Time management and materials management are important part of working on a project together.

Ask students:

- What are some ways a team can effectively manage their time?

Place students in teams of two. Teams should remain the same over the course of the unit.

Hardware

Explain to students that they will be using the SPIKE Prime set for the robotics unit.

Ask students:

- What other types of material do you think engineers use when designing a robot? (i.e., screws, bolts, specially designed plastic pieces, wires, motors, sensors, motherboard)
- Tell students hardware is what you can touch. Software are the applications or apps that make the hardware do things.

Explain to student for this unit, materials will include LEGO® bricks and elements. Distribute the SPIKE Prime sets to each team of students.

2. Explore

Have students open the box to investigate the types of elements and bricks contained in the set.

Ask each student to select one brick or element from the set. Have them share with their partner what they think that brick or element is used for. You may ask two or three students to share with the entire class.

Have students look in their sets to find each of the following:

- A rubber band
- A 2 x 4 brick of any color
- A black connector peg
- A medium blue frame
- A tire
- An 8-tooth gear
- A 5-hole beam

Next, ask students to work with their partner to find a brick to match the function listed below:

- Allows a design to move
- Secures a piece in place
- Joins pieces together at an angle
- Decorates
- Connects pieces
- Reduces build time
- Controlled by software

3. Explain

Have students share out bricks or elements they matched with each function:

- Allows a design to move (gear, tire, rubber band, motor, tube)
- Secures a piece in place (bushings, pins)
- Joins pieces together at an angle (couplers, biscuits)
- Decorates (minifigs, eyes)
- Connects pieces (axle rods, pins, bricks)
- Reduces build time (technic plate, technic frames)
- Controlled by software (hub, motor, sensors)
- Show students the area on the back of the lid insert that shows how to measure the length of an axle. Then ask students to locate an axle with a length of 5.

Ask students:

- Why do you think it is important to know the names of the pieces when working with the set?
- Why do you think it is important to understand the function of different LEGO® bricks and elements when designing a solution?
- What resources in the kit might help with organization of materials?

Have students place LEGO® bricks and elements into the correct location in the trays.

4. Elaborate

BRICKGO

Have students create a large 5 x 5 grid in their journals to play BRICKGO, a type of bingo game.

Teacher selects 30 different LEGO® elements and has each team find the pieces from their sets.

- Have teams randomly place an element in each square of the BRICKGO board.

The teacher randomly gives the name and/or description/function of a LEGO® element from the selected group of pieces.

- Have teams look to see if the piece is on their BRICKGO board. If so, the team removes it.
- Continue playing until a team has BRICKGO (one row, one column, or one diagonal completely empty on their card).

5. Evaluate

Teacher Assessment

Evaluate students' understanding of element names.
Evaluate students' understanding of how technology can help us solve a problem.

Self-Assessment

Tell students that they will be responsible for materials management. Parts should not be shared between sets. If a part is missing, the teacher will have limited spare parts. Let the teacher know immediately if you cannot locate something. Every day, students will make a journal entry about the materials management of their set. They will grade themselves using a three-point scale.

1) Materials are not all located in their correct tray; some parts are still together.

2) Materials are located correctly, but only one person helped put things away.
3) Both partners worked together and all parts in the correct locations.

Have students answer the following in their journals:

- What is the name of a LEGO® brick or element you found interesting today? What is its function?
- What characteristics of a good teammate did you display today?
- Rate yourself on a scale of 1-3 on your time management today.
- Rate yourself on a scale of 1-3 on your materials (parts) management today.

LEGO® Education SPIKE™ Prime

What is a Robot?

Grade 6-8 | **45 minutes** | **Beginner**

What is a Robot?

Define a robot and how software and hardware work together.

Questions to Investigate

- What is a robot?
- How can technology help us solve problems?
- How do engineers plan for the use of hardware in a solution to a problem?

Materials Needed

- SPIKE Prime sets ready for student use. Prior to the first lesson, please visit the following website for help with set up, kit organization and software app. https://education.lego.com/en-us/start/spike-prime/intro
- Device with SPIKE App installed
- Images or video clips of different robots
- Student journal
- Sticky notes

Prepare

- Check to make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.

KEY OBJECTIVES

Students will:

- Define a robot
- Identify ways we use robotics to solve problems
- Describe the function of hardware and software
- Identify hardware components

STANDARDS

CSTA

2-AP-18 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts

2-CS-02 Design projects that combine hardware and software components to collect and exchange data.

VOCABULARY

Hub
Robot
Software
USB

1. Engage

What is a robot?

Launch a discussion about robots. Ask students the following questions:

- What is a robot?
- Where might I find a robot? (i.e., manufacturing, storage, at home, e.g., programmable vacuum cleaners and lawnmowers)
- How do these robots help us solve a problem? (i.e., they are fast, accurate, complete repetitive tasks)

Ask students to think of three words that describe a robot, then write words on a piece of paper or sticky note.

> **Note:** Criteria for becoming a robot is suggested at the end of the session. Encourage students to discuss questions like the following:
>
> - Is it powered?
> - Is it programmable?
> - Can it sense things?
> - Can it move?
> - Can it complete a task automatically?

Have students share their three words with three different classmates. Allow students to add or remove words from their list.

Show students images, video clips, or different robots and ask students to think about the words they have collected on their list. Allow students to modify the list.

Using a graphic organizer (in appendix) and their word list, have teams create a definition of a robot, providing examples and non-examples based on their definition.

Have students test their definition by determining if some everyday objects could be defined as a robot based on their definition. These objects could include, but are not limited to a:

- Washing machine
- Toaster
- Television
- Cell phone
- Video game system

Have students make modifications to their definitions. Ask students to think about the pieces and parts that might make up a robot or a robotic system.

Distribute the SPIKE Prime Sets and devices with SPIKE App. Have students select the hub and USB cord (if not connecting Bluetooth).

Tell the students they need to investigate the following:

- What is the function of the hub?
- How does the hub and SPIKE App work together?
- Is the hub a type of robot?

2. Explore

Have students complete SPIKE Prime **Getting Started 1 Start Here** lesson.

Suggest some simple troubleshooting tips if students are having difficulty connecting (i.e., is the device powered, is the device connected properly, does the device software need an update, etc.). Allow students to have 3-5 minutes to change the light pattern and the sounds in an exploration.

3. Explain

Ask students the following questions:

- What do you think is the function of the hub?
- How do the hub and SPIKE App work together?
- Based on your definition, is SPIKE Prime a robot?
    - Is it powered?
    - Is it programmable?
    - Can it sense things?
    - Can it move?
    - Can it complete a task automatically?

Challenge students to think about what they would add to the hub to build a robot to fit their definition of a robot (i.e., additional sensors, motors).

Have a discussion on pair programming. What are some ways the students should work together to ensure both partners have a chance to program, debug, etc.? Ask students why this is important.

- Each team should have a driver and a navigator. The students will switch roles several times during each class.
- A “driver” is thinking about the programming needed and using the computer to type in the code.
- A “navigator” helps direct the driver, making sure all the code is correctly written – the ports are correct, the operators are precise, and so forth.
- Think of it as the driver holds and moves the steering wheel on the car, while the navigator has the map and tells the driver when and where to turn.
- Switch roles several times within a class period.

4. Elaborate

Have students explore ways to change the display on the hub. Ask students to synchronize the lights and the sounds to create a 10-15 second sound and light show. Remember to have them pair program! Have teams share their sound and light show with the class.

5. Evaluate

Teacher Assessment

Evaluate students’ understanding of how a robot helps us solve a problem.

Self-Assessment

Have students answer the following in their journals:

- What hardware and software did you use for your project today?
- What different things did you try to create a sound and light show?
- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on time management today.
- Using a scale of 1-3, rate yourself on materials (parts) management today.

Definition	Characteristics
Robot	
Examples	Non-Examples

Notes:

LEGO® Education SPIKE™ Prime

Sound and Light

Grade 6-8 | **45 minutes** | **Beginner**

Sounds and Light

Get an introduction to SPIKE Prime parts and program.

Questions to Investigate

- Why would engineers use sound and light on a robot?
- Why would engineers use sensors and motors?
- What are ways engineers could change the parameters for sound, light, and motors?

Materials Needed

- SPIKE Prime Sets
- Device with SPIKE App installed

Prepare

- Check to make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.

1. Engage

Ignite a discussion. Divide the class into three groups. Ask one group to sit, another group to stand, and the third group to kneel. Tell students they are robots and they need to execute a “sing tone” function based on the parameter you give to them. A **function** is a piece of code you can all over and over again. A **parameter** is a type of variable. It is what changes based on what the code asks to be done.

The parameters are the tones given to the Kneelers, Standers, and Sitters.

Now have the students that are kneeling make a low tone, the ones that are sitting make a bit higher tone, and the ones standing to make a very high sound.

- Ask them to make the sounds in an ascending order. What order would the groups sing? Ask them to make the sounds in a descending order.

KEY OBJECTIVES

Students will:

- Explore light and sound blocks
- Create an inspirational message using sound and light
- Use pseudocode to plan prior to writing code

STANDARDS

CSTA

2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms.

2-CS-02 Design projects that combine hardware and software components to collect and exchange data.

3A-AP-13 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.

VOCABULARY

Ascending

Descending

Parameter

Ask students to explain the parameters and how they were used. Ask students when it would be useful for a robot to make ascending or descending tones. What types of robots would use tones?

2. Explore

Students will write pseudocode and then program the hub to play sound and show shape patterns.

Ask students to choose **New Project.** Connect the hub to the SPIKE App.

- Suggest some simple troubleshooting tips if students are having difficulty connecting (i.e., Is the device powered? Connected properly? Does the device software need to be updated? Are you using the correct programming block?)

Have students create a program to play a sound and show a shape pattern and test their program.

Ask students if they are ready to try a musical challenge.

Ask students to make a short song and make the lights show the letters of their names. Prior to creating programs in the SPIKE App, ask students to write a pseudocode. Pseudocode is writing in words what you want the program to do.

First, have students write the pseudocode in their journals that show the letters of their names. Before they write, have them compare the following examples. Ask students which would be more helpful when writing and checking code.

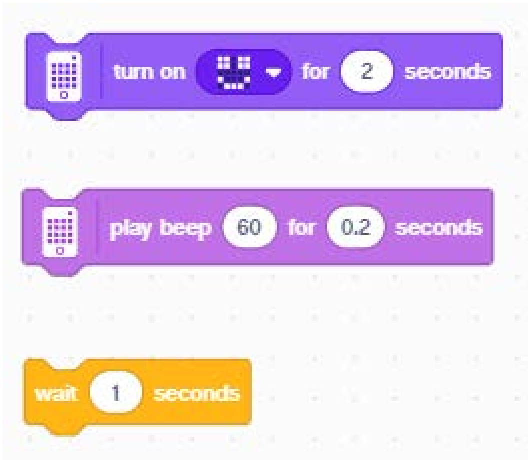
The following example is a beginning pseudocode:

- Show K, play 3 sounds
- Show E, play 3 sounds
- Show L, play 3 sounds

The following example is a more detailed pseudocode:

- Show K, play 3 ascending sounds for .04 seconds each
- Show E, play 3 descending sounds for .04 seconds each
- Show L, play 3 much higher sounds for .04 second each
- Complete your program and debug as needed.

Model for students how to use the “turn on for seconds” block from the Light palette, the “play beep for seconds” block from the Sound palette, and the wait for seconds from the Control palette. Allow students to write a program that they believes matches their pseudocode.



Have students play the program and debug as needed.

Ask students to share their model and program with at least two other teams.

3. Explain

Ask students the following:

- What was easy and what was hard about the last activity?
- What parameters did they use for light and sound?
- What could and could not be done with the three blocks given?
- What did you like about another team's program?

Ask students to think about the challenges at the end of the unit – environment, transportation, and games. Discuss ideas on how using lights or sounds could be part of any solution.

4. Elaborate

Students will create an inspirational message for the class.

Ask students to consider messages that inspire people to do well. Tell them each team will write pseudocode for a program that will share an inspirational message. Each program must contain sound and lights. Students may experiment with any block in the Light and Sound palettes.

Allow students time to share their messages with the entire class or at least two other groups if time is limited.

5. Evaluate

Teacher Assessment

Evaluate the students' understanding of:

- How the programming and the physical model must work together.
- Why engineers would use sound and light on a robot.

- How sound and light can be used to deliver a message.
- Using pseudocode to help with the planning process for writing code.

Ask students what challenges they encountered when writing code.

Self-Assessment

Have students answer the following in their journals:

- What was the coolest thing you discovered today while exploring?
- How could you use motors and sensors in the solution to any of the challenges?
- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on time management today.
- Using a scale of 1-3, rate yourself on materials (parts) management today.

LEGO® Education SPIKE™ Prime

Motors and Sensors

Grade 6-8 | 90 minutes | Beginner

Motors and Sensors

Independently program motors and sensors.

Questions to Investigate

- How do engineers plan for the use of hardware and software in a solution to a problem?

Materials Needed

- SPIKE Prime sets
- Device with SPIKE App installed
- Student journal
- Graphic organizer from Day 1 with student notes (they will be revising again today)

Prepare

- Check to make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.

1. Engage

Ask a few students to share their definitions of a robot.

Engage students in a conversation:

- What characteristics do robots share?
- From previous experiences, is the SPIKE Prime hub without attachments a robot? Support your claim.

KEY OBJECTIVES

Students will:

- Describe the function of motors and sensors
- Program independently motors and sensors
- Describe how hardware and software work together

STANDARDS

CSTA

2-AP-18 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.

2-CS-02 Design projects that combine hardware and software components to collect and exchange data.

3A-AP-13 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.

VOCABULARY

Ultrasonic sensor

Gyro sensor

Force sensor

Motors and Sensors

Have students brainstorm:

- How do engineers use motors in a design to help solve a problem or make life easier? (e.g., conveyor belts to move items; robotic vacuums to move across a floor)
- How do engineers use sensors in a design to help solve a problem or make life easier? (e.g., automatic doors at stores)

Have students create a table in their journals for investigating SPIKE Prime motors and sensors. In this lesson, students will work in teams to

- Predict the use or purpose of the motor or sensor.
- Write a program that uses a motor or sensor. Observe and describe how it functions.
- Brainstorm examples of how that type of motor or sensor may be used in the real world.

Here is an example of a table students might create before they begin using the programming app:

Name	Prediction	Description	Real World Examples
Medium motor			
Large motor			
Distance Sensor			
Color Sensor			
Force Sensor			
Gyro Sensor (tilt sensor in hub)			

After students have created the table, they are ready to Explore.

2. Explore

Students complete Getting Started Motors and Sensors and explore beyond the guided practice.

Ask students to open the SPIKE App and choose Start. Then, choose **2 Motors and Sensors.** Direct students to the build section.

Ask students to build the models. Ask students the names of parts they are using as you monitor the room. When they finish building have them move to connecting to the ports. Students will be extending the given activities.

Large Motor Investigation

Ask students to follow these steps:

- Have students connect a Large Motor to Port C on the hub.
- Add blocks so that the motor starts slowly and gets faster over a total of ten rotations. Change the direction of the motor.
- Ask students how to duplicate blocks. (Right click on the block and choose duplicate.)
- Ask students to add a comment. (Right click on the block and choose add comment.)
- Ask students to explain what their current program does by writing in the comment box.

Distance Sensor Investigation

Ask students to follow these steps:

- Use the right arrow key and move to Connect the Distance Sensor to Port D. Create and run the program as designed. What happens?
- Now, move your hand so that only one side of the distance sensor can see it. What happens when you play the program now?
    - Can you tell which side of the ultrasonic sensor sends information and which side receives it?
    - Ask students why the orientation of the sensor would be important when adding it to a model.

Medium Motor and Color Sensor Investigation

Ask students to follow these steps:

- Use the right arrow key and move to Connect 2 Medium Motors and a Color Sensor.
- Find the green and the blue bricks. Change the program so the A motor will run when the color sensor senses green, and the F motor will run when the color sensor senses blue.
- Add start sound Win after the color green is sensed. Add the Boop Bing Bop sound after the color blue is sensed.
- Give real world examples of the use of a color sensor.

Gyro Sensor Investigation

Ask students to follow these steps:

- Change the program so the trigger for the actions is the gyro sensor in the brick.
- If the front is up, the motor runs clockwise, and the Dun Dunn Dunnn sound plays. If the back is up, the motor runs counterclockwise, and the Laughing Baby sound plays.
- Explain how the gyro sensor works and give real world examples of its use.

Allow students time to continue to investigate on their own.

3. Explain

Ask students the following questions:

- What is the function of the medium motor? (Can be both a motor and a sensor because the motor contains an integrated rotation sensor which can report speed and position; it can also sense direct user input if the motor is rotated by hand.)
- What is the function of the large motor? (Can be both a motor and a sensor because the motor contains an integrated rotation sensor which can report speed and position; it can also sense direct user input if the motor is rotated by hand.)
- What is the function of the distance sensor? (Can measure distance to an object or surface using ultrasonic technology.)
- What is the function of the color sensor? (Can detect color, reflectivity, or ambient light; can be used as a non-programmable light output.)
- What is the function of the force sensor? (Can detect simple touch and measure for in Newtons.)
- What is the function of the gyro or tilt sensor? (Hub contains three-axis accelerometer and three-axis gyroscope.)
- How do the hub, sensors, motors and SPIKE App work together?
- As you were programming your motors and sensors, what similarities did you notice between the SPIKE App and Scratch 3 software?
- Based on your definition, is the SPIKE Prime hub a robot?
    - Is it powered?
    - Is it programmable?
    - Can it sense things?
    - Can it move?
    - Can it complete a task automatically?

4. Elaborate

Challenge students to learn a little more about the different motors and sensors.

Direct students to:

- Go to the home page of the software.
- Select the small gear in the upper right-hand corner of the screen.
- Select help.
- Select hardware overview.

Ask students to read about the hub, motors, and each sensor. Have students document questions they may have regarding additional ways a motor or sensor could be used or how it works. Encourage students to try different ideas with their hardware.

Have students program the motors and sensors to work together in a way that has not been done yet. Tell students to be creative and to try new ideas.

Allow students to modify their definition of a robot, if desired, based on what they learned today.

Take the models apart and place the pieces into the correct tray locations.

5. Evaluate

Teacher Assessment

Evaluate students' understanding of why they might want to include a motor or a sensor in a robot design.

Self-Assessment

Have students answer the following in their journals:

- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on time management today.
- Using a scale of 1-3, rate yourself on materials (parts) management today.

LEGO® Education SPIKE™ Prime

Make It Move

Grade 6-8 | 90 minutes | Beginner

Make It Move

Investigate how changes in programming can alter the performance of a robot.

Questions to Investigate

- What is a robot?
- How do changes in a program affect connected hardware?
- Why do engineers document changes to a design solution?

Materials Needed

- SPIKE Prime sets
- Device with SPIKE App installed.
- Student journal
- Graphic organizer from Day 1 with student notes from Day 2 (they will be revising again today)

Prepare

- Check to make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.

KEY OBJECTIVES

Students will:

- Define prototype
- Investigate how changes in software can alter the performance of a robot
- Develop a system for documenting changes made to a design solution
- Make a claim if the SPIKE Prime Hopper is a robot

STANDARDS

CSTA

2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.

2-AP-19 Document programs in order to make them easier to follow, test, and debug.

2-CS-02 Design projects that combine hardware and software components to collect and exchange data.

VOCABULARY

Prototype

1. Engage

Review

Ask a few students to share their observations on how sensors and motors function.

Engage students in a conversation. From the previous activity, did adding motors and sensors to the SPIKE Prime hub change your claim or change your definition of a robot?

Lesson introduction

Distribute the SPIKE Prime Sets and devices with SPIKE App. Ask students to locate the following pieces from their set:

- 1 blue 2 x 4 brick
- 1 green 2 x 4 brick
- 1 red 2 x 4 brick
- 1 yellow 2 x 4 brick

Have students use this pseudocode to create a stack of bricks.

- Place the red brick in front of you.
- Connect the blue brick on top of the red brick so no red studs are showing.
- Connect the green brick on top of the blue brick so no blue studs are showing.
- Connect the yellow brick on top of the green brick so no green studs are showing.

Ask students to describe how to build the model.

Explain to students that we are going to investigate how a change in the pseudocode changes the build. Note how this example also documents a change in the program. Strikeout (~~make a line through the words you want to delete~~) then add the words needed to correct the code. Consider modeling this method for students.

- Place the red brick in front of you.
- Connect the blue brick on top of the red brick so no red studs are showing.
- Connect the yellow brick on top of the blue brick so no blue studs are showing.
- Connect the green brick on top of the yellow brick so no yellow studs are showing.

Ask students:

- Describe the build by listing the order of colors in the stack (i.e., red, blue, yellow, green).
- Identify how the change in the pseudocode affected the order of the colors in the stack of bricks.
- Why do you think it was important to document the change made?
- One way to document changes in pseudocode is to use a strikeout (drawing a line through what I want to remove) and then add new text. What are some other ways I can document changes made to a design?

2. Explore

Students will build a prototype of a hopper robot in order to investigate how a change in a program affects connected hardware. Share with students that a prototype is an early model of a design that is often tested then modified. Students are practicing ways to document changes in their journals.

Show students how programs can be saved to different location within the hub by using the button to the right of the large circle. Twenty locations are available, so 20 programs can be saved at one time. Explain the following steps to students:

- Open the SPIKE App.

- Turn on the hub.
- Open a new program.
- Connect the hub.
- In the lower right corner of the screen, there is a 5x5 grid that is gray and it shows a 0. Click the grid.
- A window pops up that shows download and streaming along with the grid inside a drawing of a hub.
- Click the arrow to the right to change the number from 0 to 1. Repeat moving right or left to choose the location on the hub to place a program.
- Write a short program with lights and/or sound.
- Download the program to location 2.
- Use the right button on the hub to move to location 2. A number 2 will show on the 5x5 grid in lights.
- Press the large circle and your program will play.

Direct students to build and program the Hopper model by completing **Getting Started 3 Make It Move**.

- Suggest some simple troubleshooting tips if students are having difficulty connecting (i.e., is the device powered, is the device connected properly, does the device software need an update, are you using the correct programming block, etc.)
- When students test the prototype in the lesson, they will need to adjust the program blocks and rerun the program.
- Ask students to document any changes to the program they made in their journal and what the outcome was from the change made.

Allow students time to explore and investigate the hopper program.

3. Explain

Discuss the Hopper model and program with students.
Ask students the following questions:

- What block type did you need to change? (movement)
- What happened when you made a change to the program for the Hopper robot?
- How do the movement blocks in the program affect motors plugged into the hub?
- Based on your definition, is the SPIKE Prime Hopper a robot?
    - For example,
        - Is it powered?
        - Is it programmable?
        - Can it sense things?

- Can it move?
- Can it complete a task automatically?

4. Elaborate

Challenge students to investigate further. Discuss different ways that the Hopper model can move. Ask students to consider how to program the model in the following ways.

- Program the Hopper to move slower.
- Program the Hopper to move backwards.

Allow students to modify their definition of a robot, if desired, based on what they learned today.

5. Evaluate

Teacher Assessment

Evaluate students' understanding of what a robot is.
Evaluate students' understanding of how motors and sensors can work together.

Self-Assessment

In their journals, have students answer these questions:

- What is a benefit of documenting changes made to a program?
- What were two changes you made to the hopper program?
- How did those changes affect the movement of the hopper?
- What was the most surprising thing you learned today?
- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on time management today.
- Using a scale of 1-3, rate yourself on materials (parts) management today.

LEGO® Education SPIKE™ Prime

Connecting to Careers: Human Services and Education & Training

Grades 6-8 | 240 minutes | Beginner

Connecting to Careers: Human Services and Education & Training

In this lesson series, students will have the opportunity to explore and research careers.

Prepare

Prior to starting the lesson, prepare the following:

- Students will need access to the SPIKE Prime set.
- Make sure you have enough devices and access to the Internet for student use during this lesson.
- Make copies of the handouts (if desired) or place into a digital platform for student use.
- Prepare images of different jobs. There should be a variety of images to show different jobs within the 16 career clusters. (Example for Architecture & Construction: Architectural drafter, engineer, carpenter, engineer, electrician, HVAC technician, painter, environmental designer, etc.).
- Optional materials: Creative Brick Set, markers, craft materials

Vocabulary: Career, Career Cluster, Qualifications, Skills, Education, Knowledge

KEY OBJECTIVES

Students will:

- Articulate their personal interests and goals.
- Relate their personal interests, skills, and goals into possible career pathways.
- Explore various careers in career pathways.

STANDARDS

Career Ready Practice 10 - Plan education and career path aligned to personal goals. (CCTC)

1. Engage

Ignite a discussion with students:

- How do people decide the career they want to pursue?
    - For example, if a someone wants to be a teacher - what kind of interests might they have? What kind of skills would they need to have to be successful in this job? What goals would they need to set to become a teacher? What other jobs support being a teacher? (Custodians, principals, payroll, accounts payable, food service, bus drivers, and so forth)

- Present the images of the jobs to students. As a class, decide how you would categorize the jobs. Ask questions like:
  - What jobs belong together?
  - What kind of similar skills do these different jobs use?
  - What kind of environments are associated with these jobs?
  - Do any of these jobs interact or rely on one another? If so, how?

Ask students to think about what interests them. What kind of job(s) would they like to have in the future?

2. Explore

Note: This part of the lesson will be completed over several days. There will be two career clusters focused on and researched in each career lesson. Students will be assigned a career cluster each time career connections lessons are taught. By the end of the course, students will have explored all 16 career clusters.

Give each student one My Career Plans worksheet. Allow time for each student to complete the worksheet. The worksheet should be kept for comparison at the end of the course.

Split the students into small groups of four. Explain to students that there are thousands of different jobs with different career clusters, also known as "industry sectors."

There are 16 career clusters:

- Agriculture, Food & Natural Resources
- Architecture & Construction
- Arts, A/V Technology & Communications
- Business Management & Administration
- Education & Training
- Finance
- Government & Public Administration
- Health Science
- Hospitality & Tourism
- Human Services
- Information Technology
- Law, Public Safety, Corrections & Security
- Manufacturing
- Marketing
- Science, Technology, Engineering & Mathematics
- Transportation, Distribution & Logistics

The two career pathways they are studying today are:

- **Human Services**
- **Education & Training**

In this first unit, students have been learning about robots and have also started to learn about engineering and programming. In Education & Training, careers focus on how to help people learn new things. In Human Services, careers can include jobs where people match interests, skills, and job openings. People in Human Services may also include training people for the culture of the company they have just joined. Students may consider themselves being trained for some new opportunities that will arise during this course. Help students to correlate these careers with what they have experienced in the first lessons of Unit 1.

Ask students to brainstorm as many jobs as you can within each career cluster. Have one person write down the jobs as they are named.

In small groups of 4, students complete online research to find out more about each career clusters. Allow time to get information about the jobs that included, especially jobs they had not heard of or had not thought of in their brainstorm.
Students should also research:

- Skills needed
- Forecast of future job openings (and current)
- Certifications, licenses, apprenticeships, etc. required

In their group, students create a visual representation by building a model of one of the two career clusters. They will build one or more physical models with LEGO bricks. Each person can build a model, or the group can create one large model that represents all ideas about one of the career clusters - Human Services or Education and Training.

Each group will be responsible for a quick one-minute presentation of their LEGO model(s) to explain how it represents their career cluster.

3. Explain

When the students have finished building, allow each group to present their model. Ask students questions like:

- Tell us about your LEGO build and how it represents the career cluster.
- What kind of interests would these career clusters have?
- What are some of examples of jobs in these career clusters?
- Can you explain the difference between a job and career cluster?
- How are these career clusters like one another?

4. Elaborate

Ask students if there were any jobs in today's career cluster that were interesting to them. In small groups, have each student talk about one job they thought was interesting or that they had never heard of. What skills were associated with that job? Do they know anyone who has those skills?

Ask students what interested them about careers in Human Services and Education & Training. Ask students what skills they have that would make them a suitable candidate for a job in one of these career pathways.

5. Evaluate

Evaluate the students' skills development by observing if they:

- Articulate their personal interests, skills, and goals.
- Relate their personal interests and goals into career pathways.
- Explore various careers in career pathways.

Name: ____ Date: ________________

My Career Plans

WHO AM I?

Things I Am Interested In:	My Favorite Subjects in School:
1.	1.
2.	2.
3.	3.

Things I Am Good At:	Things I Do Not Like to Do:
1.	1.
2.	2.
3.	3.

I need to improve on ____________________ because ______________________.
1.
2.
3.

LEGO® Education SPIKE™ Prime

Iteration and Perseverance

Grade 6-8 | **90 minutes** | **Beginner**

Iteration and Perseverance

Students will create an initial design and iterate to create a working model with effective programming.

Questions to Investigate

- How do engineers improve upon a design?
- What are characteristics of specific feedback?
- How can feedback be used to improve a design?

Materials Needed

- SPIKE Prime sets
- Device with SPIKE App installed
- Student journals
- Masking tape

Prepare

- Check to make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.
- Using masking or painters' tape, create two lines that are three feet apart. One is the starting line and the other is the finish line. Students will create robots that walk from start to finish.

1. Engage

Engage students in a discussion about ways an object or animal can move (e.g., roll, crawl, slide, hop, walk, glide and so forth). Discuss how engineers use these types of movement when designing robots. Have students research robots without wheels and share ideas.

- What designs did they see that did not go straight?
- What designs did they find that seemed to work well?
- How do engineers test their designs to find which ideas work?

Ask students to share their research and any videos or images that are helpful.

KEY OBJECTIVES

Students will:

- Use an iterative process to improve the performance of a working model
- Identify characteristics of specific feedback
- Describe how feedback can be used to improve a design
- Give and use specific feedback in the design process

STANDARDS

CSTA

2-CS-03 Systematically identify and fix problems with computing devices and their components.

2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.

2-AP-17 Systematically test and refine programs using a range of test cases.

2-AP-15 Seek and incorporate feedback from team members and users to refine a solution that meets user needs.

3-AP-16 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.

3A-AP-21 Evaluate and refine computational artifacts to make them more usable and accessible

VOCABULARY

Iteration
Feedback
Rapid prototyping

2. Explore

Students will modify their hopper robots, collect data to test efficiency, and repeat modifications. Students will determine how modifications to the model and programming affect performance. Then, the students will race their modified robots.

Explain to students that they are going to build the hopper model and make improvements on the design. When engineers modify the hardware or program for a robot, making improvements along the way, they use an iterative process.

Explain that an iterative process is one where people create, test, refine based on results, test again, and continue the testing and refining or modifications until they are satisfied with the results. It is important to keep track of the changes and the outcomes in a journal so you can refer to the changes made and the effect on the model's behavior.

Have students create a table in their journal for collecting observational data for how the hopper model moves at different speeds. Be sure to include space for multiple trials at different speeds.

Speed/Power	Trial 1 Time	Trial 2 Time	Trial 3 Time
Ex. 100%			
Ex. 50%			

When students have created the table, they should find the **Hopper Race lesson** in the **Invention Squad unit**. Working as a team of two, build the hopper model. Complete **Step 03** and stop.

Ask students:

- How well does the program and the model work?
- What changes could you make on the program to make it more efficient?

Direct students to complete **Step 04** and stop.

- Modify the legs of the hopper. Students may **not** use wheels nor any element (for example, gears) to act as a wheel.

Compare the results. Write your analysis of each type of leg in your journal.\

Continue to iterate on both the model and the programming in order to create a robot without wheels that can go three feet from the starting line across the finish line. Have students take notes in their journals (and photograph if possible) about each iteration. Students should write what they change, the outcome, and when possible, the reason for the change.

Remind students that engineers often get inspiration for new designs from looking at what others have created. Looking at what other teams are doing is not cheating and can give them new ideas.

Have students race other teams in small groups or have one or two heats with all the hopper models.

3. Explain

Discuss with model and program with students. Let students share what iterations they made and how the process went.

Ask students:

- How the length of the leg affected the movement
- How friction affected movement

Let students know that this type of iteration is also called rapid prototyping – because small changes are made several times in a short period of time. Explain to students that part of the iterative process is using feedback to improve of a design.

Ask students:

- What are some examples of helpful feedback we might give to another team?
- Why do you think feedback needs to be very specific?

4. Elaborate

Students will practice giving and taking feedback to improve their model. Have students find their Peer Review page in their Student Guide.
Tell students:

- They will be providing feedback today to each other.
- Feedback needs to be focused on how to help someone see things from a new perspective, teach a new skill, or explain a concept in a different, more understandable way.
- Each person will be asked to give and receive feedback graciously.
- Each person needs to think about how to be kind and helpful in their words.

Feedback Model

Explain to students the following guidelines for giving feedback. Consider also posting the guidelines for students for reference.

- Feedback is not doing something for someone else.
- You should not rebuild a model for someone else.
- You should not type into someone's program.
- You should ask questions of each other.
- You should share your ideas and show your own programming, explaining why and how you did something.

- You should be encouraging and helpful to others and not provide negative or mean comments.

Students can provide feedback as a whole group, but we recommend having two teams work together to provide feedback to each other. Consider using the following procedure:

- Team A provides feedback while Team B takes notes in their journal. Then teams can switch roles.
- Students should start with something they really like.
- Then, if someone sees something confusing or that could be improved, let the other team/person know what is confusing or could be improved in a kind way.
- Ask the group to show their models working. Provide feedback to the team as needed if the model is not working correctly.
    - If some models turn in circles or veer to one side, suggest how to modify the model and/or the program.
    - Provide ideas to the group about ways to speed up or slow down the model.

Model providing feedback often for the class and help them learn to use positive language and not negative language when providing feedback. Also practice taking feedback and thinking about how to use it rather than becoming defensive.

Incorporating Feedback

Give students time to modify their hopper designs and program based on the feedback they received from another group.

- Now, it's time for a second race! Place multiple hoppers at the start line.
- Once ready, have all teams place their final models at the start and then begin the race. If models get tangled or veer off course, the teacher can put them back into the race. (The goal is not to have anyone touch their models between start and finish.)
- Have students discuss what behaviors they saw in successful models. Ask if students have questions for other teams about programming. Allow students 5 minutes to make minor changes and then race again.
- Have students write their observations in their journals.

5. Evaluate

Teacher Assessment

Evaluate students' understanding by iteration.

Ask students:

- How rapid prototyping and iteration was helpful in the hopper challenge.

- How they used feedback to modify their models or programs.
- If the feedback improved the outcome.
- If the robot moved as expected.

Self-Assessment

Have students answer the following in their journals:

- What was one thing you really enjoyed about today's lesson?
- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on time management today.
- Using a scale of 1-3, rate yourself on materials (parts) management today.

LEGO® Education SPIKE™ Prime

Pseudocode

Grade 6-8 | 45 minutes | Intermediate

Pseudocode

Students will read, write, and compare pseudocode.

Questions to Investigate

- How does reading and writing pseudocode help programmers with accuracy and debugging?

Materials Needed

- SPIKE Prime Sets
- Device with SPIKE App installed
- Student journals

Prepare

- Check to make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.

KEY OBJECTIVES

Students will:

- Write pseudocode before programming
- Read pseudocode and program to match
- Compare pseudocode to a program

STANDARDS

CSTA

2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms

VOCABULARY

Pseudocode

Clarify

1. Engage

Ignite a discussion:

- Ask students to talk about abbreviations they use when texting (e.g., LOL = laugh out loud, OTW = on the way, etc.).
- Ask students why they use abbreviations and emojis instead of writing full words and thoughts.
- Can those abbreviations and emoticons make the receiver of the text confused?

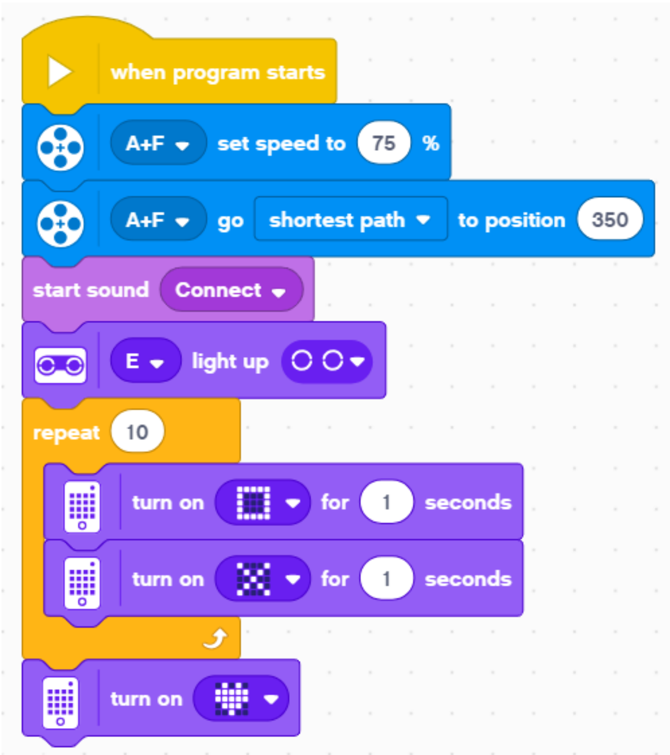
How are these abbreviations and emoticons like pseudocode?

- What is pseudocode and how is it used?
- Where have you used pseudocode in this course?

2. Explore

Tell students they will be using pseudocode today. They will read and write pseudocode and compare pseudocode to coding.

Ask students to compare the pseudocode and the code below. How could they use the pseudocode to debug the program? Where are the differences and how would you change the code to match the plan in the pseudocode?



When the program starts

Set motors' speed to 50%

Move motors to position 350

Play the connect sound.

Light up the lights on the distance sensor in port C

Repeat the following ten times.

Turn on an X for 1 second. Change the X to a heart for one second.

Then turn the heart on.

What changes need to be made? What should be clarified or added?

Sometimes people write pseudocode to explain new code they encounter. They use pseudocode as notes to help remember the purpose of the new code.

Find the **Place Your Order** lesson in the **Kickstart a Business** unit.
Complete **Steps 01-03** in **Place Your Order**. Write pseudocode for the Power up! Program shown in **Step 03**. Erase the comment and add your pseudocode.

3. Explain

Allow students to share their pseudocode.
Ask students:

- What did you write as your pseudocode?
- Do the words need to be exactly the same for all teams?
- What must all the pseudocodes have in common?

4. Elaborate

Students will read pseudocode, write code, and verify it works.

Have students complete **Step 04** in **Place Your Order**. Read the pseudocode and write the code. Run the program and verify it works. Debug as needed.

5. Evaluate

Teacher Assessment

Evaluate the students' understanding of pseudocode.
Evaluate the students' understanding of how pseudocode can be used to debug code.

Self-Assessment

Have students answer the following in their journals:

- How can writing pseudocode help with making a plan for writing code?
- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on time management today.
- Using a scale of 1-3, rate yourself on materials (parts) management today.

Create a New Product

Grade 6-8 | 45 minutes | Intermediate

Create a New Product

Students will demonstrate creativity and present their ideas to the class.

Questions to Investigate

- How can you be creative and look at things from a different perspective?
- How can you sell your idea to someone else?

Materials Needed

- SPIKE Prime Sets
- Device with SPIKE App installed
- Student journals
- Sticky notes or small colored paper
- Markers
- Tape

Prepare

- Check to make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.

KEY OBJECTIVES

Students will:

- Demonstrate creativity in creation of ideas for ways to use something familiar
- Design a new product and present it to the class in a commercial

STANDARDS

CSTA

2-CS-02 Design projects that combine hardware and software components to collect and exchange data.

2-AP-15 Seek and incorporate feedback from team members and users to refine a solution that meets user needs.

VOCABULARY

Creativity

1. Engage

Ignite a discussion. Watch a couple of commercials or have students talk about a memorable commercial.

- What makes them remember the commercial?
- What makes them want to buy the product?
- What information is given?

Are there commercials that make someone **not** want to buy a product or service?

Ask students to get a piece of paper and a pencil. In order to make good commercials, marketers need to be creative and be able to look at one thing from a variety of perspectives. Discuss items that many people relate to or something completely unique that will draw the audience's attention.

Draw a circle on the board or show a picture of a circle. Ask the class what it is. They will likely respond a circle or an "O." Tell them that is one answer. Then, give students 1 minute to individually write down 10 things other than a circle or an "O" that it could be. Remind them to consider it from several perspectives.

After they have finished writing down several ideas, ask students to name some of the items on their list. Share these possible answers:

- The eye of the Cyclops
- A black hole
- Zero
- Basketball hoop from above
- Pizza
- Baseball
- Basketball
- Wedding ring
- Steering wheel
- Donut

Now, ask students to add to their lists five more things the shape could be.

2. Explore

Tell students they will be coming up with an idea. They will all have the same model, but they will have to come up with what it is. Then, they will be presenting their ideas to the class to see who can get support for their original idea. Encourage students to be creative and to have fun.

Find the **What is This?** lesson in the **Extra Resources** section of **Start**.

Complete Steps 01-03. Discuss what the model does. You may modify the program if you like. You may not add any pieces to the model, but you can add sound and lights and change the movement.

3. Explain

Allow students to share some general ideas about their projects in order to spark some new thoughts.

Ask students as a class about their ideas without getting into specifics so all the ideas presented will be a surprise.

- Ask the class in a show of hands which teams changed the programming.
- Ask the class how many teams added sound.
- Ask the class how many teams added lights.
- Ask the class how many teams changed the motion.

4. Elaborate

Students will communicate their ideas.

Allow students a short amount of time to incorporate new ideas from listening to other teams.

Have students choose one:

- Create a short video explaining their idea and showing their model in a commercial; videos are then shown to the class
- Present a live commercial to the class

Remind students that their presentations should include:

- What the invention is
- Why you need this great invention

Remind students to have fun and be creative.

5. Evaluate

Teacher Assessment

Evaluate the students' understanding of how presentations can affect the way you feel about an idea.

Ask students

- What types of presentations were memorable and why.
- What types of presentations made them more interested in a product.

Self-Assessment

Have students answer the following in their journals:

- What was memorable about the presentations you saw?
- What characteristics of a good teammate did you display today?
-
- Using a scale of 1-3, rate yourself on time management today.
- Using a scale of 1-3, rate yourself on materials (parts) management today.

LEGO® Education SPIKE™ Prime

Share Your Ideas

Grade 6-8 | 45 minutes | Intermediate

Share Your Ideas

Students will identify something that could be automated and then present their ideas to the class.

Questions to Investigate

- How do engineers determine if a task is suitable to automate?
- How can an idea become a design?

Materials Needed

- SPIKE Prime Sets
- Device with SPIKE App installed
- Student journals
- Sticky notes
- Markers
- Tape
- Tape measures

Prepare

- Check to make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.

KEY OBJECTIVES

Students will:

- Determine if a task can be automated.
- Communicate to others your ideas.

STANDARDS

CSTA

2-CS-02 Design projects that combine hardware and software components to collect and exchange data.

VOCABULARY

Automate

1. Engage

Ignite a discussion with students about things they do that they wish were easier. Perhaps something in their morning routine, how they get to school, or at school.

Ask students which of these ideas could be automated – or improved with the use of robotics?

Let students discuss their ideas.

2. Explore

Students will explore ways that robots can help people do typical activities.

Tell students they will explore robots helping them to do normal activities. They will bring their ideas to life.

Students will work in pairs to design and build a robot. Find the **Ideas, the LEGO Way** in the **Extra Resources** section of **Start**. Direct students to complete **Steps 01 - 03**.

Brainstorm ideas for the type of robot they would like to build. It can be simple, using only one motor or complex using motors and sensors. Remember don't throw out any ideas.

Complete **Step 04**. Make a few rough sketches or put together a few bricks. This is not the time to complete the project – just get some ideas more fully developed. Allow times for students to work on their ideas.

3. Explain

Students will share information with another team and ask questions so projects will improve based on feedback.

Ask students to partner with another team. Complete **Step 05**. Ask clarifying questions of each other. Make positive suggestions such as "Have you considered…"

Ask the students to share interesting, cool, or fun ideas that they saw from the team they partnered with. Have those ideas more fully explained by the team that created them.

4. Elaborate

Students will incorporate information received into their projects and present their projects.

Ask students to complete **Step 06** while incorporating information learned from **Step 05**. Ask students to consider new ideas they heard that make them want to modify or add to your brainstorm ideas.

Have students choose one:

- Create a short video explaining their idea and showing their model. The videos are shown to the class.
- Present live their ideas to the class.

Remind students that their presentations should include:

- Why it is needed.
- What the idea is.
- Why class members should support it.

Optional:

Give each student one sticky note. Each student writes the name of their favorite idea on the sticky note. The teacher collects and organizes the sticky notes, creating a bar graph for each idea listed.

When the bar graphs are complete, the ideas who garnered the most votes are congratulated. Students share why they think the ideas who received the most votes were well received by the class.

5. Evaluate

Teacher Assessment

Evaluate the students' understanding of how to generate ideas with a group of people.

Ask students:

- How hearing other teams' ideas helped them to rethink some of their own ideas.
- What they would do next, if the assignment were to continue and they were to incorporate the feedback gathered during the class discussions.

Self-Assessment

Have students answer the following in their journals:

- What did you find the most challenging aspect of this lesson?
- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on time management today.
- Using a scale of 1-3, rate yourself on materials (parts) management today.

LEGO® Education SPIKE™ Prime

Mini-Challenge: Creativity in Business

Grade 6-8 | 45-90 min | Intermediate

Mini-Challenge: Creativity in Business

Create a new robot and integrate effective presentation skills into a persuasive business pitch to garner support.

Questions to Investigate

- How do entrepreneurs “sell” their ideas to others?

Materials Needed

- SPIKE Prime Sets
- Device with SPIKE App installed
- Student journals
- Sticky notes or small colored paper
- Markers
- Tape

Prepare

- Check to make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.

1. Engage

Ignite a discussion about ways students have seen someone or a group of people try to get support for an idea. It could be a commercial, a television show, or a website where people try to get funding.

Identify what information is given and how they try to persuade people to support their ideas or give their money.

Ask students to define entrepreneur and what skills an entrepreneur needs to be successful.

Let students share several examples.

KEY OBJECTIVES

Students will:

- Create a new robot.
- Present your ideas persuasively to others.
- Integrate effective presentation skills into a business pitch.

STANDARDS

CSTA

2-CS-02 Design projects that combine hardware and software components to collect and exchange data.
2-CS-03 Systematically identify and fix problems with computing devices and their components.
2-AP-15 Seek and incorporate feedback from team members and users to refine a solution that meets user needs.

VOCABULARY

Entrepreneur

2. Explore

Tell students they will build and program a robot that is the newest and greatest idea and that they will be sharing their ideas with the class.

Tell students they are now part of the new products team for a start-up company. They need to create a new, useful, “I need it now!” product and the marketing around it. They will be presenting their more complete ideas to the class to see who can get support.

3. Explain

Ask students if they are having fun creating new ideas and trying new ways to do things. Remind them they are not looking for perfection. This is a quick introduction to brainstorming and refining. While they will be showing a finished prototype – it is not a product ready to sell to the public.

Have students share their initial ideas with the class. They don’t have to show their prototypes, but they can if they wish. They can ask for feedback or answer questions.

4. Elaborate

Allow students time to incorporate ideas gathered from the class.

Have students choose one:

- Create a short video explaining their idea and showing their model. The videos are shown to the class
- Present live their ideas to the class including a prototype

Remind students that their presentations should include:

- Why it is needed.
- What the idea is.
- Why class members should support it.

Optional:

Give students one sticky note each. Each student writes the name of their favorite idea on the sticky note. The teacher collects and organizes the sticky notes, creating a bar graph for each idea listed.

When the bar graphs are complete, the ideas who garnered the most votes are congratulated. Students share why they think the ideas who received the most votes were well received by the class.

5. Evaluate

Teacher Assessment

Evaluate the students’ understanding of how they can persuade others to agree with their ideas.

Ask students:

- How presentations can affect the way you feel about an idea.
- What types of presentations were most affective at gaining support of an idea.

Self-Assessment

Have students answer the following in their journals:

- What was one important thing you learned completing the presentation?
- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on time management today.
- Using a scale of 1-3, rate yourself on materials (parts) management today.

LEGO® Education SPIKE™ Prime

Connecting to Careers: Marketing, Arts, A/V Technology & Communications

Grades 6-8 | **90 minutes** | **Beginner**

Connecting to Careers: Marketing, Arts, A/V Technology & Communications

In this lesson series, students will have the opportunity to explore and research careers in Marketing, Arts, A/V Technology & Communications.

Prepare

Prior to starting the lesson, prepare the following:

- Set aside enough LEGO® bricks for students' models.
- Make sure you have enough devices and access to the Internet for student use during this lesson.
- Make copies of the handouts (if desired) or place into a digital platform for student use.
- Prepare images of different jobs. There should be a variety of images to show different jobs within the 16 career clusters. (Example for Architecture & Construction: Architectural drafter, engineer, carpenter, engineer, electrician, HVAC technician, painter, environmental designer, etc.).

1. Engage

Ignite a discussion with students:

- What are some of the jobs adults you know have?
- What do you think they like about their jobs? Is there such a thing as the perfect job?
- Present the images of the jobs in the area of Arts, AV Technology & Communication and Marketing to students—as a class, decide how you would categorize the jobs. Ask questions like:
    - What jobs belong together?
    - What kind of similar skills do these different jobs use?

KEY OBJECTIVES

Students will:

- Articulate their personal interests and goals
- Relate their personal interests and goals into possible career pathways
- Explore various careers in career pathways

STANDARDS

Career Ready Practice 10- Plan education and career path aligned to personal goals. (CCTC)

VOCABULARY

Career
Career Cluster
Qualifications
Skills
Education
Knowledge

- What kind of environments are associated with these jobs?
- Do any of these jobs interact or rely on one another? If so, how?

Ask students to think about what interests them. What kind of job(s) would they like to have in the future?

2. Explore

Students will be assigned two career clusters each time career connections lessons are taught. By the end of the course, students will have explored all 16 career clusters.

There are 16 career clusters:

- Agriculture, Food & Natural Resources
- Architecture & Construction
- Arts, A/V Technology & Communications
- Business Management & Administration
- Education & Training
- Finance
- Government & Public Administration
- Health Science
- Hospitality & Tourism
- Human Services
- Information Technology
- Law, Public Safety, Corrections & Security
- Manufacturing
- Marketing
- Science, Technology, Engineering & Mathematics
- Transportation, Distribution & Logistics

Students previously studied careers in the Education & Training and Human Services pathways. The two career pathways we are studying today are:

- **Arts, A/V Technology & Communications**
- **Marketing**

Ask students to brainstorm as many jobs as you can within each career cluster. Have one person write down the jobs as they are named.

Tell students to think about the lessons they completed thus far in Unit 2. They have been creative and communicated ideas to others, gathered ideas from others, and made presentations. What skills were needed to complete the lessons? How do these skills and activities relate to the two pathways – Marketing and Arts, A/V Technology & Communications?

In small groups of 4, students complete online research to find out more about each career clusters. Allow time to get information about the jobs that included, especially jobs they had not heard of or had not thought of in their brainstorm.

Students should also research:

- Skills needed
- Forecast of future job openings (and current)
- Certifications, licenses, apprenticeships, etc. required

In their group, students create a visual representation (build a model) of one of the two career clusters. They will build one or more physical models with LEGO® bricks. Each person can build a model, or the group can create one large model that represents all ideas about one of the career clusters.

Each group will be responsible for a quick one-minute presentation of their LEGO® model(s) to explain how it represents their career cluster.

3. Explain

When the students have finished building, allow each group to present their model. Ask students:

- Tell us about your LEGO® build.
- What kind of interests would these career clusters have?
- What are some of examples of jobs in these career clusters?
- Can you explain the difference between a job and career cluster?
- How are these career clusters like one another?

4. Elaborate

Students will investigate how are Marketing, Arts, A/V Technology & Communications related.

What are some overarching ideas that run through all these career pathways? (For example, all these pathways focus on communicating an idea to others.)

What are some jobs that overlap or are interdependent between the pathways? (For example, someone in marketing may have an artist to design a logo or a background that will be used in a video shoot to communicate a way of looking at a product they are helping to sell.)

Students should create a Venn Diagram, with each circle representing one of the career pathways - Marketing or Arts, A/V Technology & Communications. Students should add information, skills, and jobs that are in each pathway or are shared between the pathways. Have students compare their answers, explaining their thought processes.

5. Evaluate

Evaluate the students' skills development by observing if they:

- Articulate their personal interests, skills, and goals.
- Relate their personal interests and goals into career pathways.
- Explore various careers in career pathways.

LEGO® Education SPIKE™ Prime

Model Debugging

Grade 6-8 | 90-135 min | Intermediate

Model Debugging

Determine what is wrong with a physical model and fix it.

Questions to Investigate

How do engineers identify and repair hardware and software problems within a design?

Materials Needed

- SPIKE Prime Set
- Device with SPIKE App installed
- Student journal
- Pencils
- Pen or marker (one per team)
- Tape measures or rulers
- Scissors (one per team or per person)
- Printed worksheet for Broken Lesson (one per team; link to document)
- Printed worksheet for unplugged activity containing light ovals (located after Evaluate)
  Note: These ovals are meant to be very light in color so students will need to trace in order to be able to cut accurately.

Prepare

- Check to make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.
- The answer for what is wrong with the hardware is provided on website within the lesson. https://education.lego.com/en-us/lessons/prime-invention-squad/broken#Planitem2

KEY OBJECTIVES

Students will:

- Identify and repair a hardware problem in a physical model.

STANDARDS

CSTA

2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

3A-CS-03 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.

3A-AP-16 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.

3A-AP-17 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

VOCABULARY

Debugging

CNC

1. Engage

Give each team a sheet with 4 ovals on it, an unsharpened pencil, and a pair of scissors. Tell students to use the pencil to connect the lines prior to cutting out the ovals. (The first issue should be that they must sharpen the pencils in order to connect the lines. They need to troubleshoot.)

Ask students why they can't use the pencil as is – i.e., because it is not sharpened. Tell them the pencil needs to be repaired, and in this case, the pencil needs to be sharpened. The pencil represents the hardware – which needs to be fixed in order to complete the task of completing the circles.

Each student will carefully cut out two ovals. The class should put all the ovals together – the ovals should be placed on a flat surface where all can be viewed at the same time. Ask students if all the ovals are cut exactly the same. Why not?

Discuss with students what type of machines can make an exact cut repeatedly so all pieces are the same. A CNC [computer numerical control] machine is programmed to make the same repetitive cuts. For example, a laser cutter cuts plastic, wood, or cardboard into exact shapes.

Allow students to research what CNC machines are and how they are used. Ask students to name objects that they believe are made with a CNC machine.

2. Explore

Troubleshoot fixing a model or hardware. Have students find the **Broken** lesson in the **Invention Squad** unit.

Ask students to complete the lesson through **Step 03** and then stop. Try the program. Notice the machine is not working properly. The machine is supposed to help you cut parts, but it doesn't work. Ask students to fix the model so that it works with the given program without changing the program.

Support students in fixing the model by asking these questions:

- What do you notice about the CNC machine when you run the program? (It doesn't work; the machine is broken).
- How do you know it is a hardware problem?
- What parts of the machine seem to be working?
- What parts of the machine do not seem to be working?
- As engineers, what is the first step we should take in fixing the machine? (Identifying problems; if more than one problem has been discovered, then deciding which problem to repair first.)

As a class, identify the problems. Prompt students to do the following:

- Have students write the issues found in their journals or on chart paper.
- Have students take a close look at the machines and see how it behaves versus what they hope it will do.

- Have students brainstorm solutions.

Demonstrate fixing the model with students. First, focus on the left side of the back of the model, which isn't secured.

- Look at the lack of symmetry between the two sides where the machine is attached to the base. The right side is tight, but the back-left side is not attached.
- Ask students what suggestions they have for attaching the back-left side.
    - Students should suggest adding a grey peg so the blue L-beam is held tightly to the magenta biscuit.
- Have students add the grey peg. Then, test the machine. Does it function properly? No. Does it function better? Probably a little better. Run the program a few more times and observe the results.
- As a class, go back to the brainstorming of solutions. Do we need to add anything to our brainstorm ideas?

Allow the students to work on the next issue, the paper moves too fast.

- Prompt students to notice that the paper moves too fast.
- Ask students what would influence making the model move too fast (The motor). The motor speed in programming is fine. However, look carefully at the gears.
- Ask students how the gears affect the speed. Then, ask if anyone has an idea of how to fix this area (Hopefully, someone suggests changing the gears – moving the large gear to the small gear position and the small gear to the large gear position).
- Have students transpose the gears. Then, try the program again. How does the model behave? Is it better? Yes. Is the model completely correct? No.

3. Explain

Review how you approached fixing the first two issues. Ask students to explain:

- Procedures or steps for identifying a hardware problem
- The problems that were found so far and how they were fixed

4. Elaborate

Challenge students to continuing fixing the model.

Tell students there are 2 additional hardware problems with the model. Teams should use the method of problem solving you have modeled.

- Identify the problem.
- Take a close look.
- Observe how the model behaves verses what you think it should do.
- Brainstorm solutions.
- Make one change and test the model. Did it help?
- You may ask other teams for help as needed.

Have teams work together to:

- Identify hardware problem(s)

- If more than one problem has been identified, then discuss with their partner which problem you should try first to correct.
- Document hardware problems in their journals. Then try and repair it. Were you successful? If not, iterate or repeat discussing, noting, making a change, and testing.

  Note: The Broken Lesson shows hints for what is wrong with the model. https://education.lego.com/en-us/lessons/prime-invention-squad/broken#Planitem0

  Note: Have students keep their models built once they have fixed all the hardware issues and the program works properly. They may use the model in a future lesson.

5. Evaluate

Teacher Assessment

Evaluate the students' understanding of how the programming and the physical model must work together.

Ask students what challenges they encountered in trying to fix a broken model they had never seen before.

Self-Assessment

Have students answer the following in their journals:

- How could determining a problem and fixing a broken physical model be helpful with the challenge?
- Did you ask other teams for help? If so, what did you learn?
- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on time management today.
- Using a scale of 1-3, rate yourself on materials (parts) management today.

Engage – Unplugged Activity for CNC

LEGO® Education SPIKE™ Prime

Software Debugging

Grade 6-8 | 90 minutes | Intermediate

Software Debugging

Determine what is wrong with a software program and debug it.

Questions to Investigate

How do engineers identify and repair hardware and software problems within a design?

Materials Needed

- SPIKE PRIME Set
- Device with SPIKE App installed
- Student journal
- Pencils
- Pen or marker (one per team)
- Tape measures or rulers
- Printed worksheet for Track Your Package Lesson (one per team; link to document)

Prepare

- Check to make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.
- If students have previously built the model for the Broken lesson and still have it put together, they can use that model without a complete rebuild. Otherwise, they can build the model for Track Your Packages when they start the lesson.

1. Engage

Ask students to give directions to go from Point A to Point B, which is a diagonal line on a grid. The rules are you can only move forward or backward, left or right one square at a time.

How many unique pathways can the class find in 3-5 minutes?

KEY OBJECTIVES

Students will:

- Debug a software problem.
- Learn and follow a debugging strategy involving multiple steps.

STANDARDS

CSTA

2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms.
2-AP-13 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
2-AP-17 Systematically test and refine programs using a range of test cases.
3A-CS-03 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
3A-AP-16 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.
3A-AP-17 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

VOCABULARY

Debugging
Pseudocode

				Point B	
Point A					

Allow students time to investigate several pathways. Discuss the pseudocode for their ideas for pathways.

2. Explore

Students will investigate a model that is not working. Have students launch the **Track Your Packages** lesson in the **Kickstart a Business** unit.

If students do not have a model from the **Broken** lesson available:

- Ask students to build the model from **Track Your Package**.

If students have the model from the **Broken** lesson:

- Ask students to slightly change the model from the **Broken** lesson.
- The arm on the tracker bottom is slightly different.
- Change the arm starting at **Building Instruction 8**. Do **not** remove the arm, just change the “flag” on the arm.
- The remainder of the tracker bottom is the same. Students may wish to look through the remainder of the instructions. If the Broken model worked well, then this model should work well.

Ask students to complete **Step 03** and then stop. Allow students time to try the program. Students will notice the machine is not working properly.

3. Explain

Discuss why the machine is not working properly.
Ask students:

- What do you notice about the CNC machine when you run the program (it doesn't work, the machine looks to be functioning but there is a bug in the program)
- How do you know if it is a hardware or software problem?

Ask students to explain the procedures or steps for debugging a program.

Go through the following debugging procedures:

1. Say aloud what the problem is. For example, "The robot is not moving" or "Only one wheel on the robot is turning."
2. Say aloud what you expected it to do. Example, "Both motors should turn on at the same time."
3. Have you checked the physical model to be sure all the connections are solid? If so, then test your program a couple of times to be sure it always happens.
4. Check the ports to be sure they match with the wiring to the hub.
5. Find the most recent code you added. Did it work BEFORE you added the code? If so, read through the new code carefully.
6. If you started with one large program, break it into parts, verifying a small amount of code at a time. Keep adding code until you find where it breaks. Then, debug that section.
7. Explain the code to your partner. Have your partner verify the code.
8. Look for pieces of code that are directly related to the issue.
9. Make one small change and test the code.
10. Continue to make small changes until you get the intended behavior.
11. Once you have debugged the program, take a minute to document what you have learned. What tips can you give yourself for the next time you need to debug?

4. Elaborate

Challenge students to debug the program.

Ask students to:

- Create pseudocode for how the tracker should move in order to arrive at the destination.
- Compare their program to the pseudocode to identify what parts of the program seem to be working, what parts are not working.
- Discuss with their partner to determine which part of the program to correct first.
- Document identified problems in their journal. Then, try and debug it. Did you fix it? If not, iterate or repeat discussing, noting, making a change, and testing.

Allow time for students to explore the program after changes are made.

Discuss the experience with students. Ask students to:

- Explain how they identified bugs in the program
- Explain what procedures they used
- Share how they made the tracking move on a diagonal road

5. Evaluate

Teacher Assessment

Evaluate the students' understanding of how the programming and the physical model must work together.

Ask students if they asked other teams for help. If so, what did they learn?

Self-Assessment

Have students answer the following in their journals:

- How could determining a problem and debugging a program could be helpful with the challenge?
- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on time management today.
- Using a scale of 1-3, rate yourself on materials (parts) management today.

LEGO® Education SPIKE™ Prime

Dancer Break Down

Grade 6-8 | 45 minutes | Intermediate

Dancer Break Down

Students will investigate strategies for debugging programs.

Questions to Investigate

How do software engineers identify and fix bugs in a program?

Materials Needed

- SPIKE Prime Set
- Device with SPIKE App installed
- Student journal

Prepare

- Check to make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.

1. Engage

Discuss with students how to identify if a problem is with the hardware (model) or the program.

Now students will need to find ways to identify if problems are in the model design or the program.

Watch the video of the break dancer model dancing to get an idea of how the model should move.
https://education.lego.com/en-us/lessons/prime-life-hacks/break-dance#ignite-a-discussion

Discuss with students the different ways the dancer can move. Ask students questions like:

- Which parts of the model were moving?
- How can you replicate that movement with the model?
- If something goes wrong, how will you know if it is the model or the program?

KEY OBJECTIVES

Students will:

- Identify a problem and debug the program.
- Compare pseudocode to code to help with debugging.
- Write pseudocode from code to help with debugging.

STANDARDS

CSTA

2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms.
2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
2-AP-17 Systematically test and refine programs using a range of test cases.
2-AP-19 Document programs in order to make them easier to follow, test, and debug.

VOCABULARY

Debugging

2. Explore

Students will build a break dancer model to investigate identifying bugs and fixing them.

Direct students to the **BUILD** section in the SPIKE App. Here students can access the building instructions for the **Break Dancer** model. Ask students to build the model. The building instructions are also available at https://education.lego.com/en-us/support/spike-prime/building-instructions.

Ask student to open a new project.

Test the Model

Students will identify how the model moves and create a new program.

Ask students to examine their model closely to identify how it should move. Students should try moving the motors and other parts of the model to see if it will move similar to the video based on the discussion in the engage section.

Allow students time to investigate how the model moves without creating a program. Ask students questions like:

- Did the model move in similar ways to the video?
- Does anything on the model seem to not be working properly?

Test the Program

Ask students to add this program into the programming canvas. Have them write pseudocode to explain what the program should do. Ask students to run the program.

Ask students how the program worked. One of the motors did not move. Why? Students should be able to debug the issue that the motor is in port F, not port B. This is a programming issue or a hardware issue – meaning that the fix could be to change the program or the wiring. If a model was designed correctly, the program should be changed. Have students discuss which fix makes the most sense with the Break Dancer.

3. Explain

Discuss the program with students and why it does not seem to be working correctly. Ask students questions like:

- What does it mean to debug a program?
- Why was it important to test the model before running the program?

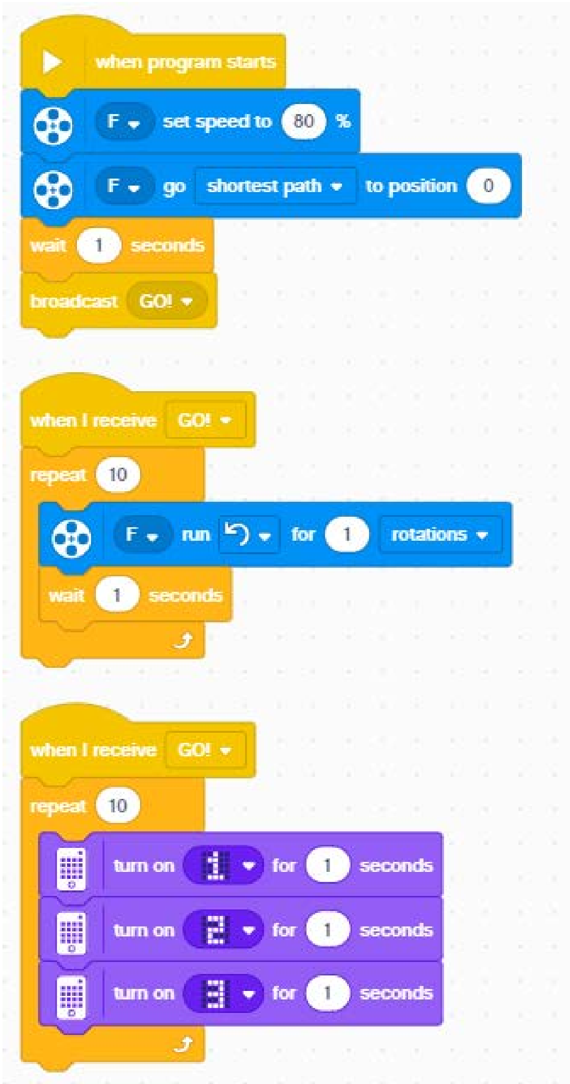
Students should be able to debug the issue that the motor is in port F, not port B. This is a programming issue or a hardware issue – meaning that the fix could be to change the program or the wiring. If a model was designed correctly, the program should be changed. Have students discuss which fix makes the most sense with the Break Dancer.

4. Elaborate

Allow students additional time to explore different types of errors.

Example 1:

Have students copy this program. Write pseudocode to explain what the program should do.



Tell students the dancer's legs should move 10 times in sync with the light. Test the program. Debug the program. **Hint:** Change the timing.

Example 2:

Have students copy this program. Write pseudocode to explain what the program should do.

Have students test the program. What happens? The students must have a force sensor attached in order to start the program running by a press. The model does not have a force sensor attached; therefore, the program will not run. How can you fix the program? **Hint:** Choose a different start block.

5. Evaluate

Teacher Assessment

Evaluate students' understanding of debugging.

Ask students questions like:

- How does testing help identify bugs in your program?
- How can you tell the need to troubleshoot hardware and not the program?
- What are the types of errors that could be made in hardware and in software?

Self-Assessment

Have students answer the following in their journals:

- What did you learn today about identifying and fixing bugs?
- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on time management today.
- Using a scale of 1-3, rate yourself on materials (parts) management today.

LEGO® Education SPIKE™ Prime

Dance to the Color

Grade 6-8 | **45 minutes** | **Intermediate**

Dance to the Color

Students will investigate strategies for debugging programs.

Questions to Investigate

How do software engineers identify problems within a program?

Materials Needed

- SPIKE Prime sets ready for student use. Prior to the first lesson, please visit the following website for help with set up, kit organization and SPIKE App https://education.lego.com/en-us/start/spike-prime/intro
- Devices with the SPIKE App installed.
- Student journals

Prepare

- Ensure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.
- Ensure students have the Break Dancer model built, which was used in the Break Dancer Break Down lesson.

1. Engage

Create a plan for testing programs to identify and fix bugs.

Spark a discussion with students on how to fix problems. Ask students to think of a time when they had an issue or problem or a time that they had an experience that did not turn out as expected.

Discuss what could have helped change the situation. Ask questions like:

- What steps could you have taken upfront to have a better outcome?
- What could you have done when you realized there was a problem to help change the outcome?

KEY OBJECTIVES

Students will:

- Identify a problem and debug the program.
- Identify and fix problems that are hardware or software or the interface of hardware and software.

STANDARDS

CSTA

2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms.
2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
2-AP-17 Systematically test and refine programs using a range of test cases.
2-AP-19 Document programs in order to make them easier to follow, test, and debug.

VOCABULARY

Debugging

Discuss students' ideas for troubleshooting their experience and list ideas for how to troubleshoot. Share out the steps for identifying bugs or post in the classroom for all students to reference.

Steps to help identify bugs:

- Plan the program by creating a pseudocode or flowchart
- Document the program by using comments within the program
- Test your program a section at a time, adding more features after the current
- Test your program with all types of data that are relevant or expected to be used
- Test your program with types of data that are not expected to be used or outside the range expected

2. Explore

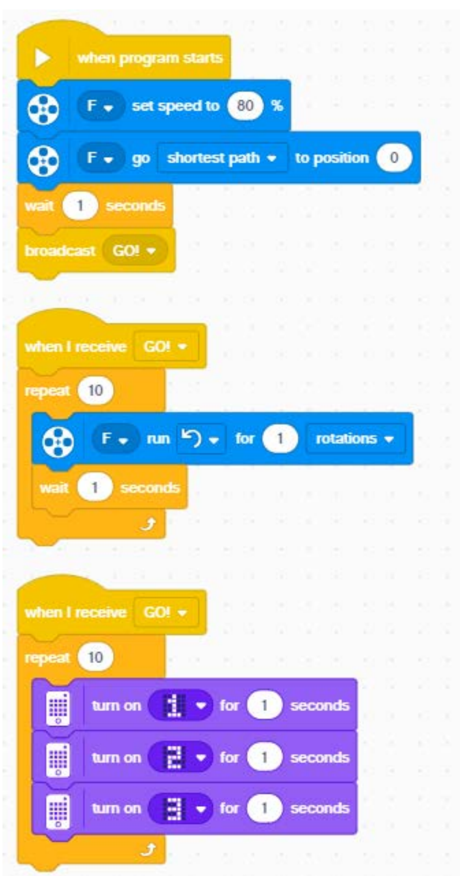
Students will use a break dancer model to investigate identifying bugs and fixing them.

Direct students to open a new project. Students should connect their hub.

Adding to a Working Program

Students will identify strategies to find bugs when adding to a program.

Ask students to copy this program into their programming canvas.



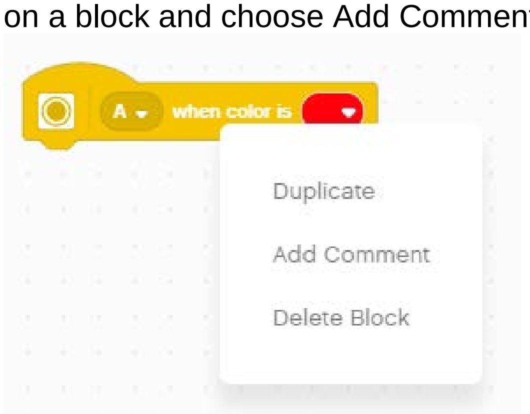
Write pseudocode to explain what the code has the robot do. Run the program to confirm there are no current errors in the program. Verify that the legs should move, while the arms should not.

Ask students to modify their program so the dancer only moves when the color sensor senses red. When the color red is sensed, move the legs only.

Hint: Suggest to students that they use this hat block. They may choose not to use the broadcast block or they may put the first set of blocks under the when color is block.

Allow students time to create their new program. Students should write pseudocode first and then add comments for each section of the program.

Hint: Remind students that to make a comment in a program, right click on a block and choose Add Comment.



Students should test their program to make sure it works. Remind students to utilize the process discussed in the engage section to test the program for bugs.

Add arm movement to the leg movement when the color red is sensed.
Allow students time to create their new program, test the program, and debug as needed.

Ask students to record all debugging activities in their journal. Students should explain in their journals any errors discovered while testing and how they resolved the issues.

3. Explain

Discuss the program with students and where they found bugs.
Ask students questions like:

- What bugs did you find in your program?
- Could any of the problems be with the model? Why or why not?
- How did you utilize the color sensor and the blocks in programming?

4. Elaborate

Challenge students to create a program that has their dancer moving arms and legs quickly when one color is sensed and slowly if another color is sensed when the program begins.

Allow students time to investigate the program to see what changes are needed.

Ask students to share their final movements with the class. Consider having a dance party for all the models to move at the same time.

5. Evaluate

Teacher Assessment

Evaluate students' understanding of debugging hardware and software.

Discuss the program in this lesson with students.
Ask students questions like how does testing help identify bugs in your program.

Self-Assessment

Have students answer the following in their journals:

- What did you learn today about identifying and fixing bugs?
- How do you add comments to a coding block?
- Why would you add comments to a section of code?
- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on time management today.
- Using a scale of 1-3, rate yourself on materials (parts) management today.

LEGO® Education SPIKE™ Prime

Mini-Challenge: New Routes

Grade 6-8 | 90 minutes | Intermediate

Mini-Challenge: New Routes

Create a new program to move a robot along a specific path and debug it.

Questions to Investigate

How do programmers identify software problems when creating a new program?

Materials Needed

- SPIKE PRIME Set
- Device with SPIKE App installed
- Student journal
- Pencils
- Pen or marker (one per team)
- Tape measures or rulers
- Printed worksheet for Track Your Packages Lesson (one per team; link to document)

Prepare

- Check to make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.
- If students have previously built the model for the Broken lesson and still have it put together, they can use that model without a complete rebuild. Otherwise, they can build the model for Track Your Packages when they start the lesson.

1. Engage

Ask students to give directions to go from Point A to Point B, which is a diagonal line on a grid. The rules are you can only move forward or backward, left or right one square at a time. You cannot use any of the squares that are filled with color.

How many unique pathways can the class find in 1-2 minutes?

KEY OBJECTIVES

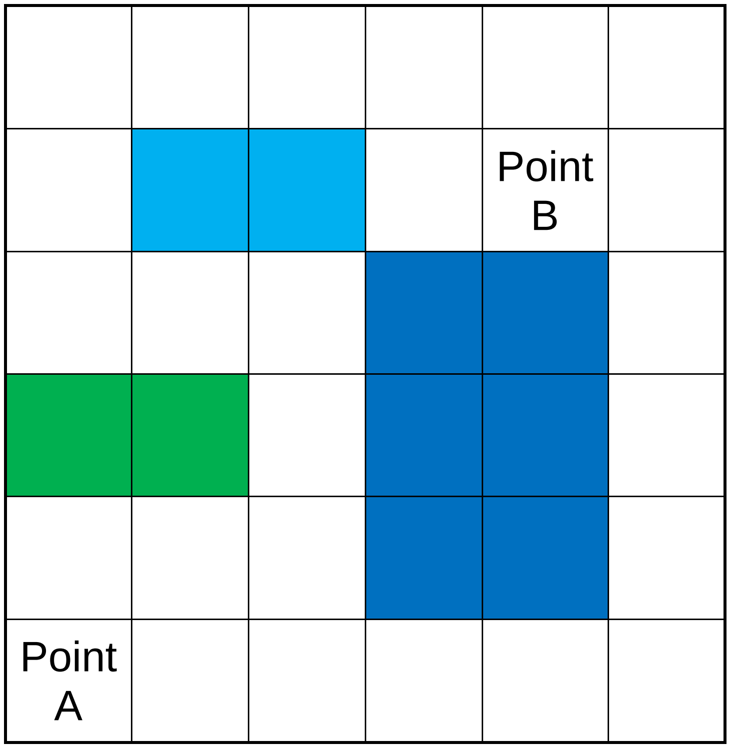
Students will:

- Program a robot that can autonomously move from one location to another following a prescribed route.
- Iterate a solution and debug a software problem.

STANDARDS

CSTA

2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms.
2-AP-13 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
2-AP-17 Systematically test and refine programs using a range of test cases.
3A-CS-03 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
3A-AP-16 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.
3A-AP-17 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.



Allow students time to investigate several pathways. Discuss the pseudocode for their ideas for pathways. Have students share with the class.

2. Explore

Students will create new routes and write a program to track the package.

Give each team a Track Your Packages worksheet. Have students modify one of the routes by drawing a new road in pencil. When the team is happy with the design, draw the road in marker.

Write pseudocode for following the route.

3. Explain

Allow students to share their pseudocode. Ask students:

- How did you modify the route?
- How detailed is your pseudocode? Will it be easy to write code based on your plan?

Remind students of the procedures or steps for debugging a program.

Go through the following debugging procedures:

1. Say aloud what the problem is. For example, “The robot is not moving” or “Only one wheel on the robot is turning.”

2. Say aloud what you expected it to do. Example, “Both motors should turn on at the same time.”
3. Have you checked the physical model to be sure all the connections are solid? If so, then test your program a couple of times to be sure it always happens.
4. Check the ports to be sure they match with the wiring to the hub.
5. Find the most recent code you added. Did it work BEFORE you added the code? If so, read through the new code carefully.
6. If you started with one large program, break it into parts, verifying a small amount of code at a time. Keep adding code until you find where it breaks. Then, debug that section.
7. Explain the code to your partner. Have your partner verify the code.
8. Look for pieces of code that are directly related to the issue.
9. Make one small change and test the code.
10. Continue to make small changes until you get the intended behavior.
11. Once you have debugged the program, take a minute to document what you have learned. What tips can you give yourself for the next time you need to debug?

4. Elaborate

Challenge students to write a program that will follow their designed route.

Ask students to:

- Compare their program to the pseudocode to identify what parts of the program seem to be working, what parts are not working.
- Discuss with their partner to determine which part of the program to correct first.
- Document identified problems in their journal. Then, try and debug it. Did you fix it? If not, iterate or repeat discussing, noting, making a change, and testing.

Allow time for students to debug their programs.

5. Evaluate

Teacher Assessment

Evaluate the students’ understanding of how the programming and the physical model must work together.

Self-Assessment

Have students answer the following in their journals:

- Did you ask other teams for help? If so, what did they learn?
- What was the one thing you had to iterate during this challenge?
- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on time management today.
- Using a scale of 1-3, rate yourself on materials (parts) management today.

LEGO® Education SPIKE™ Prime

U3-06 Connecting to Careers: Information Technology and Law, Public Safety, Corrections & Security

Grades 6-8 | **90 minutes** | **Beginner**

Connecting to Careers: Information Technology and Law, Public Safety, Corrections & Security

In this lesson series, students will have the opportunity to explore and research careers.

Prepare

Prior to starting the lesson, prepare the following:

- Set aside enough LEGO® bricks for students' models.
- Make sure you have enough devices and access to the Internet for student use during this lesson.
- Make copies of the handouts (if desired) or place into a digital platform for student use.
- Prepare images of different jobs. There should be a variety of images to show different jobs within the 16 career clusters. (Example for Architecture & Construction: Architectural drafter, engineer, carpenter, engineer, electrician, HVAC technician, painter, environmental designer, etc.).

1. Engage

Ignite a discussion with students:

- When you think of IT, what skills come to mind?
- What would be some of the benefits to working in Public Safety, Corrections and Security?
- Where are some of the places that people in these two career pathways would work?

KEY OBJECTIVES

Students will:

- Articulate their personal interests and goals.
- Relate their personal interests and goals into possible career pathways.
- Explore various careers in career pathways.

STANDARDS

Career Ready Practice 10- Plan education and career path aligned to personal goals. (CCTC)

VOCABULARY

Career
Career Cluster
Salary
Qualifications
Skills
Education
Knowledge

Present the images of the jobs in the area of Information Technology and Engineering & Law, Public Safety, Corrections & Security to students—as a class, decide how you would categorize the jobs. Ask questions like:

- What jobs belong together?
- What kind of similar skills do these different jobs use?
- What kind of environments are associated with these jobs?
- Do any of these jobs interact or rely on one another? If so, how?

Ask students to think about what interests them. What kind of job(s) would they like to have in the future?

2. Explore

Students will be assigned two career clusters each time career connections lessons are taught. By the end of the course, students will have explored all 16 career clusters.

There are 16 career clusters:

- Agriculture, Food & Natural Resources
- Architecture & Construction
- Arts, A/V Technology & Communications
- Business Management & Administration
- Education & Training
- Finance
- Government & Public Administration
- Health Science
- Hospitality & Tourism
- Human Services
- Information Technology
- Law, Public Safety, Corrections & Security
- Manufacturing
- Marketing
- Science, Technology, Engineering & Mathematics
- Transportation, Distribution & Logistics

Students previously studied careers in the Education & Training and Human Services, Arts, A/V Technology & Communications, Marketing. The two career pathways they are studying today are:

- **Information Technology**
- **Law, Public Safety, Corrections & Security**

Ask students what connections they see between these career clusters and the lessons in Unit 3. For example, debugging hardware and software relate to information technology, planning routes for delivery relate to public safety, working through a problem to find a solution relate to law and law enforcement, and so forth.

Ask students to reflect on what skills they practiced in Unit 3 lessons that they would bring to a job in the career pathways being studied today.

Ask students to brainstorm as many jobs as you can within each career cluster. Have one person write down the jobs as they are named.

In small groups of 4, students complete online research to find out more about each career clusters. Allow time to get information about the jobs that included, especially jobs they had not heard of or had not thought of in their brainstorm.

Students should also research:

- Skills needed
- Forecast of future job openings (and current)
- Certifications, licenses, apprenticeships, etc. required

In their group, students create a visual representation (build a model) of one of the two career clusters. They will build one or more physical models with LEGO® bricks. Each person can build a model, or the group can create one large model that represents all ideas about one of the career clusters.

Each group will be responsible for a quick one-minute presentation of their LEGO® model(s) to explain how it represents their career cluster.

3. Explain

When the students have finished building, allow each group to present their model. Ask students:

- Tell us about your LEGO® build.
- What kind of interests would these career clusters have?
- What are some of examples of jobs in these career clusters?
- Can you explain the difference between a job and career cluster?
- How are these career clusters similar to one another?

4. Elaborate

- Ask students if there were any jobs in today's career cluster that they had never heard of.

- Ask students if there were any jobs in today’s career cluster that were interesting to them.

Ask students to create a word web starting with one career pathway. Add as many words that relate to the pathway as possible in 5 minutes. Have teams exchange word webs and share with the class words they found interesting from another team’s web.

5. Evaluate

Evaluate the students’ skills development by observing if they:

- Articulate their personal interests and goals.
- Relate their personal interests and goals into possible career pathways.
- Explore various careers in career pathways.

LEGO® Education SPIKE Prime

Testing Prototypes

Grade 6-8 | **45 minutes** | **Beginner**

Testing Prototypes

Students will explore the engineering design process.

Questions to Investigate

- How do engineers take an idea and make it into a product?

Materials Needed

- SPIKE Prime sets ready for student use. Prior to the first lesson, please visit the following website for help with set up, kit organization and SPIKE App https://education.lego.com/en-us/start/spike-prime/intro
- Devices with the SPIKE App installed
- Student journals
- Unopened water bottles

Prepare

Ensure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.

1. Engage

Engage students in thinking about the design engineering process.

Engineers use a process called the design engineering process when they are trying to find a solution to a problem. Introduce students to the design engineering process. You may use your own process or the one provided below.

- Define a problem
- Research the problem
- Brainstorm possible solutions
- Select the most promising solution
- Construct a prototype
- Test and evaluate the prototype
- Communicate
- Redesign

KEY OBJECTIVES

Students will:

- Brainstorm ideas and develop solutions to a problem involving a bridge with movement.
- Build a working model and write code.
- Test the model and debug the program.
- Use the design engineering process.

STANDARDS

CSTA

2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.

2-AP-18 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.

2-CS-02 Design projects that combine hardware and software components to collect and exchange data.

VOCABULARY

Engineering Design Process
Constraint
Prototype

Ignite a discussion with students on how engineers use this process to design solutions to problems. Consider sharing examples. Encourage students to share when they have used this process or parts of this process.

2. Explore

Students will use the design engineering process to build a bridge.

Present students with a problem to solve. Explain to students that they have been hired to construct a model of a bridge. The roadbed of the model, which is the large yellow technic plate, needs to raise above the table by 3 inches.

Ask students to record the steps of the design engineering process in their notebooks as they complete them. Together, complete step 1 of the process by defining the problem. Students should identify that they need to build a bridge that has the roadbed 3 inches off the table. Identify other needs – like what should a bridge do – carry weight, not move side to side, and so forth.

Allow students time to complete step 2, to research the problem, as needed.

Students should record all ideas as they move to the next step, brainstorming possible solutions. Provide students with these constraints for the challenge:

- Students may only use the following elements from their set
  - Large yellow technic plates
  - Technic beams
  - Connectors or axles
- Students may **not** use frames.
- The roadbed must hold 2 unopened water bottles.

Have students select their best option and build their prototype.

Ask students to verify that their original model does not move side-to-side and can hold some weight. Start with 2 unopened plastic water bottles.

Have students take a picture of their first bridge prototype.

3. Explain

Have students share their model and explain how they arrived at their final design. Ask students questions like:

- What type of bridge could you build given the materials you can use?
- Why is planning prior to building important?
- What happened as you built? Did you have to change anything from the design?
- Why were you asked to verify that the model would hold the weight the water bottles?
- What step were you completing when you added the water bottles?

Discuss with students that they were completing the test and iterate phase of their design when they were adding the water bottles. Students may indicate that their bridge did not hold the weight of the water bottles.

Discuss what is learned when this happens and what students can do next. Students should recognize that testing determine if a design works as intended or not. It allows the designer to investigate what needs to change or be fixed.

Prompt students to think about how to make the design better by iterating or making small changes based on the testing and then retesting to identify additional iterations needed.

Ask students questions like:

- Why is testing the model important?
- What can we learn from testing our model?
- What is a next step to take after testing the model and learning from that test?
- When do you know that the testing and iteration phase is complete?
- Did you see ideas from other teams that you might like to incorporate in your own design?

4. Elaborate

Allow students to have time for further iteration of the design based on adding new materials.

Explain to students that a new material design has been fabricated and is available for use – the frame and any other pieces in the set. Tell students that they can use frames, but do not have to do so. Additionally, the customer has asked that the prototype must have part of the roadbed move up and down to allow a tall vessel to pass underneath. This is a new constraint that must be included.

Tell students to experiment with adding the new technology to their design. Prompt students to consider if they have to start the entire process over or just begin in the middle. Discuss the process to follow as a group.

Allow students time to redesign their bridge with the new pieces. Remind students to test and iterate on their design to ensure it meets the expectation of movement of the roadbed.

5. Evaluate

Teacher Assessment

Evaluate students' understanding of how the software and hardware must work together.

Evaluate students' understanding of the engineering design process.

Ask students questions like:

- How well did your first prototype work?
- Why was testing your design so important?
- How does an engineer move from idea to prototype?

Self-Assessment

Have students answer the following in their journals:

- What did you learn today about design engineering and the importance of testing?
- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on time management today.
- Using a scale of 1-3, rate yourself on materials (parts) management today.

LEGO® Education SPIKE Prime

Human vs. Robot

Grade 6-8 | **90 minutes** | **Beginner**

Human vs. Robot

Identify why people would use a robotic hand instead of a human hand, even when a human hand might be more efficient.

Questions to Investigate

- How do engineers determine if a task is suitable for a robot to complete?
- How can data be used to support a claim?

Materials Needed

- SPIKE Prime sets
- Device with the SPIKE App installed
- Student journal
- Sticky notes
- Place to organize sticky notes – either chart paper or whiteboard
- Craft materials such as rubber bands, tape, chenille strips, etc.

Prepare

- Check to make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.

1. Engage

Present students with the opportunity to view photographs of different types of robotic arms/hands. (Examples may include welding, manufacturing, shipping, picking, sorting).

Ignite a discussion. First, ask students to observe the photographs. Discuss with their partner the observations they found by looking at the photographs. Then,

KEY OBJECTIVES

Students will:

- Identify problems that are difficult for a robot to solve.
- Use data to support a claim as to why a robot would be preferred over a human for a task.
- Use data to compare a task completed by a human and a robot.
- Provide data to show a time when a human is more effective than a robot.

STANDARDS

CSTA

2-AP-18 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.
2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
2-CS-01 Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices
CS-2-02 Design projects that combine hardware and software components to collect and exchange data.

VOCABULARY

Accessibility
Manufacturing
Categorize

individually write their observations of types of robotic arms/hands and uses on sticky notes.

Ask students to categorize their ideas into groups (i.e. transportation, medical, manufacturing, etc.).

Ask students to consider if a human could ever be faster than a robot at doing a task. Lead a discussion with students about the items they have listed on their sticky notes.

Ask students what problems are difficult for a robot to solve? Have them consider how they have categorized items. Why were robotic arms/hands used instead of human ones?

2. Explore

Students work in pairs to design and build a robotic arm and hand.

Find the **Pass the Brick Lesson** in the **Extra Resources** section of **Start.** Build and program the robotic hand. Test the robotic hand to be sure it opens and closes as designed.

Remind students to debug the model and program as needed while testing the model. Both people should be able to use the robotic hand. Modify the program as needed to make it work the best. (Students may wish to bring up other motor blocks – for example to control motor speed.)

Gather two blue rectangular frames and three 2x4 bricks for use in the relay race.

Relay Race

- Place the two blue rectangular frames about 12 inches apart.
- Place three 2x4 bricks inside one of the frames.
- Place the bricks so that the smallest surface is on the table and they are "standing."
- In your journal, have students create a chart so they can record the time it takes to move bricks using a robotic hand or a human hand. Each partner will have three tries using a robotic hand and three tries using their human hand.
- If students struggle with chart creation, an example of a chart is shown below. (RH stands for Robotic Hand; HH stands for Human Hand.)

Name	**RH 1**	**RH2**	**RH3**	**HH1**	**HH2**	**HH3**
Partner 1						
Partner 2						

Using the robotic hand, Partner 1 moves the 3 bricks from one frame to the other while Partner 2 keeps track of the time it takes. Students may need some additional time to modify the program if desired.

When moving bricks:

- Only move one brick at a time.
- The brick must stand up.
- If one falls, you need to pick it up and place it back in the first frame and move it again.
- Write the time in the journal under RH1 (Robotic Hand 1).

Switch roles.

- Partner 2 moves the 3 bricks from one frame to another while Partner 1 keeps track of the time.
- Write the time in the journal.

Repeat, with both partners having 2 more turns, writing the times in the journals under RH2 and RH3.

Next, try moving the bricks with your own hands.

- Write the time in the journal. (Human Hand 1).
- Repeat the test just like you did with the robotic hand.
- Keep track of the time required.
- Each partner gets three turns.

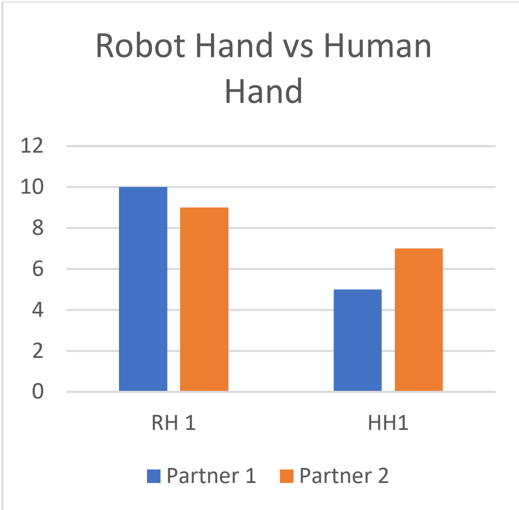
Allow students time to complete their investigation.

3. Explain

Use the information in your chart to create a graph. Compare the time required for moving 3 bricks with the robotic hand to the time required for moving the bricks with a human hand. Which took longer? Which would you consider most effective? Would you consider taking less time to be more efficient? Why or why not?

If students struggle creating a graph, have them use Time on the left axis (y-axis), and Trials 1, 2, 3 on the right axis (x-axis). Draw a bar to the time it takes for each type of hand in Trial 1, Trial 2, Trial 3. Place bars of different colors for each Trial to compare Partner 1 RH (robot hand), Partner 1 HH (human hand), Partner 2 RH (robot hand), Partner 2 HH (human hand).

- An example of a bar graph for Trial 1 is shown below as reference:



Use data from your graph to make a claim as to which hand worked better – robotic hand or human hand – for this challenge? Discuss why.

When the students have finished the chart and graph, ask students to explain their findings.

Ask questions like:

- If you could do this task better than a robotic hand, why would we ever use a robotic hand?
- How could you make the test using your hands more realistic? What if the item you were moving (instead of a LEGO brick) was extremely hot, poisonous, dangerous, etc.? What would you want to cover your hands to keep them safe?

4. Elaborate

Tell students they will improve one of the models. They can use non-LEGO materials such as tape, rubber bands, and so forth.

Each team should try to improve the grabber. They can change the LEGO elements, the programming, add additional materials, etc., to improve the performance. Allow students time to explore, experiment and iterate.

Students should share their improved designs with the class.

Take the models apart and return the elements to the correct locations.

5. Evaluate

Teacher Assessment

Evaluate the students' understanding of why robotic hands are used.

Ask questions like:

- Who would benefit from this type of design?
- How would this help someone with accessibility issues? Would it allow more people to be independent?
- Referring to the sticky notes the class created at the beginning of the lesson. What areas could a robotic hand be useful?

Self-Assessment

Have students answer these questions in their journals:

- What are some ways to use a robotic hand in any of the challenges?
- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on time management today.
- Using a scale of 1-3, rate yourself on materials (parts) management today.

LEGO® Education SPIKE Prime

Comparing Robotic Grabbers

Grade 6-8 | **90 minutes** | **Intermediate**

Comparing Robotic Grabbers

Identify how different types of robotic hands might be effective in doing certain tasks.

Questions to Investigate

- How do engineers determine if a task is suitable for a robot to complete?
- How can data be used to support a claim?
- Why should engineers consider impacts on society when designing solutions?

Materials Needed

- SPIKE Prime sets
- Devices with SPIKE App installed
- Student journals
- Water/soda bottles, paper, apple or objects about the same size and weight (The items are to mimic trash. Do not use actual trash as the trash may contain germs/bacteria/viruses.)
- Variety of gloves and mittens – e.g., leather work gloves, nylon or latex gloves, cotton gloves, garden gloves, snow mittens, oven gloves, paper bag as a glove
- Sticky notes

Prepare

- The SPIKE Prime set should be prepared and ready for student use. Students work with a partner using one set.

KEY OBJECTIVES

Students will:

- Identify problems that are difficult for robots to solve.
- Use data to support a claim as to why a robot would be preferred over a human for tasks.
- Use data to compare a task completed by a human and a robot.
- Identify positive and negative consequences for automating a human task.

STANDARDS

CSTA

2-AP-18 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.
2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
2-CS-02 Design projects that combine hardware and software components to collect and exchange data.
2-CS-01 Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices

VOCABULARY

Effective

- Pictures of different robotic arms/hands or websites containing images of robotic arms/hands. Students use the sticky notes they created in Lesson 9 regarding robot hand usage.

1. Engage

Provide students with the opportunity to view photographs of different types of robotic hands. Select one or two pictures. Discuss with students to the positive and negative consequences of moving tasks that were once performed by a person but are now completed by a robot. (Benefits might include time, safety, etc.; drawbacks might include job loss, cost of development, and so forth.)

Ask students to:

- Group robotic arms or robotic hands by function
- Select one or two pictures of robots that move things. Have students share some positive and negative consequences for moving task (e.g., people no longer get hurt – safety, less time required to do job//loss of jobs, cost of development).
- Review the sticky notes on areas where robotic hands are used.

Ask students to share what they learned through previous exploration of a robotic hand

Tell students they will explore picking up certain items that are frequently found as trash – show them the examples you have. Ask them if they have seen or used a robotic arm that would be good for picking up any of the types of items shown.

2. Explore

Students work in pairs to design and build a robotic hand. Find the **Super Cleanup Lesson** in **Invention Squad.**

Ask students to build and program the robotic arm and the two grabber attachments. Tell students to test the robotic hand with the first grabber attachment to be sure it opens and closes as designed.

Ask students to test and debug their models and programs as needed. They should modify and debug the program as needed to make it work the best.

Compare the Two Grabbers

Place the items of “trash” on the floor (Do not use actual trash, only items to represent trash.) In your journal, create a chart similar to the one shown with each partner’s name and space for each type of “trash” you have. You will be adding notes to each box, so leave plenty of room.

Hand Type	Plastic bottle	LEGO® Brick	Fruit	Flat Paper	LEGO Wheels	Paper Ball
Grabber 1						
Grabber 2						
Human Hand						

Using Grabber 1, Partner 1 should try to pick up each item and move it to a table or desk. Partner 2 keeps track of which items could be moved effectively and writes notes in the chart.

Switch roles. Partner 2 tries to pick up and move each item. Partner 1 adds notes to the chart.

Switch to Grabber 2. Partner 2 will go first. Try to pick up and move each item. Partner 1 adds notes to the chart.

Switch roles. Partner 1 tries to pick up and move each item. Partner 2 adds notes to the chart.

Each team member should try moving the bricks with his/her own hand. Remind students to use only one hand. They should keep the other hand behind his/her back.

Have students repeat the test just like they did with the robotic hand. Add notes to the chart. Both partners should take a turn.

3. Explain

Ask students to compare the effectiveness of each grabber attachment and the human hand for each type of trash. Discuss which worked the best and why.
Ask students to explain their findings. Ask questions like:

- If you could do this task better than a robotic hand, why would we ever use a robotic hand?
- Would you want to use your bare hands to clean up items around school? Around the neighborhood? In the city? Why or why not?
- How could you make the test using your hands more realistic? What could you add to your hands to make the test a fair one?
- Did you modify the program? What recommendations would you make for changes in programming or in the model itself?

4. Elaborate

Students will complete the trials again using different hand coverings.

Ask students to add additional lines to the chart in their journals. They will need lines for testing with hand coverings like gloves or mittens.

Repeat the test using hands covered with one glove/mitten. Each partner should take a turn.

Ask students to add a row to their chart for each type of glove/mitten used.

- Have students write a description of the glove/mitten and add information about how it worked when trying to pick up each item.

Using the data in their charts, have students make a claim as to the quickest way to pick up items and the safest way to pick up items.

Ask students:

- Using the data from the chart, what is the quickest way to pick up items?
- Using the data from the chart, what is the safest way to pick up items?
- What other items would be useful to compare human hands and robotic grabbers?
- What modifications to the grabbers or the programming would make them work better?

Have students take apart the models and place the elements into the correct locations.

5. Evaluate

Evaluate students' understanding of how to do a comparison between models.

Evaluate students' understanding of how to communicate in a chart the findings from a comparison.

Have students answer the following in their journals:

- What are some ways to use robotic grabbers in any of the challenges?
- What characteristics of a good teammate did I display today?
- Using a scale of 1-3, rate yourself on time management today.
- Using a scale of 1-3, rate yourself on their materials (parts) management today.

LEGO® Education SPIKE Prime

Repetitive Tasks

Grade 6-8 | **90 minutes** | **Intermediate**

Repetitive Tasks

Identify that a robot is better suited for a repetitive task than a human.

Explain how speed can affect the accuracy of a robot.

Questions to Investigate

- How do engineers determine if a task is suitable for a robot to complete?
- How can data be used to support a claim?
- Why should engineers consider impacts on society when designing solutions?

Materials Needed

- SPIKE Prime Sets
- Device with SPIKE App installed
- Student journals
- Large paper (like chart paper)
- Markers, tape
- Tape measures

Prepare

- Check to make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.
- Students will be adding markers to their robots. You may wish to limit the locations for using markers or ensure there is enough thickness of paper, so the marker doesn't bleed through to the carpet/floor.

1. Engage

Ignite a discussion:

KEY OBJECTIVES

Students will:

- Collect data around the accuracy of a robot at various speeds.
- Identify the benefits of using robots for repetitive tasks.
- Use data to support the claim that for some tasks a robot is more effective than a human.

STANDARDS

CSTA

2-AP-18 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.
2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution

VOCABULARY

Repetitive
Clockwise
Counterclockwise

Have students stand up and put their right arm straight up and their left arm straight ahead. Move their wrists on both hands clockwise and counterclockwise. Keep those positions and movements while they answer the following questions.

- Would you be happy doing the same task repeatedly for hours at a time? Why or why not?
- Would your arms and wrists be tired or maybe even hurt if you had to do this for the entire class period or even longer – like an 8-hour shift. Why? Do you think robots would be better than humans at doing repetitive tasks? Why?

Have students put their arms down. Provide students with the opportunity to view a video of different types of robots doing repetitive tasks. (For example, filling liquid into containers, moving boxes from one location onto a pallet, welding the same spots over and over, pick and place, holding an item at the correct angle).

2. Explore

Tell students they will explore robots doing repetitive tasks and how accurately their robot can work.

Students will work in pairs to design and build a robot with wheels. Find the **Going the Distance Lesson** in the **Extra Resources** section of **Start**.
Have students:

- Build the rhino robot.
- Read the directions in the lesson through **Step 04.**
- Read the Hint with the entire class. Have a discussion on circumference and how you can measure it via rotations.

Discuss with students if speed affects the distance of a rotation.

> **Note:** In the SPIKE App students change speed on the block. What they are actually doing is changing the power level of the motor which in turn affects the speed.

Ask students to create a table or chart in their journal that will show the number of rotations, the speed, and the distance the robot moves. Have students complete the table or chart as they test the robot throughout **Step 04**. Have students complete **Step 04.**

Ask students to lower the speed of the motor to 60% and run the program. Discuss how that affected the accuracy of the robot. Did they have to adjust other parameters? Why or why not?

Ask students to lower the speed of the motor to 30%, then 10% and run the program. Discuss how that affected the accuracy of the robot. Did they have to adjust other parameters? Why or why not?

Have students complete **Step 05**.

3. Explain

Discuss the data in students' charts.
Ask students:

- How did you get your robot to stop exactly in front of the brick without knocking over the brick?
- How did your software and hardware work together?
- What was difficult about this challenge?
- Explain the factors that affect the accuracy of the robot.

4. Elaborate

Have students add a marker to the rhino and redesign the model to help hold the marker in place. The robot needs to be able to trace a line with the marker on paper.

Have students use a large paper and place their rhino so it will keep the marker on the paper. They should run the program and check to see if the marker made a line. If not, students should adjust the marker.

Ask students to modify the program so the rhino will go forward, backward, forward, backward, and forward again, the exact same distance. The robot should draw the same line again and again.

Ask students to debug their program until the rhino performs the repetitive task accurately. (It will not necessarily be perfect.)

Ask students to modify the program so speed is at 100%. Compare the accuracy of this program. Does the robot still work well? How did speed affect accuracy?
Remove the marker.

Note: Do **not** take the robots apart. They will be used again.

Optional Extension:

> **Note:** Students may not be familiar with Newtons. You may wish to give them more information about the equivalent of 1 Newton by giving them an object to hold in their hand. For example, an average size apple gives a decent equivalent of the force of 1 Newton.

Have students read **Step 06**. Tell students to notice the force sensor icon and live reading is shown near the hub icon. When the force sensor is not pressed it shows 0N. That does not mean "on" it stands for 0 Newtons, which is the measurement of force.

Ask students to create a table or chart in their journal that will show the speed and if the force sensor was triggered. Have students complete the table or chart as they test the robot throughout **Step 06**. Students should try at least 4 different speeds.

> **Note:** The force sensor requires a lot of pressure to be activated. If the robot does not stop when it hits the wall (when properly programmed), students should try changing the speed.

Ask students:

- What did you learn about the force sensor? (The force sensor requires a minimum amount of force to register a touch against the wall. At a slow speed, the force may not be enough to register that it has encountered something.)

Do **not** take apart the models.

5. Evaluate

Teacher Assessment

Evaluate the students' understanding of why robots are suited to doing repetitive tasks.

Ask students like:

- How speed can affect the accuracy of a model completing a repetitive action?
- Thinking about the model, what would you change to make it work better?
- Thinking about the program, is there a way to make your coding more efficient?
- What are some positive and negative consequences of using a robot in place of a human for a repetitive task?

Self-Assessment

Have students answer the following in their journals:

- How well did your program work?
- What parts of the program did you need to debug?
- How well did the robot perform its repetitive task?
- Did speed affect the accuracy?
- What are some ways that a robot doing a repetitive action might be useful in any of the challenges?
- What characteristics of a good teammate did I display today?
- Using a scale of 1-3, rate yourself on time management today.
- Using a scale of 1-3, rate yourself on their materials (parts) management today.

LEGO® Education SPIKE Prime

Turns, Speed, & Accuracy

Grade 6-8 | 90 minutes | Intermediate

Turns, Speed & Accuracy

Identify how well a robot can do a set of turns and how speed can affect accuracy.

Questions to Investigate

- Why would an engineer need to have different ways to turn a machine or vehicle?
- How does sharing ideas improve programming?

Materials Needed

- SPIKE Prime set
- Device with SPIKE App installed
- Unbendable rod like a dowel rod, yardstick, broom stick, that two students can hold when standing shoulder to shoulder

Prepare

- Check to make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.
- Determine a place for groups to maneuver in order to move from start area to finish area while holding the unbendable rod between two students. This activity will require turns, so the area may need to be large.
- Have some items for students to use as obstacles for their robot to maneuver around. Chairs, bins, and trash cans work well.

1. Engage

Ask students to name some things that move clockwise and counterclockwise.

KEY OBJECTIVES

Students will:

- Identify the three types of turns – point, pivot, and arc.
- Identify how speed affects accuracy.
- Determine how to give credit to others for their intellectual property.
- Share programming ideas and adjust programs based on information gathered from others (feedback).

STANDARDS

CSTA

2-AP-18 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.

2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.

2-CS-01 Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices

VOCABULARY

Arc turn
Pivot turn
Point turn
Clockwise
Counterclockwise

Ask students to demonstrate moving clockwise. How do they remember which way to move because very few of them have watches or clocks?

Provide each team of students with a rod and ask the pair of students to stand shoulder-to-shoulder. They hold the stick with both hands.

Point Turns

Partner 1 is on the left and Partner 2 is on the right. Tell students they need to make the tightest right turn possible while holding the stick. Both students will move. That means the students are turning to the right (clockwise) and they are using the least amount of space possible to turn.

Allow students try on their own to figure out that one student moves forward, and one moves backward to create the tightest turn. When at least one group has figured it out, stop all groups and have them watch the teams that have it.

Explain to students they are seeing a **point turn**. Ask all teams to turn in a point turn. Ask them what "point" they are turning around. (The center of the stick they are holding.) One example of a point turn is the way a zero-turn radius mower turns.

Pivot Turn

Ask students to have the first student stay as close to the starting point (stand still) and the other move forward slowly. (It will cause the moving student and the stationary student to make a turn.)

Tell students this turn is called a **pivot turn**. Ask all teams to turn in a pivot turn. It is like a basketball player who holds one foot in place and moves the other foot around to pivot.

Arc Turn

With both students still holding the stick, ask both students to move forward. The person on the left should move extremely slowly while the person on the right moves at medium speed.

Tell students the resulting turn is an **arc turn**. This is the way the front wheels on a car moves. What direction did they move? Counterclockwise. Ask all teams to turn in an arc turn counterclockwise.

Ask the person on the right to move forward slowly while the person on the left moves quickly. This arc turn is clockwise. Ask all teams to turn in an arc turn clockwise.

Discuss the three types of turns – point, pivot and arc and have students take notes in their journals. A sheet showing the wheel movements for the turns has been given at end of the lessons and is located in the student guide.

Place students in groups of 4 – 2 teams of 2. Continuing to use the rods between two students, have Team 1 give directions to Team 2 how to navigate through an area. They should give directions such as

- Pivot turn to the left.
- Move straight 3 feet.
- Point turn to the right.

The teams should help each other to get the instructions completed correctly. Change roles and let the other team try to follow the directions given to them.

2. Explore

Students use the Rhino model, from the **Going the Distance** Lesson in the **Extra Resources** section of **Start,** to maneuver through some obstacles.

Have students **start a new program.**
First, have students make their robots go forward 3 seconds and backward 3 seconds – starting and stopping at the same location.

Write instructions for what the teams need to do on the board in pseudocode.
One example would be:

1) Program the robot to go forward
2) Make a point turn
3) Move backward
4) Make a pivot turn
5) Go forward
6) Make an arc turn

Students may ask you for more details, like which direction to turn. You may determine the answers or allow students the freedom to choose and then add details to their pseudocode.

Allow students to determine how far to go in each of the actions.
The constraints are:

- Move forward and backward must be more than 1 second and less than 3 seconds. Remind them they can use decimal numbers.
- Point and Pivot turns must be more than 90 and less than 360 degrees.
- Arc turn should cover an area more than 2 feet and less than 5 feet.
- Speed should be between 30-60%.

Create a Pattern

Students will program a robot to move in a pattern repeatedly.

Challenge students to move their robot in a square, or as close as possible. Next, change the speed to 100% and then again to 10% and determine if the robot is accurate at each speed.

When the team is successful, they should program the robot to go forward around a chair about 4 feet in front of them. They need to start in front of the chair, drive around the chair, and then come back to the beginning location. One continuous program is to be used. Have students time their runs. Taping a starting area and where the legs of the chair should be helps keep programs focused and helps students visualize the task as well.

When several teams are successful, have all teams simultaneously try to do the challenge, each around their own chair. The teams that are the fastest to accomplish the maneuver should share their programs on screen and/or explain their programming.

3. Explain

Analyze and discuss students' programs as a group.

Ask students:

- What robot behaviors did they see that really handled that part of the challenge well?
- Have students discuss the "best" programs and why they believe the programming is successful.
- Did the robot move quickly as well as accurately?
- Did the robot easily maneuver around obstacles?
- What about this challenge was the most difficult?

Discuss which programs are efficient or if more efficiency is possible. Efficiency, in this instance, will be coding that makes the robot move accurately around the obstacle and return to the starting location. Remember, efficiency of programming may mean that the "winning" robot program might be made more efficient through use of loops, stacks, or other programming components that can cause the same action with fewer blocks. However, shorter programs are not always better.

Have students write in their journals about programming ideas. Make a note about which team or person helped them learn about new programming ideas. Discuss why programming is Intellectual Property. How could you credit another person's idea for programming in your own programming? (Comments in program)

4. Elaborate

Challenge teams to take what they have learned and incorporate it into their next program.

Have students write a program to go around the chair and complete a Figure 8 before returning to the starting position. The goal is to complete the circuit in the least amount of time. Have students estimate the time it will take, then time the results.

Note: Students should keep the driving base built for use in other lessons.

5. Evaluate

Teacher Assessment

Evaluate the students' understanding of how to give credit for others programming ideas (IP).

Ask students:

What ideas and tips they incorporated into their new program?

- Is it cheating to use good ideas that others have shared?
- When would they consider it cheating or inappropriate use of others' programming?
- How much they feel their programming improved after listening to others?
- What are the three types of turns and how do they differ?

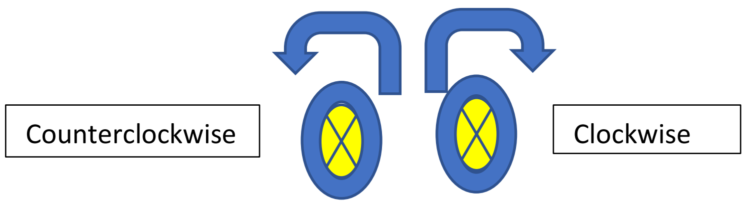
Self-Assessment

Have students answer the following in their journals:

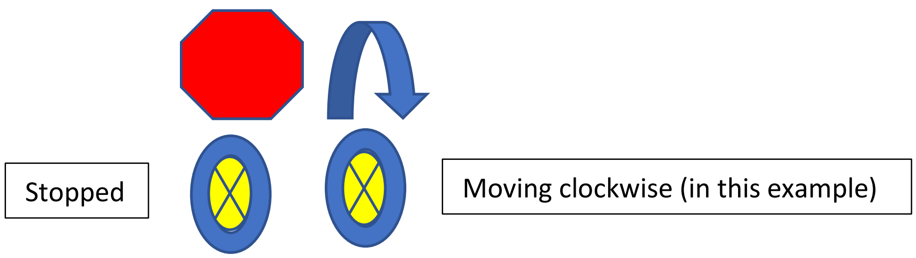
- Reflect on learning from others. What tips and ideas did you use, and did it spur other ideas of your own?
- What are some programming tips based on today's lesson that might be helpful in the challenges?
- What characteristics of a good teammate did I display today?
- Using a scale of 1-3, rate yourself on time management today.
- Using a scale of 1-3, rate yourself on their materials (parts) management today.

Types of Turns

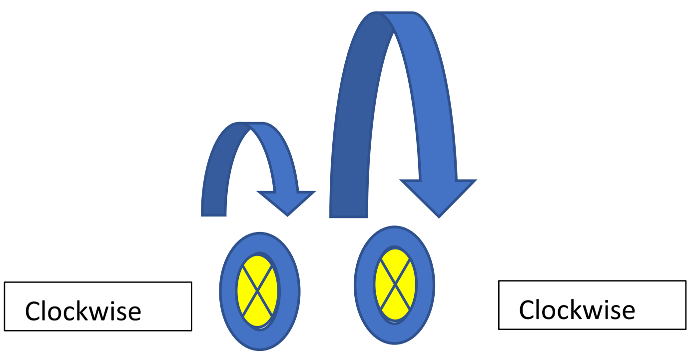
Point - one wheel forward and the other wheel backward



Pivot – one wheel stopped and the other wheel turns



Arc – both wheels turn in the same direction, but at different speeds



LEGO® Education SPIKE Prime

Uphill Climb

Grade 6-8 | **90 minutes** | **Intermediate**

Uphill Climb

Graph and analyze data created data.

Questions to Investigate

How do engineers create and analyze data to learn more about energy?

Materials Needed

- SPIKE PRIME Set
- Device with SPIKE App installed
- Student journal
- A piece of cardboard for a ramp

Prepare

- Check to make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.

1. Engage

Students compare the energy needed to roll on a flat surface compared with up an incline.

Have students get two wheels and connect then with an axle. Ask students to roll them on a flat surface. The roll them up a slight incline (ramp).

Ask students to explain how the wheels roll differently in speed and length.

Change the angle of the incline to be steeper. Ask students to predict how the wheels will behave when they try to roll them up the steeper incline. Allow students to roll the wheels and then ask them to explain the difference in behavior.

2. Explore

Students will build a motorized bike and create a graph on the energy needed.

KEY OBJECTIVES

Students will:

- Build and program a robot to move uphill autonomously.
- Create data around energy.
- Graph and analyze data.

STANDARDS

CSTA

2-CS-02 Design projects that combine hardware and software components to collect and exchange data.

2-DA-09 Refine computational models based on the data they have generated.

VOCABULARY

Incline

Potential energy

Ask students to launch the **This is Uphill** lesson in the **Training Trackers** unit. Have students complete Step 01. Watch the video and answer the questions.

Have students complete **Steps 02 and 03**. Build the bike. Run the program and answer the questions. Ask students to look at the graph carefully. They may want to repeat the program several times to see how the movement of the bike and the graph are related. Ask students what information the graph signifies?

3. Explain

Students will explain information shown on a graph.
Ask students questions like:

- Explain the information shown on the graph – what do the two lines represent?
- Why isn't the angle read by the Hub "0"?
- Why is there a weird curve at the beginning of the power line?

4. Elaborate

Students test a model going up and collect and analyze the data.

Ask students to complete **Steps 04 and 05**.

Have students answer the following in their journals:

- What is happening to the motor's power as the bike goes uphill?
- Where on the graph is the bike gaining potential energy?
- On the current (final) graph, identify where your bike was going up an incline, going straight, and going down an incline.

Allow students to investigate further as time allows.

5. Evaluate

Teacher Assessment

Evaluate the students' understanding of how the data on a graph can be analyzed.

Self-Assessment

Have students answer the following in their journals:

- What characteristics of a good teammate did I display today?
- Using a scale of 1-3, rate yourself on time management today.
- Using a scale of 1-3, rate yourself on their materials (parts) management today.

LEGO® Education SPIKE Prime

Mini-Challenge: Design for Someone

Grade 6-8 | 90-135 min | Intermediate

Mini-Challenge: Design for Someone

Design and program a prosthetic hand.

Questions to Investigate

- How do engineers create devices to help those with different abilities?
- Why do engineers share ideas with others?

Materials Needed

- SPIKE Prime Sets
- Device with SPIKE App installed
- Student journals
- Sticky notes or small colored paper
- Markers

Prepare

- Check to make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.

1. Engage

Ignite a discussion about what a prosthesis is and who might be in need on one.

There are videos available that show several different types of prosthetics for people and for animals.

Let students research and share several examples.

2. Explore

Students will plan, create and program a prosthetic device.

KEY OBJECTIVES

Students will:

- Design and program a prosthetic hand.
- Sketch your idea and use pseudocode to assist with writing code.
- Present your model and ideas to the class.

STANDARDS

CSTA

2-CS-01 Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices

2-CS-02 Design projects that combine hardware and software components to collect and exchange data.

2-CS-03 Systematically identify and fix problems with computing devices and their components.

2-AP-17 Systematically test and refine programs using a range of test cases.

2-AP-18 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.

VOCABULARY

Entrepreneur

Prosthesis

Students should open the **Design for Someone** lesson in the **Invention Squad** unit. Complete **Steps 01 and 02**. Allow students plenty of time to brainstorm.

Brainstorm
Brainstorm different ideas for a prosthetic device and who could benefit from it.

In their small groups, brainstorm several ideas for creating a prosthetic device, drawing from ideas used in previous lessons. Each group should come up with unique ideas that support their specific needs.

Design and Choose the Right Idea
Students will design, build, and program a prosthetic device. Students should create a sketch of their building idea and a flowchart or pseudocode of their programming idea.

Test and Iterate
Allow time for students to test and analyze their idea as they go, making improvements where needed. Students should test and evaluate their designs against the design criteria set and their flowcharts as they started making their solutions.

Ensure students use sketches and photos of their models to record in their design journey.

3. Explain

Review the model for providing feedback with students.

Explain to students the following guidelines for giving feedback. Consider posting the guidelines for student reference.

- Feedback is not doing something for someone else.
- You should not rebuild a model for someone else.
- You should not type into someone's program.
- You should ask questions of each other.
- You should share your ideas and show your own programming, explaining why and how you did something.
- You should be encouraging and helpful to others and not provide negative or mean comments.

Ask students to share their initial ideas with the class. Ask students not to judge the ideas but rather add new ideas as they think of something so more ideas are generated.

4. Elaborate

Students will create a prototype and program it. They will then modify and choose the best design and program.

Have students complete **Steps 03 and 04**. These steps will take time. Allow and encourage students who are struggling to move around the room and look at other groups work. Encourage students to freely share their ideas and how they created or programmed a model.

Have students share their final prototype with the class, explaining what it is good at and what they would work to improve if they had more time.

5. Evaluate

Teacher Assessment

Evaluate the students' understanding of what a prosthetic hand is and what it should be able to do.
Evaluate the students' ability to work together and to create and test a prototype – not if it works perfectly.
Evaluate the students' ability to share ideas and help each other.

Self-Assessment

Have students answer the following in their journals:

- What characteristics of a good teammate did I display today?
- Using a scale of 1-3, rate yourself on time management today.
- Using a scale of 1-3, rate yourself on their materials (parts) management today.

Connecting to Careers: Health Science and Agriculture, Food, & Natural Resources

6-8 | 90 minutes | Beginner

Connecting to Careers: Health Science, Agriculture, Food, & Natural Resources

In this lesson series, students will have the opportunity to explore and research careers.

Prepare

Prior to starting the lesson, prepare the following:

- Set aside enough LEGO® bricks for students' models.
- Make sure you have enough devices and access to the Internet for student use during this lesson.
- Make copies of the handouts (if desired) or place into a digital platform for student use.
- Prepare images of different jobs. There should be a variety of images to show different jobs within the 16 career clusters. (Example for Architecture & Construction: Architectural drafter, engineer, carpenter, engineer, electrician, HVAC technician, painter, environmental designer, etc.).

Vocabulary: Career, Career Cluster, Qualifications, Skills, Education, Knowledge

1. Engage

Ignite a discussion with students. Ask questions like:

KEY OBJECTIVES

Students will:

Articulate their personal interests and goals.
Relate their personal interests and goals into possible career pathways.
Explore various careers in career pathways.

STANDARDS

Career Ready Practice 10- Plan education and career path aligned to personal goals. (CCTC)

Present the images of the jobs in the area of Agriculture, Food and Natural Resources and Health Science to students—as a class, decide how you would categorize the jobs. Ask questions like:

- What jobs belong together?
- What kind of similar skills do these different jobs use?
- What kind of environments are associated with these jobs?
- Do any of these jobs interact or rely on one another? If so, how?

Ask students to think about what interests them. What kind of job(s) would they like to have in the future?

2. Explore

Students will be assigned two career clusters each time career connections lessons are taught. By the end of the course, students will have explored all 16 career clusters.

There are 16 career clusters:

- Agriculture, Food & Natural Resources
- Architecture & Construction
- Arts, A/V Technology & Communications
- Business Management & Administration
- Education & Training
- Finance
- Government & Public Administration
- Health Science
- Hospitality & Tourism
- Human Services
- Information Technology
- Law, Public Safety, Corrections & Security
- Manufacturing
- Marketing
- Science, Technology, Engineering & Mathematics
- Transportation, Distribution & Logistics

Students previously studied careers in the Education & Training and Human Services, Marketing and Arts, A/V Technology & Communications, Information Technology and Law, Public Safety, Corrections & Security pathways. The two career pathways we are studying today are:

- **Agriculture, Food and Natural Resources**
- **Health Science**

Ask students to think about the lessons they completed in Unit 4. They compared robot capabilities and human capabilities. They compared different versions of robotic hands. They observed the use of energy, gathering data on energy and accuracy and speed. How do these relate to the two career pathways? For example, prostheses are part of health science. Tractors use various types of turns when plowing/planting/harvesting fields. Use of robotic hands for pollination or harvesting and use of data collection by robots is done in some careers within natural resources.

Ask students to brainstorm as many jobs as they can within each career cluster. Have one person write down the jobs as they are named.

In small groups of 4, students complete online research to find out more about each career clusters. Allow time to get information about the jobs that included, especially jobs they had not heard of or had not thought of in their brainstorm.

Students should also research:

- Skills needed
- Forecast of future job openings (and current)
- Education levels, certifications, licenses, apprenticeships, etc. required

In their group, students create a visual representation (build a model) of one of the two career clusters. They will build one or more physical models with LEGO bricks. Each person can build a model, or the group can create one large model that represents all ideas about one of the career clusters.

Each group will be responsible for a quick one-minute presentation of their LEGO model(s) to explain how it represents their career cluster.

3. Explain

When the students have finished building, allow each group to present their model.
Ask students:

- Tell us about your LEGO build.
- What kind of interests would these career clusters have?
- What are some of examples of jobs in these career clusters?
- Can you explain the difference between a job and career cluster?
- How are the skills for these career clusters similar to one another?
- What interests (or does not interest) you about this career cluster?

4. Elaborate

Ask students if there were any jobs in today's career cluster that were interesting to them.

What skills used in Unit 4 lessons would be helpful in any of the jobs in these career clusters?

- What skills were required in both pathways? Which of these skills do you have?
- Build a model of your greatest strength and explain it to your partner. Then, write a short paragraph about your strength in your journal.

5. Evaluate

Evaluate the students' skills development by observing if they:

- Articulate their personal interests and goals.
- Relate their personal interests and goals into possible career pathways.
- Explore various careers in career pathways.

LEGO® Education SPIKE™ Prime

Sensors Trigger Reactions

Grade 6-8 | 90 minutes | Intermediate

Sensors Trigger Reactions

Utilize sensors to trigger reactions through conditional statements.

Questions to Investigate

- How do engineers plan computer programs for a design solution?
- How do engineers design complex robotic systems?

Materials Needed

- SPIKE Prime set
- Device with SPIKE App installed
- Different colors of ½ inch wide tape or colored paper (black, red, yellow, blue) Some colors are not recognizable by the color sensor. Please check to see your choice is recognized. **Hint**: Try placing multiple layers of the same color tape if one layer is not recognized.

Prepare

- Check to make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.
- Students will use colored lines as targets for the robot to identify and then react. You may wish to locate black and colored (red/yellow/blue) lines in the classroom or you can allow students to choose where to locate the lines.

KEY OBJECTIVES

Students will:

- Write pseudocode to design a program.
- Use comments to explain code.
- Use conditional statements in a program using sensors and motors.

STANDARDS

CSTA

2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms.
2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
2-AP-18 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.
2-CS-02 Design projects that combine hardware and software components to collect and exchange data.

VOCABULARY

Conditional

1. Engage

Students will be placing bricks and minifigures in exact locations based on the conditional statements given.

Have students build the two minifigs from the set. Choose one each of five different colors of 2 x 4 bricks.

Ask students to put the bricks in any order. Then, based on the conditions of their stack, tell students:

- If the red brick is first, then stand up.
- If the green brick is last, then raise your right hand.
- The green brick is between the blue and yellow brick, say “green between.”

Discuss this experience together as a group. What was happening? Students were responding to a condition.

Explain to students a condition is requirement that must be met to make a statement true. Using a Boolean expression, a condition is evaluated as either true or false. “The first brick is red” if evaluated as a true statement, would make you stand. If “the first brick is red” is evaluated as false, then you keep your seat.

Ask students to reset the elements, then create a new stack. Explain the new rule to students: if the stack does not meet the condition, you will be given an else.

Have students review the new stack against the following conditions:

- If the first brick is red, add the minifig Kate to it, else take away the last brick
- If the second brick is green, place the minifig Kyle next to the stack on his back, else place Kyle on his front.
- If the last brick is blue, place a yellow brick behind it, else remove the last brick.

Discuss how this experience was different from the first.

Ask students to reset the elements using the following sequential steps:

- Place the red brick on the table so the long side is facing you.
- Add a green brick turned sideways, overlapping in the center of each brick.
- Place a blue brick on top of the green brick connecting only two studs on the right center of the green brick and two studs on the short side of the blue brick. The blue brick will have its long side facing you.
- Add a yellow brick to left end of the blue brick connecting 4 studs on each brick.
- Add the magenta brick so the studs face down directly on top of the yellow brick.

Discuss any differences in the models and come to an agreement on placement of the bricks.

Ask students questions like:

- What wording caused the differences?
- How could you improve the instructions?
- How does sequential programming works?
- Discuss ways conditional statements can be used in programming.

2. Explore

Students will use a conditional statement to make a motor turn on and off and change speed.

Consider modeling sketching and writing pseudocode in the journal before the students begin this activity.

Explain to students that before creating code, many programmers determine what they want to code and engineers make sketches of what they want to build.

Have students make sketches in their journals of the hub, motor, and sensor. Then, write pseudocode in the journals before programming.

Ask students to **start a new program**, attach a motor to Port C of their hub, and add an axle to the motor and a brick or gear to the other end of the axle. Students should program the motor to start slowly and get to maximum speed over a total of ten rotations.

Provide time for students to investigate.

Ask students to modify their program. Students should write pseudocode in their journal to stop the motor and then change the direction of the motor after 5 rotations. Allow students time to program their model and investigate the new program.

Tell students that comments should be used to help them remember what more complex or long programs represents. Ask students to add a comment (Right click on the block and choose add comment) above the stop motor line of code. Write “now stop and reverse direction.”

Explore Distance

Ask students to add the Distance Sensor to Port D on the hub.

Ask students to place the distance sensor so it is facing up. Students should create a program that will turn the motor on when the distance sensor sees an object less than 10 cm away. Remind students to use comments to explain the program.

3. Explain

Discuss the programs with students.
Ask students about the experience with the distance sensor:

- How does the sensor work?
    - If they say light, ask them if they know the word sonic. What do they associate with sonic? Try to draw students to the conclusion that sonic derives from sound. Ultrasonic sensors use sound with a frequency that humans cannot hear.

Explain to students:

- One side of the ultrasonic sensor sends out a sound, the other side receives it back. The placement and the orientation of the ultrasonic sensor makes a difference as to its accuracy.
- The sound is sent out and received back, the time it takes from when a sound leaves to when it is received back determines the distance from an object.

Ask students to move their hands so that only one side of the distance sensor sees it. Ask students:

- What happens when you play the program now?

Tell students they need to remember that orientation of the sensor will be important when they use it in their challenge model.

Ignite a discussion. Have students give real-world examples of uses of distance/ultrasonic sensors.

4. Elaborate

Ask students how to duplicate blocks (Right click on the block and choose duplicate). Duplicate the distance sensor program.

Ask students to add a color sensor to Port E and attach a second motor to Port A. Direct students to:

- Find the green and the blue bricks.
- Modify one of the distance-sensor programs.
- Change the program so the A motor will run when the color sensor senses green and the C motor will run when the color sensor senses blue.

Ask students to add additional actions when the color sensor senses either green or blue.
Direct students to:

- Add the "play sound until done" block using the "Win" sound after green is sensed.
- Add the play sound until done block using the" Boop Bing Bop" sound after blue is sensed.
- Use comments to explain the conditions set in the program.

Ask students what happens when the color sensor senses a color other than green or blue. Ask them to explain that behavior.

Have students write real-world examples of using a color sensor in their journals.

Gyro Sensor

Challenge students to try the gyro sensor. Direct students to:

- Duplicate the color sensor program.
- Modify the program so the trigger for the actions is the gyro sensor in the brick.
- If the front of the hub is up, then the C motor runs clockwise, and the “Dun Dunn Dunnn” sound plays.
- If the back of the hub is up, then the A motor runs counterclockwise, and the “Laughing Baby” sound plays.
- Use comments to explain the program.

Ask students to explain how the gyro sensor works.

- The gyro sensor measures angular or rotational movement, and the accelerometer feature detects the orientation of the hub.

Ask students to write real-world examples of a gyro sensor in their journals.

5. Evaluate

Teacher Assessment

Evaluate the students’ understanding of how an ultrasonic sensor works. Ask them what the ultrasonic sensor uses to determine distance. (sound)

- Ask students to compare how sensors can be used in a program.
- Ask students to explain the relationship between speed and accuracy.

Self-Assessment

Have students answer the following in their journals:

- How using a distance or a color sensor might be helpful in one or more of the final challenges.
- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on your time management today.
- Using a scale of 1-3, rate yourself on your materials (parts) management today.

LEGO® Education SPIKE™ Prime

Sensors and Data

Grade 6-8 | **90 minutes** | **Intermediate**

Sensors and Data

Utilize an ultrasonic sensor and collect and analyze data.

Questions to Investigate

- How and why do engineers use sensors to gather data?

Materials Needed

- Spike Prime set
- Device with SPIKE App installed

Prepare

- Check to make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.

1. Engage

Students will be placing bricks and minifigures in exact locations based on the conditional statements given.

Have students choose two each of six different colors elements and 1 blue frame.

Direct students to complete the following task:

- Working with a partner, write a condition. For example, if the element has 8 studs, put it in the frame. All elements with 8 studs would be inside the blue frame and all the remaining elements would be outside the blue frame.
- Look at each element. Evaluate the condition. If the condition is true, place the element in the frame. If the condition is false, place the element outside the frame.

Explain to students that this is an example of looping. By going through each of the elements, you are seeing if the condition is true, which is repeating the same function over and over.

KEY OBJECTIVES

Students will:

- Write pseudocode as a planning tool.
- Use conditional statements in a program using sensors and motors.
- Collect, graph, and analyze data.

STANDARDS

CSTA

2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms.
2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
2-AP-18 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.
2-CS-02 Design projects that combine hardware and software components to collect and exchange data.

VOCABULARY

Conditional

Have teams check each other's work and discuss.

After students reset the elements, move them to the second experience. Direct students to complete the following:

- Create a new condition for the frame. For example, the frame represents "The element has more than 4 holes."
- Evaluate the condition. If the red piece has more than 4 holes, then place the piece in the frame, else put it outside the frame.
- Evaluate the condition. If the yellow piece has 4 holes, place the piece outside the frame; else it would go inside the frame.
- Work with a partner, write two conditionals. Each partner takes a turn writing a condition for the frame. Then, the partner who did not write the condition, evaluates the condition, then places the elements where they belong. Check each other's work and agree on where the elements belong.
- Have another team execute what should happen after evaluating the conditionals.

Discuss how conditional statements affect the way a program works.

- What is the difference between evaluating a conditional and executing the steps?
- What are ways conditional statements are used in programming?

2. Explore

Have students find their partners and their SPIKE Prime set. Students should find **Training Camp 2** in the **Competition Unit**.

Ask students to build the driving base and add **only** the distance sensor. Have them complete only **Steps 23 through 29** of **Tools and Accessories** build instructions. Tell them **not** to add the arm or the touch sensor **or** build the stand with bricks. (Skip steps 1-23 and 30-40.)

Once all students have built the driving base and added the sensor, discuss how the distance sensor works. The distance sensor is an ultrasonic sensor. Show students that their device shows live data from the ultrasonic sensor at the top of the programming palette next to the icon of the hub.

Students should write pseudocode to have the driving base move forward and stop 10 cm from the wall.

Ask students to write their program for their driving base to stop 10 cm from a wall. Students should test their program worked correctly.

Check in with students before allowing them to move to the next part of the lesson.

Completing Trials
Have students create a chart in their journals. If they struggle you can show them this chart as inspiration.

	100%	60%	40%	20%	10%
Trial 1 distance from wall					
Trial 2 distance from wall					
Trial 3 distance from wall					
Average distance from wall					
Programmed distance	10 cm	10 cm	10 cm	10 cm	10 cm
Difference from programmed distance					

Ask students to complete the following tasks:

- Change the program so the robot moves at 100% speed.
    - Test it three times.
    - Write the distance from the wall when it stops after each trial.
- Change the program so the robot moves at 60% speed.
    - Test it three times.
    - Write the distance from the wall when it stops after each trial.
- Change the program so the robot moves at 40%, 20%, and 10% speed.
    - Test it three times at each speed level.
    - Write the distance from the wall when it stops after each trial.

Ask students to average the three numbers representing the distance from the wall when it stopped at each speed. Write the averages on their charts.

Ask students to subtract the average distance from the distance in the program (10 cm). Write their answers on their charts.

3. Explain

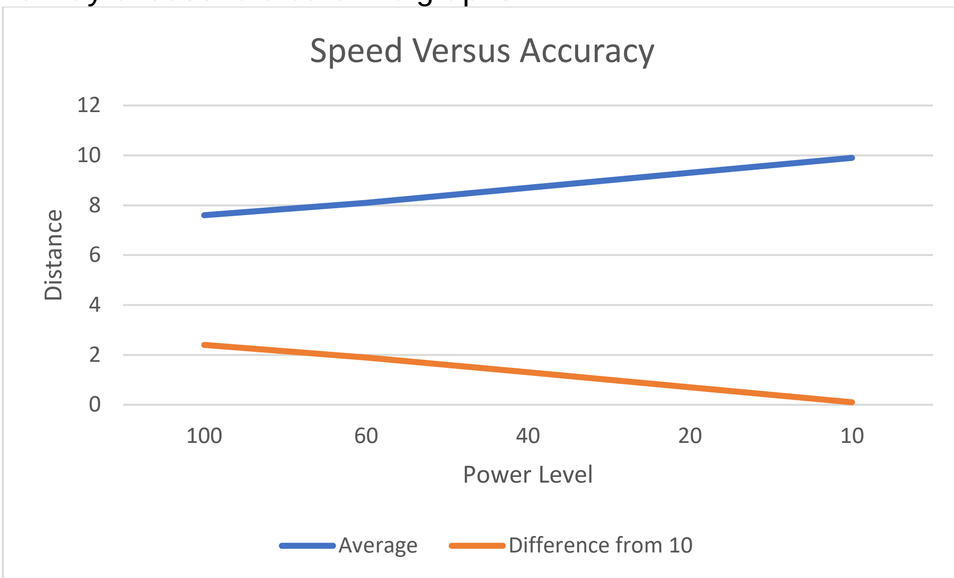
Analyze and discuss the data.
Ask students:

- What conclusions can you draw?
- Consider discussing how speed affects accuracy. [Speed negatively affects accuracy because it takes time (part of a second) to send a message from the sensor to the computer (hub) and more time (part of a second) for the computer to send a message to the motor.]
- What was difficult about this challenge?

Remind students of the delays in their unplugged activities.

4. Elaborate

Ask students to graph the information by creating a line graph. An example of a line graph is shown below if students need assistance. This graph shows both lines. Students may choose to create two graphs.



Have students complete the following tasks:

- Draw a line showing the averages. In another color draw the line showing the differences from program requirements.
- Analyze the results. What conclusions can you draw?

Discuss conclusions together as a class

Note: Do not take apart models. They will be used for the next lesson.

5. Evaluate

Teacher Assessment

Evaluate the students' understanding of how an ultrasonic sensor works.

- Ask them what the ultrasonic sensor uses to determine distance. (sound)
- Ask students to explain the relationship between speed and accuracy.

Self-Assessment

Have students answer the following in their journals:

- How using a distance or a color sensor might be helpful in one or more of the final challenges.
- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on your time management today.
- Using a scale of 1-3, rate yourself on your materials (parts) management today.

LEGO® Education SPIKE™ Prime

Dance to Debug

Grade 6-8 | **45 minutes** | **Intermediate**

Dance to Debug

Students will investigate strategies for debugging programs.

Questions to Investigate

How do software engineers identify problems within a program?

Materials Needed

- SPIKE Prime sets ready for student use. Prior to the first lesson, please visit the following website for help with set up, kit organization and SPIKE App https://education.lego.com/en-us/start/spike-prime/intro
- Devices with the SPIKE App installed.
- Student journals

Prepare

- Ensure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.

1. Engage

Create a plan for testing programs to identify and fix bugs.

Spark a discussion with students on how to fix problems. Ask students to think of a time when they had an issue or problem or a time that they had an experience that did not turn out as expected.

Discuss what could have helped change the situation. Ask questions like:

- What steps could you have taken upfront to have a better outcome?
- What could you have done when you realized there was a problem to help change the outcome?

Discuss students' ideas for troubleshooting their experience and list ideas for how to troubleshoot. Share out the steps for identifying bugs or post in the classroom for all students to reference.

KEY OBJECTIVES

Students will:

- Program a robot to react to color.
- Program movement that is rhythmic.
- Identify a problem and debug the program.

STANDARDS

CSTA

2-CS-03 Systematically identify and fix problems with computing devices and their components.
2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
2-AP-17 Systematically test and refine programs using a range of test cases.
3A-CS-03 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
3A-AP-17 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

VOCABULARY

Debugging

Steps to help identify bugs:

- Plan the program by creating a pseudocode or flowchart
- Document the program by using comments within the program
- Test your program with all types of data that are relevant or expected to be used
- Test your program with types of data that are not expected to be used or outside the range expected

2. Explore

Students will use a break dancer model to investigate identifying bugs and fixing them.

Students should build the **Break Dancer** model in the **Life Hacks Unit** of the SPIKE App. Complete **Steps 01 and 02**.

Ask students to use the program given on page 03 and 04. Running the existing programs will allow students to confirm there are no current errors in the program.

Adding to a Working Program

Students will identify strategies to find bugs when adding to a program.

Ask students to modify their program so the dancer only moves when the color sensor senses a color according to the following assignment. You can use this chart or work together to create your own.

Blue	Move legs only
Green	Move arms only
Yellow	Move both arms and legs slowly
Violet	Move both arms and legs quickly

Allow students time to create their new program. Students should test their program to make sure it works. Remind students to utilize the process discussed in the Engage section to test the program for bugs.

Ask students to record all errors and debugging activities in their journal.

3. Explain

Discuss the program with students and where they found bugs (errors).
Ask students questions like:

- What bugs did you find in your program?
- Did you receive any error messages in the console? If so, how was the error message helpful in locating and debugging the program?
- How did you test the program for the expected values? And the unexpected ones?
- Could any of the problems been with the model? Why or why not?

4. Elaborate

Challenge students to create a program that has their dancer moving to a beat using their color sensor to control movements.

Ask students to utilize their program to create a new dance that has their dancer model moving to a beat. Students can play a song or just hum a beat. The dancer should move fast and slow to the beat which can be controlled when the students hold each color brick up to the color sensor.

Allow students time to investigate the program to see if any changes are needed. Then students should create their movements to match the beat.

Ask students to share their final movements with the class. Consider having a dance party for all the models at the same time.

5. Evaluate

Teacher Assessment

Discuss the program with students. Ask students questions like:

- How does testing help identify bugs in your program?
- Why test your program for expected values and unexpected ones?

Self-Assessment

Have students answer the following in their journals:

- What did you learn today about identifying and fixing bugs?
- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on your time management today.
- Using a scale of 1-3, rate yourself on your materials (parts) management today.

LEGO® Education SPIKE™ Prime

Maze

Grades 6-8 | 90-135 minutes | Intermediate

Maze

Students will explore how motors and sensors work together to enable a robot to traverse a maze.

Questions to Investigate

- How can sensors be used to improve speed and accuracy?

Materials Needed

- SPIKE Prime sets ready for student use
- Devices with the SPIKE Prime App installed
- Blue painters' tape or blue lines drawn on paper
- Red or green tape or lines drawn on paper
- Student journals
- Ruler

Prepare

Ensure SPIKE Prime hubs are charged, especially if you are using them with Bluetooth.

1. Engage

Ignite a discussion about beating a maze. Mice can learn to go through a maze quickly to get to the cheese or other treat. Consider sharing images or videos of mice or others completing mazes as a reference for the discussion.

Ask students to consider how to program a robot to go through a maze rapidly when using sensors?

2. Explore

Students will create a maze and a robot to autonomously traverse the maze.

KEY OBJECTIVES

Students will:

- Create a maze and program a robot to autonomously navigate it.
- Program a model to move safely and accurately using sensors and motors.
- Use iteration and debugging skills.

STANDARDS

CSTA

2-CS-02 Design projects that combine hardware and software components to collect and exchange data.
2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms.
2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution
2-AP-17 Systematically test and refine programs using a range of test cases.
2-AP-19 Document programs in order to make them easier to follow, test, and debug.

VOCABULARY

Traverse
Autonomous

Challenge students to create a simple maze that requires the use of ultrasonic and color sensors to autonomously traverse the maze. Start with no more than 3 turns required. Students will be challenged to be accurate within the maze. Students should work in teams of 4 or 6 to create the mazes.

Have students use tape to create a simple maze. Then, have the teams program their robots to go through the maze they created. Each program must include using a color sensor.

Direct students to open a new project in the SPIKE Prime App. Have them name their projects Maze Challenge. Ask students to build a robot that contains a color sensor in the down position. In the **Build** section, choose **Driving Base 3**. Build the model and add the color sensor module.

Ask students to program their robot to move through the maze. First, tell students to write detailed pseudocode. Remind students to use comments to document each part of the program as they write.

Direct students to complete their maze with the following hints:

- Start with the entrance to the maze – the first movement.
- Work on one movement at a time
- Make sure they can maneuver the first line or obstacle before moving forward.
- Add a new section to the program.
- They should continue moving through the maze until each team has a working program so the robot moves autonomously through the simple maze.

Note: For teams that finish quickly, have them create a more difficult maze and then program their robot to autonomously maneuver through it.

3. Explain

Discuss with students what they found when creating a program to maneuver through a maze.

Ask students questions like:

- How did the speed of the robot affect the accuracy when using sensors?
- Why did you choose to use a sensor at a specific location?
- What was difficult about this challenge?

4. Elaborate

Students will increase the complexity of the maze and add a sensor.

Ask students to find **Driving Base 2** instructions in the **Build** section of the SPIKE App. In **Tools and Accessories** add only the distance sensor.

Prompt students to increase the difficulty of the maze by adding obstacles and walls to some parts of the maze. Challenge students by requiring the use of the distance sensor to avoid a wall and cause a reaction – a turn, reverse, and so forth.

Additionally, explain these constraints to students:

- Add sounds and display on the hub to show which color or obstacle the robot passed.
- Add a countdown on the hub light matrix before starting and an ending sound and display should be used at the completion of the maze.

Have students write detailed pseudocode before they start to modify the program. Remind students to also add comments to explain their programming.

Allow students time to adjust the maze, add the sensor to the robot, and modify their program.

Ask student to consider how does speed affect the model reaction when using the distance sensor.

5. Evaluate

Teacher Assessment

Discuss the program with students.
Ask students questions like:

- How did you program the sensors to allow the robot to move more safely?
- Why does the speed (power) of the motor affect the way the model reacts when stopping with the distance sensor?

Self-Assessment

Have students answer the following in their journals:

- What did you learn today about using sensors to maneuver through a maze?
- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on your time management today.
- Using a scale of 1-3, rate yourself on your materials (parts) management today.

LEGO® Education SPIKE™ Prime

Factory Robot

Grade 6-8 **90-135 minutes** **Intermediate**

Factory Robot

Create a robot that can help sort and ship packages.

Questions to Investigate

- How do engineers and programmers work together to build robots that can identify and move packages to specific locations?

Materials Needed

- SPIKE Prime Sets
- Device with SPIKE App installed
- Student journals

Prepare

- Check to make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.

1. Engage

Ignite a discussion. Show students pictures or video clips of robots working in factories. Discuss the common actions the robots complete.

Ask students:

- Do all robots have wheels? Why or why not?
- What types of sensors are used?
- What are different ways that packages can be shipped? (mail, trucks, rail, and so forth)

Allow students to investigate further.

2. Explore

Tell students they will be designing a factory robot that can identify and move packages.

Ask students to find the **Automated** lesson in the **Kickstart Your Business unit**.

Have them complete **Steps 01 and 02** and build the packages.

KEY OBJECTIVES

Students will:

- Design, build, and program a factory robot that can identify and move materials autonomously.
- Use pseudocode and comments to assist in writing code and debugging.

STANDARDS

CSTA

2-CS-02 Design projects that combine hardware and software components to collect and exchange data.

2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms.

2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.

2-AP-17 Systematically test and refine programs using a range of test cases.

2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-19 Document programs in order to make them easier to follow, test, and debug.

VOCABULARY

Automate

Have students read **Step 03** including the hint. Ask students to brainstorm with another team ways that the packages could be identified and how to set up the factory so that packages that are ready to be shipped come to one location for sorting. The packages are then sorted and moved to the correct shipping locations. Complete **Step 03** – Brainstorm.

Students need to set up their factories with packages and locations where the packages must start and end. Students should not build a robot nor start programming.

3. Explain

Allow students to share their factory set up.
Ask students:

- What are some ways to sort the packages for shipment?
- What are some ways to move the packages – using wheeled vehicles or using a robot that does not have wheels to move?
- How did each team set up its factory?

4. Elaborate

Students have their robots sort packages.

Have students build a robot that can identify the packages and sort them into locations. If it is a wheeled robot, it should return to the starting area to get a new package after it delivers one.

Ask students to write pseudocode prior to writing their program. Remind students to add comments as the write their program so they can debug sections of code.

All robots should indicate through sound or light when it is ready for a new package or when it has identified a package ready for shipment.

5. Evaluate

Teacher Assessment

Evaluate the students' ability to build and program a robot that can identify packages. Evaluate the students' ability to build and program a robot that move a package to the correct location.

Self-Assessment

Have students answer the following in their journals:

- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on your time management today.
- Using a scale of 1-3, rate yourself on your materials (parts) management today.

Parking Lot

Grade 6-8 | 90-135 minutes | Intermediate

Parking Lot

Utilize computational thinking to park cars.

Questions to Investigate

- How do people safely find a place to park in a parking lot?

Materials Needed

- Spike Prime set
- Device with SPIKE App installed
- Sticky notes
- Driving base with distance sensor attached
- Parking lot taped on floor

Prepare

- Check to make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.
- Write the numbers 1 through 12 on sticky notes. Each group will receive one sticky note. If you have more than 24 students in your class, write additional numbers so each team has a unique number.

1. Engage

Students will be parking their robots into given spaces.

Give each team a number from 1 to 12 on a sticky note. Tell students that this is the parking space that they have been assigned when they purchased a parking pass.

Assign each team as Main Street entrance or A street entrance. Be sure to mix it up so that lower and higher numbers are entering from each street.

Show students a picture of the parking lot.

KEY OBJECTIVES

Students will:

- Coordinate programming with another team.
- Apply computational thinking to real-life problems.
- Coordinate robots movements within constraints to obtain a goal.

STANDARDS

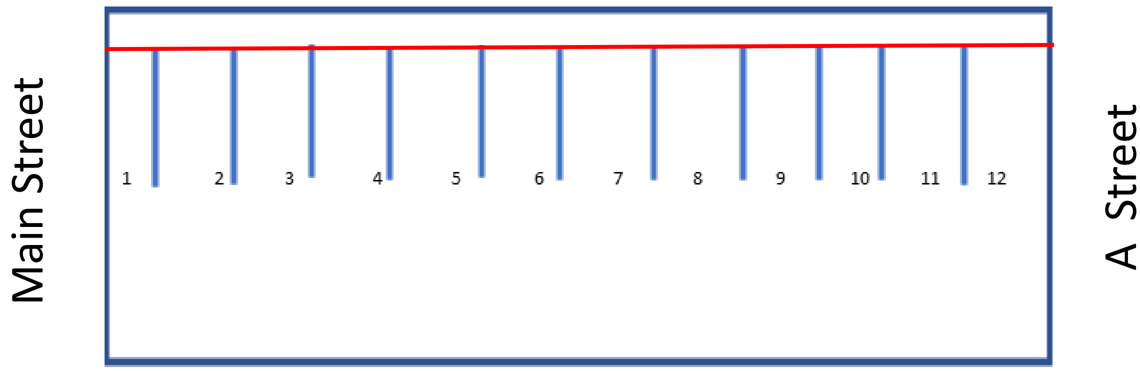
CSTA

2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms.
2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
2-AP-15 Seek and incorporate feedback from team members and users to refine a solution that meets user needs.
2-AP-18 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.
2-AP-19 Document programs in order to make them easier to follow, test, and debug.
2-CS-02 Design projects that combine hardware and software components to collect and exchange data.

VOCABULARY

Constraint

You may use this example or use the one on taped on the floor. All students will need to see the parking for class discussion, so it may be easier to show a picture on an overhead projector.



Students should have one team member line up on each side of the parking lot on their entrance streets.

Explain the constraints to students:

- Both cars must enter at the same time into the parking lot
- Neither car can stop except in a parking space
- Cars can only back up when they are backing out of a space

Discuss the challenge as a group. Identify at least 2 different scenarios that would work. Go through each pair of cars entering at the same time. Challenge students to think about:

- If the person on Main Street's parking space is to the left of the parking space for the person entering from A Street, both cars can easily enter the parking lot and park their cars.
- If the person entering from A Street has a parking space to the left of the space used by the person entering from Main Street, how will both cars park safely?

Remind students of the constraints throughout the discussion.

Have students return the sticky notes. You will need them again.

2. Explore

Students will work together across teams to make a plan for entering the parking lot.

Have students build **Driving Base 3** found the **BUILD** section. No sensors are needed.

Give each team a new sticky note (parking space number) and a street (Main or A). Have students line up again on the Main Street and A Street sides. This time their order is fixed. No changing sticky notes or places. Ask a student to write the order of each team and the parking space at each entrance to the parking lot from each street.

Have students work with the team opposite them at the parking lot. The two teams need to come up with a plan to enter and park their cars according to the rules or constraints. Remind students of the rules.
Ask students to write pseudocode to have the driving base move forward and park in the assign space. When writing the code, make sure students add comments to document their programs. Each set of teams should verify that their robots are in constant motion or they are parked in a space. If they have to back up, it can only be from a parking space.

Have students program the robots following their pseudocode.

Ask all students to line up with their robots and, in order, have two teams park their robots simultaneously. The parking lot will be full at the end. If there are more spaces that robots, you can choose to have them remain empty or you can fill empty spaces with objects or block the entrances to the spaces.

3. Explain

Discuss the challenge with students. Ask students:

- To explain how they determined how to handle the movement into the correct parking spaces. Was it easier to park in an empty parking lot or a full one? Did it matter?
- What issues were the most difficult to overcome?
- What was easy about the challenge?
- What are some ideas you might try next time to make it easier?

4. Elaborate

Students will work across teams to exit the parking lot.

Challenge students to exit the parking lot. The class has a choice to make, they can choose one of the following scenarios to empty the parking lot:

A. All robots exit to the street from which they entered in the order they entered.
B. All robots exit to the street opposite from where they entered in the opposite order in which the entered.
C. All robots exit at the same time to Main Street in a synchronous movement.
D. All robots in odd numbered spaced exit to Main Street in a synchronous movement followed by all robots in even numbered spaces exiting to A Street in a synchronous movement.

Have the class discuss and choose a scenario.

Direct students to write pseudocode and then program the robots to leave. When everyone is ready, set up the parking lot and have all robots exit the parking lot.

If time allows, consider trying more than one scenario.

Keep the robots built. They will be used in a mini-challenge.

5. Evaluate

Teacher Assessment

Evaluate the students' understanding of:

- Working together to obtain a goal.
- How a sensor can be used for safety.

Evaluate the students' ability to come up with multiple ways to solve a problem.

Self-Assessment

Have students answer the following in their journals:

- How using a distance or a color sensor might be helpful in one or more of the final challenges.
- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on your time management today.
- Using a scale of 1-3, rate yourself on your materials (parts) management today.

Mini-Challenge: Parking Lot

Grade 6-8 | **90-135 minutes** | **Intermediate**

Mini-Challenge: Parking Lot

Utilize sensors to help park cars.

Questions to Investigate

- How can sensors help people find a place to park in a parking lot?
- How can sensors be used to determine location?

Materials Needed

- SPIKE Prime set
- Device with SPIKE App installed
- Sticky notes
- Driving base
- Parking lot taped on floor with colored tape (red, green, or blue) added to the front of each space.

Prepare

- Check to make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.
- Write the colors used in the parking lot on sticky notes – one third red, one third green, and one third blue.

1. Engage

Students will be parking their robots into specific spaces.

Give each team a sticky note. Tell students that this is the color parking space that they have been assigned when they purchased a parking pass. Assign each team as Main Street entrance or A street entrance. Be sure to mix it up so that lower and higher numbers are entering from each street.
Show students a picture of the parking lot.

You may use this example or use the one on taped on the floor. All students will need to see the parking lot for class discussion, so it may be easier to show a picture on an overhead projector.

KEY OBJECTIVES

Students will:

- Use conditional statements in a program using sensors and motors.
- Apply computational thinking and sensors to real-life problems.
- Coordinate robots movements within constraints to obtain a goal.
- Coordinate programming with another team.
- Write pseudocode and give credit through comments for code borrowed from others.

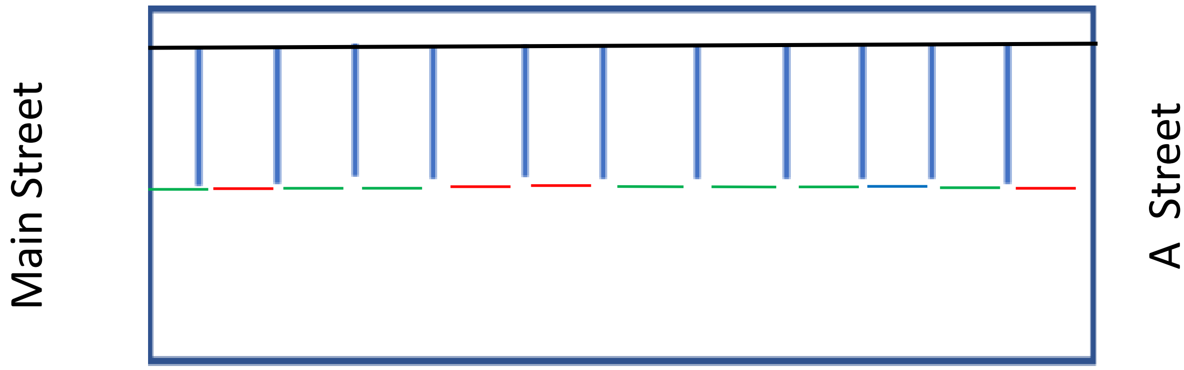
STANDARDS

CSTA

2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms.
2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
2-AP-15 Seek and incorporate feedback from team members and users to refine a solution that meets user needs.
2-AP-18 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.
2-AP-19 Document programs in order to make them easier to follow, test, and debug.
2-CS-02 Design projects that combine hardware and software components to collect and exchange data.

VOCABULARY

Constraint



Students should have one team member line up on each side of the parking lot on their entrance streets.

The constraints:

- Both cars must enter at the same time into the parking lot
- Neither car can stop except in a parking space.
- Cars can only back up when they are backing out of a space.
- Cars must take the first available space of their color.

The first student on Main Street and the first students on A Street should tell what parking space they each have. If the person on Main Street's parking space is to the left of the parking space for the person entering from A Street, both cars can easily enter the parking lot and park their cars. But what if the person entering from A Street has a parking space to the left of the space used by the person entering from Main Street.

Thinking about the constraints, how can all cars be parked safely? Discuss as a class at least two different scenarios that would work. Go through each pair of cars entering at the same time. How would the drivers work together? No part of the car can move over any of the lines – remember you are all safe drivers.

Hint: You may wish for one person from each team to physically walk through scenarios using a large parking lot marked off in a hallway.

Hint: Remind students that they will need to use the distance sensor to find open spaces as well as the color sensor to verify the color on an open space.

2. Explore

Have students get their driving bases. Add the color sensor using the instructions in **Driving Base 3** and add the distance sensor using the instructions in **Driving Base 2 Tools and Accessories** found in the **BUILD** Section.

Give each team a new sticky note with a color and a street. Have students line up again on the Main Street and A Street sides. This time their order is fixed. No changing sticky notes or places. Ask a student to write the order of each team at each entrance.

Have students work with the team opposite them at the parking lot. The two teams need to come up with a plan to enter and park their cars according to the rules. Remind students of the rules.

The constraints:

- Both cars must enter at the same time into the parking lot
- Neither car can stop except in a parking space.
- Cars can only back up when they are backing out of a space.
- Cars must take the first available space of their color.
- Do not let any part of the robot, including wires, go over any of the lines.

Students should write pseudocode to have the driving base move forward and park in the appropriately colored space. Each set of teams should verify that their robots are in constant motion or they are parked in a space. If they must back up, it can only be from a parking space.

Program the robots, documenting each section of the program using comments. Give credit for code borrowed from others through comments. Work together as two teams to perfect the parking of the robots.

Ask all students to line up with their robots and, in order, have two teams park their robots simultaneously. The parking lot will be full at the end.

3. Explain

Ask students to explain how they used sensors to find a correct parking space. Was it easier to park in an empty parking lot or a full one? Did it matter?

What issues were the most difficult to overcome?

What was easy about the challenge?

4. Elaborate

Now that all the robots are in the parking lot, it's time for them all to leave.
All robots in red spaces exit to Main Street, all robots in blue spaces exit to A Street, all robots in green spaces exit to Main Street.

The class has a choice to make, they can choose one of the following scenarios to empty the parking lot:

A. All robots in one color exit in a synchronous movement - red, blue, green.
B. All robots exit from space 1 to space 12 in order.

C. All robots exit from space 12 to space 1 in order.

Have the class discuss and choose a scenario. Then, write pseudocode and program the robot. When everyone is ready. Set up the parking lot and have all robots exit the parking lot.

5. Evaluate

Teacher Assessment

Evaluate the students' understanding of how sensors can be used to solve a problem.
Evaluate students' ability to produce multiple ways to solve a problem.
Evaluate students' understanding of how a sensor can be used for safety.

Self-Assessment

Have students answer the following in their journals:

- How using a distance or a color sensor might be helpful in one or more of the final challenges.
- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on your time management today.
- Using a scale of 1-3, rate yourself on your materials (parts) management today.

Connecting to Careers: Manufacturing and Transportation, Distribution & Logistics

Grades 6-8 | **90 minutes** | **Beginner**

Connecting to Careers: Manufacturing and Transportation, Distribution & Logistics

In this lesson series, students will have the opportunity to explore and research careers.

Prepare

Prior to starting the lesson, prepare the following:

- Set aside enough LEGO® bricks for students' models.
- Make sure you have enough devices and access to the Internet for student use during this lesson.
- Make copies of the handouts (if desired) or place into a digital platform for student use.
- Prepare images of different jobs. There should be a variety of images to show different jobs within the 16 career clusters. (Example: Truck Driver, Logistics Manager, Warehouse Worker, Supply Chain Manager, Logistics Analyst, Inventory Control, Receiving Clerk, Machine Operator, CNC Operator, Mechanic)

1. Engage

Ignite a discussion with students:

- Present the images of the jobs in the area of Manufacturing and Transportation, Distribution & Logistics to students—as a class, decide how you would categorize the jobs. Ask questions like:
    - What jobs belong together?
    - What kind of similar skills do these different jobs use?
    - What kind of environments are associated with these jobs?

KEY OBJECTIVES

Students will:

- Articulate their personal interests and goals.
- Relate their personal interests and goals into possible career pathways.
- Explore various careers in career pathways.

STANDARDS

Career Ready Practice 10 - Plan education and career path aligned to personal goals. (CCTC)

VOCABULARY

Career
Career Cluster
Qualifications
Skills
Education
Knowledge

- Do any of these jobs interact or rely on one another? If so, how?

Ask students to think about what interests them. What kind of job(s) would they like to have in the future?

2. Explore

Students will be assigned two career clusters each time career connections lessons are taught. By the end of the course, students will have explored all 16 career clusters.

There are 16 career clusters:

- Agriculture, Food & Natural Resources
- Architecture & Construction
- Arts, A/V Technology & Communications
- Business Management & Administration
- Education & Training
- Finance
- Government & Public Administration
- Health Science
- Hospitality & Tourism
- Human Services
- Information Technology
- Law, Public Safety, Corrections & Security
- Manufacturing
- Marketing
- Science, Technology, Engineering & Mathematics
- Transportation, Distribution & Logistics

The two career pathways we are studying today are:

- **Manufacturing**
- **Transportation, Distribution & Logistics**

Ask students to brainstorm as many jobs as you can within each career cluster. Have one person write down the jobs as they are named.

Ask students to think about the lessons in Unit 5. Ask students:

- How did working through a maze, creating a factory robot, and coordinating together to park cars relate to these career pathways?
- What skills did you use that would be helpful in jobs in these pathways?

In small groups of four, students complete online research to find out more about each career clusters. Allow time to get information about the jobs that included, especially jobs they had not heard of or had not thought of in their brainstorm.

Students should also research:

- Skills needed
- Forecast of future job openings (and current)
- Education level, certifications, licenses, apprenticeships, etc. are required

In their group, students create a visual representation (build a model) of one of the two career clusters. They will build one or more physical models with LEGO® bricks. Each person can build a model, or the group can create one large model that represents all ideas about one of the career clusters.

Each group will be responsible for a quick one-minute presentation of their LEGO® model(s) to explain how it represents their career cluster.

3. Explain

When the students have finished building, allow each group to present their model. Ask students:

- Tell us about your LEGO® build.
- What kind of interests would these career clusters have?
- What are some of examples of jobs in these career clusters?
- Can you explain the difference between a job and career cluster?
- How are these career clusters similar to one another?

4. Elaborate

Ask students if there were any jobs in today's career cluster that were interesting to them. Ask students to build a model that represents something that would be enjoyable in these career pathways. Allow students to share their models.

Discuss the models and ask students:

- How did the Factory Robot or Parking Lot lesson relate to these career pathways?
- What courses would be good to take in high school to help someone get a job in these career pathways?
-

5. Evaluate

Evaluate the students' skills development by observing if they:

- Articulate their personal interests and goals.
- Relate their personal interests and goals into possible career pathways.
- Explore various careers in career pathways.

LEGO® Education SPIKE™ Prime

Drone Pitch & Roll

Grade 6-8 | 90 minutes | Intermediate

Drone Pitch & Roll

Use pitch and roll to create graphs of motion.

Questions to Investigate

- How does a drone need to move to avoid obstacles?
- How do yaw, roll, and pitch affect movement?

Materials Needed

- SPIKE Prime set
- Device with SPIKE App installed
- Sticky notes

Prepare

- Check to make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.

1. Engage

Students will be acting like a drone and learning about yaw, pitch, and roll.

Have students each use 10 LEGO® elements or fewer to build a quick model that has an obvious front, back, top and bottom. It can take any shape. Give them 2 minutes or less to build.

Have the students hold their model so that the front of the model faces toward the student.

Ask students if anyone knows what the terms yaw, pitch, and roll refer to.

- Pitch tilts an object on the y-axis. Think of an airplane with the nose going up and down.
- Roll tilts an object on the z-axis. Think of an airplane tilting the wings down and up.
- Yaw tiles the object on the x-axis. Think of a car making a turn to the left or right.

KEY OBJECTIVES

Students will:

- Use conditional statements in a program using sensors.
- Apply sensors to real-life problems by analyzing sensor data.
- Use pitch and roll to move like a drone around obstacles and to create data for analysis.

STANDARDS

CSTA

2-CS-02 Design projects that combine hardware and software components to collect and exchange data.

2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.

2-AP-17 Systematically test and refine programs using a range of test cases.

3A-AP-13 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.

2-DA-08 Collect data using computational tools and transform the data to make it more useful and reliable.

VOCABULARY

Yaw

Pitch

Roll

Explain to students if they are in a plane and the pilot makes the plane turn to the right or left, it isn't the same as in a car. The plane yaws and rolls at the same time. In fact, it may pitch, roll, and yaw at the same time.

Ask students to practice using the model to move around the area of their desks. Think about how they would have to move above (over) objects, around objects, and so forth. When does the "drone" pitch? Roll? Yaw? Or use a combination?

Challenge students to move with their models as you call out a direction. Use this example movement or create your own.

- Yaw (turn the drone to right or left)
- Pitch (move the front of the drone up or down)
- Roll (move the wings of the drone up or down)
- Yaw and roll (turn to right and roll or turn to left and roll)
- Yaw and pitch and roll

Take the models apart and put the elements back into the bin.

2. Explore

Students will build a model and then modify it to add a handle.

Have students find the **Stretch with Data lesson** in the **Training Trackers unit**. Look at **Step 02**. Build the yoga ring.

Ask students to modify the yoga ring by adding a handle so they can control it like a drone. Remind students that the handle needs to allow them to easily move the drone around without breaking. Ask them to add something to indicate the front of the drone (closest to the large button).

Explain to students that they will be using their drone to "fly" over and around obstacles. The first time they fly their drone, they will fly for 10 seconds and focus on changing only the pitch. Students should slowly move the front of the drone up and down as it moves across an area to simulate going up and over an object. They should try not to move to the left or right or side-to-side, but rather slowly move the drone up and down in smooth motions.

Have students look at the programs on **Step 03**. They may need to close the lesson instructions on the right side of the screen so you can see both programs. If they click the two gray bars to close the lesson instructions, the instructions will close. They can open the lesson instructions by clicking the two gray bars.

Ask student to:

- Change "top" to "front."
- Change write "Stretch!" for "2" seconds to write "Fly" for "10" seconds.
- Add a comment to explain what you are asking the done to do.

Have students place their drone upside-down on the table so the handle is sticking up in the air. Ask them to start the program. After three seconds, they should pick up their drones by the handle and hold them so the front is facing forward. Students press the left button and prepare to “fly” the drone. Remind students to fly straight forward going up and over an object and then back down. They should move slowly for 10 or more seconds and then gently “land” and press the right button. A graph will have been created by the program.

Allow students time to investigate.

3. Explain

Discuss the graphs together and determine the peak of the flight where obstacles were flown over.

Ask students to:

- Explain pitch
- Discuss where the drone was closest to the ground and where it was over an obstacle based on the graph.

4. Elaborate

Challenge students to look at roll.

Have students change the ‘wait until “pitch” angle is +10’ to ‘wait until “roll” angle is +10.’

Tell students to follow these steps:

- They will tilt the sides (wings) of the drone down and up.
- Lower the right wing.
- Return to horizontal.
- Lower the left wing.
- Return to horizontal.
- Roll all the way over in a loop.

Remind students to be careful not to drop the drone.

Ask students to:

- Start the program.
- Begin with the drone upside down on the table.
- When it is ready to fly, press the left button.
- Go through the maneuvers and then land.
- Press the right button.

Discuss the experience. Ask students:

- What does the graph show?
- How can you identify the right and left movement?
- What does the complete roll over look like?

5. Evaluate

Teacher Assessment

- Evaluate the students' understanding of pitch and roll.
- Evaluate students' understanding of how a graph can be used to tell what happened during a flight.

Self-Assessment

Have students answer the following in their journals:

- How would using a distance or a color sensor might be helpful in one or more of the final challenges.
- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on your time management today.
- Using a scale of 1-3, rate yourself on your materials (parts) management today.

LEGO® Education SPIKE™ Prime

Drone Movements

Grade 6-8 | 90 minutes | Intermediate

Drone Movements

Analyze gyro sensor data from a drone.

Questions to Investigate

- How does a drone use yaw to avoid obstacles?
- How can you determine where obstacles were based on flight data?

Materials Needed

- SPIKE Prime set
- Device with SPIKE App installed
- Drone models from Drone Pitch and Roll lesson
- 10 feet of table space
- Materials to create objects to fly over and around

Prepare

- Check to make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.

1. Engage

Students will be acting like a drone and learning about yaw, pitch and roll. Review pitch, roll, and yaw.

Remind students:

- Pitch tilts an object on the y-axis. Think of an airplane with the nose going up and down.
- Roll tilts an object on the z-axis. Think of an airplane tilting the wings down and up.
- Yaw tiles the object on the x-axis. Think of a car making a turn to the left or right.

KEY OBJECTIVES

Students will:

- Apply sensors to real-life problems.
- Use yaw, pitch and roll to move like a drone around obstacles and to create data for analysis.
- Analyze graphs to determine movement.

STANDARDS

CSTA

2-CS-02 Design projects that combine hardware and software components to collect and exchange data.
2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
2-AP-17 Systematically test and refine programs using a range of test cases.
3A-AP-13 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
2-DA-08 Collect data using computational tools and transform the data to make it more useful and reliable.

VOCABULARY

Yaw
Pitch
Roll

If possible, show a short video clip of planes or science fiction vehicles flying through the air using roll, pitch, and yaw. After a maneuver, stop the video and ask students to explain how the pilots are controlling the movements with roll, pitch, and yaw.

Ask students to create an area on tables or desks approximately 10 feet in length and 3-4 feet in width that contains obstacles to fly around. Students will need to be able to access the left and right side of the area so that the drone can fly on one side and then turn and fly a return route using the other side.

Have students practice with the drone models used in the **Drone Pitch and Roll** lesson to be sure enough room is available to fly as close to the "ground" as possible and still maneuver around the objects.

2. Explore

Have students find the **Stretch with Data lesson** in the **Training Trackers unit**.

Ask students to:

- Look at **Step 03**.
- Look at the programs.
- Close the lesson instructions on the right side of the screen so they can see both programs. Use the two gray bars to close the lesson instructions.

Tell students they are changing the coding to use yaw angle to plot a line in addition to pitch angle so they can gather data on how the drone moves.

Tell students they need to change the program:

- The sensor inside the brick must get an orientation, so change the "set sensor orientation" block from "top" to "front".
- Change write "Stretch!" for "2" seconds to write "Fly" for "20" seconds.
- Remove wait until pitch angle = -90
- Change plot pitch angle to purple line to plot roll angle to purple line
- Add plot yaw angle to blue line underneath the plot roll angle to purple line within the repeat.
- Add comments to explain your code.

Tell students they are going to fly the drone and gather the data for analysis.
Ask students to:

- Place the drone upside on the table so the handle is sticking up in the air.
- Start the program.
- After three seconds, pick up the drone by the handle and hold it so the front is facing forward. Press the left button and prepare to "fly" your drone.
- Remind students they are flying straight forward going up, over, and around objects and then back down the other side of the table.
- Fly the route and then "land" and press the right button.

After the flights are complete, have students look at the graphs. Ask students to analyze the flight data on the graph and determine where the obstacles were avoided and where the drone turned around.

3. Explain

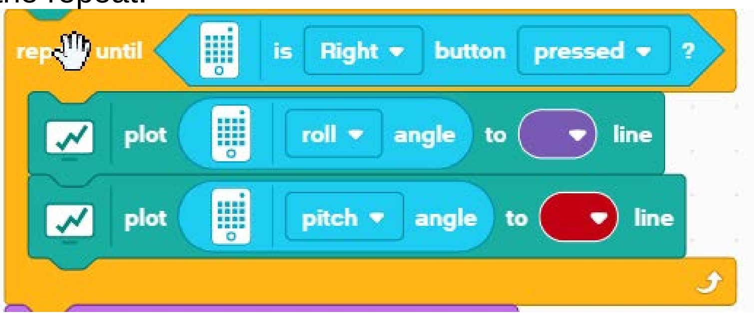
Discuss the experience with students. Ask students to explain:

- Where on the journey the drone was by looking at the graph?
- Why yaw and roll seem to work together on a turn?
- How can you tell where the turn to go back occurred?

4. Elaborate

Challenge students to now look at pitch, roll, and yaw.

Ask students to add plot pitch angle to red line underneath the plot roll angle to purple line within the repeat.



Have students repeat the flight over the table. Tell students if they want to add a roll or sudden change, they may do so. Remind students to maneuver as close to the “ground” and obstacles as they can without hitting anything.

Ask students to:

- Start the program.
- Begin with the drone upside down on the table.
- When it is ready to fly, press the left button.
- Go through the maneuvers and then land.
- Press the right button.

Have students analyze the graph. With the addition of pitch, can they determine where on the course they were at any point in time?

Ask students to identify:

- the turn-around point
- going around an object
- when the drone started to land

Ask students to consider the following:

- If this information was part of the “black box” recording what happens during the flight of a plane, how could it help to determine what happened during a crash or an inflight incident?

Discuss the experience.

5. Evaluate

Teacher Assessment

- Evaluate the students' understanding of pitch and roll and yaw.
- Evaluate students' understanding of how a graph can be used to tell what happened during a flight.

Self-Assessment

Have students answer the following in their journals:

- How were you able to determine an action of the drone by analyzing a graph of yaw, pitch, and roll?
- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on your time management today.
- Using a scale of 1-3, rate yourself on your materials (parts) management today.

Variables

Grade 6-8 | 90 minutes | Intermediate

Variables

Get an introduction to programming variables in SPIKE App.

Questions to Investigate

- How do engineers design programs that will function when data or information changes?

Materials Needed

- Spike Prime set
- Device with SPIKE App installed

Prepare

- Check to make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth

1. Engage

Have students find one 2x4 brick of each color (red, green, blue, yellow, magenta) from their set.

Have students find the parts to both minifigures. Ask students to identify how many choices they have for:

- Hair (two)
- Heads (two)
- Shirts (two)
- Legs (two)
- Bricks (five)

Have students write the pseudocode for building a minifigure on a base in simple terms. One example is shown below:

- Place a 2x4 brick in front of you.
- Place legs on top of the brick.
- Place a shirt on top of the legs.

KEY OBJECTIVES

Students will:

- Describe the function of a variable.
- Create clearly named variables.
- Program using variables after planning the program using pseudocode.

STANDARDS

CSTA

2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.

2-AP-18 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.

2-CS-02 Design projects that combine hardware and software components to collect and exchange data.

VOCABULARY

Variable

- Place a head on top of the shirt.
- Place hair on top of the head.

Ask each student to create their own minifigure based on their pseudocode.

Ask students to compare their code to their partner's – what is similar- shirt, head, hair, etc. Point out to students that they are naming the variables.

> **Note:** If students are struggling with the concepts, you can show them this chart.

Element	Choice	Choice	Choice	Choice	Choice
hair	brown	black			
shirt	dark blue	light blue			
legs	yellow	black			
bricks	red	yellow	green	blue	magenta

Discuss with the class how someone could keep track of how many times one of the variables is used to create a minifigure. As a class write pseudocode to keep track of how many times one of the variables is used to create a minifigure. Where will you repeat? How will you get through all the choices?

For example:

- Repeat 30 times: (Assuming a class of 30)
  - o Check color of brick.
    - ▪ If brick is red, add one to red score. Else
    - ▪ If brick is green, add one to green score. Else
    - ▪ If brick is yellow, add one to yellow score. Else
    - ▪ If brick is blue, add one to blue score. Else
    - ▪ If brick is magenta, add one to magenta score.

> **Note:** One could add a way to keep score of the colors of the other parts, but because two people are in each team and there are only two choices, the scores for legs, shirts, heads, and hair will all be the same.

Create a line throughout the classroom so that all students will get an opportunity to score their color brick. Start the pseudocode at one location and have all students tell the color of the 2x4 brick used. Tally the choices.

What were the final scores? What color 2x4 brick was chosen the most?
Ask students what the variables were in this activity. (brick, legs, shirt, head, hair) (Only the brick was scored, however.)

Have students put away the elements.

2. Explore

Ask students to find the **Help!** lesson in the **Invention Squad** unit.

Ask students: to build the Kiki model. Move to Step 03. Ignore the What's Kiki's problem information and play the programs, first trying the left button and then trying the right button.

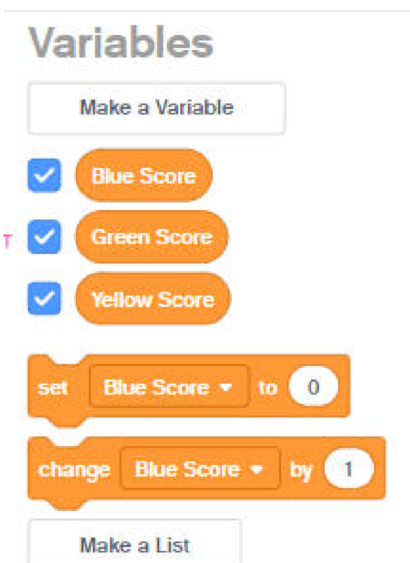
Ask students:

- What happens? (sounds are played when a color is sensed)
- What are the variables? (the color of the bricks)

Ask student to open the full palette of blocks by clicking the three dots at the bottom of the palette and choosing SPIKE™ Prime.

Ask students to choose **VARIABLES** from the palette on the left.

- Choose **Make a Variable**.
- Create three variables. Name them Blue Score, Green Score, and Yellow Score.
- Notice that the score boxes have appeared on the right side of the screen.
- What other blocks have become available under Variables?



Oval shaped reporter blocks named Score for each color

Stack block titled set (pull down menu) score to 0 (Use this block to set or reset the initial number)

Stack block titled change (pull down menu) score by 1 (Use this block to change the number by the amount you want)

Direct students to:

- Write pseudocode for a program that will count the number of times Kiki recognizes a color.
- Make sure the pseudocode includes a score for all three colors. (Students may change the sounds that are made when a color is sensed.)
  Hint: Reset the scores to 0 at the beginning of the program.
  Example pseudocode:
  - Repeat 10 times:
  - If the color is blue, add one to Blue Score. Else

- o If the color is green, add one to Green Score. Else
- o If the color is yellow, add one to Yellow Score. Else

Have students write code and test the program. Remind students to use comments to help with debugging.

3. Explain

Discuss with students what happened when they ran the program and how the score was kept.

Ask students:

- What was the variable?
- What was easy and what was challenging about this activity?
- How did creating a variable help you keep score in the program? Did the program use the variable to accurately keep track of the number of times a color was sensed?
- Why was it important to clearly name your variable?

4. Elaborate

Ask students to create a simple game using the color sensor and different colors of elements.

- The program should keep score.
- They do not have to keep Kiki together – they can use only the color sensor and the hub if they prefer.
- Some colors must reduce the score.
- Colors can be given different point values.

If students are struggling, and the other teams can't help, you can give them these hints:

Hints:

- o Reset the scores at when the program starts.
- o Use operators to determine the total score.
- o Create several program stacks to run simultaneously.

Have students share their scores at the end and explain what they did.

5. Evaluate

Teacher Assessment

Evaluate the students' understanding of how variables are used to keep score.
Evaluate the students' understand of how to use loops and conditionals.

Self-Assessment

Have students answer the following in their journals:

- How could using variables could be part of any solution to the culminating challenges.
- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on your time management today.
- Using a scale of 1-3, rate yourself on your materials (parts) management today.

LEGO® Education SPIKE™ Prime

Graphing, Speed, and Distance

Grade 6-8 | **90 minutes** | **Intermediate**

Graphing, Speed, and Distance

Graph and analyze data created data.

Questions to Investigate

How do engineers know how fast a vehicle can go? How can they measure the distance an object will go with one push without a measuring tape?

Materials Needed

- SPIKE Prime Set
- Device with SPIKE App installed
- Student journal

Prepare

- Check to make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.

1. Engage

Students will explore ways to measure distance using a sensor. First, they will make an object roll and try to determine the distance without a tape measure.

Have students take two wheels from their set and connect then with an axle. Ask students to give the wheels a push. Discuss how far they rolled. Without a tape measure or ruler, can they measure the distance? Discuss ways you could determine the distance traveled. (Measure with a tape measure, measure with steps, measure based on tiles, difference in sensor distance from wall when started and stopped)

Ask students to take apart the wheels and axles and put them back into the set.

KEY OBJECTIVES

Students will:

- Use variables to determine speed.
- Graph and analyze data.
- Create data around energy.

STANDARDS

CSTA

2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
2-AP-17 Systematically test and refine programs using a range of test cases.
2-DA-08 Collect data using computational tools and transform the data to make it more useful and reliable.
3A-CS-03 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
3A-AP-16 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.
3A-AP-17 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

VOCABULARY

Rotations
Circumference
Distance

2. Explore

Students create a vehicle and use sensors to determine the distance traveled.

Have students launch the **Aim for It** lesson in the **Training Trackers** unit. Ask students to complete **Step 01** by watching the video and consider the questions. Direct students to complete the Complete **Step 02**. Build the curling rock. How far does the curling rock move during one rotation of the wheel? (the distance equal to the circumference of the wheel)

Next, have students complete **Step 03**. Click Variables on the left side of the programming area. Click each box to place a checkmark in front of each variable so you can see the data. Click the Variables box on the right side of Lesson.

Have students start the program. Press the button attached to the force sensor on the top of the curling rock and release it when you let the curling rock go during the push. When the curling rock comes to a stop, press the left button on the hub.

Ask students to look at the graph. The blue line represents the distance and yellow line represents the color changes from the color sensor – black to white to black and so forth.

3. Explain

Ask students:

- What pattern do you see on the yellow line representing the color sensor sensing color changes? (They get farther apart as the curling rock slows.)
- How can you explain the blue line showing distance? Why does it arc? (The distance traveled over a given period diminishes.)
- Looking at the variables, what is the total distance your curling rock traveled? What is the relationship between distance, rotations, and circumference? (The circumference multiplied by the number of rotations should be equivalent to the distance traveled.

4. Elaborate

Students determine the speed and initial kinetic energy of the vehicle when they give it a push.

Have students complete **Step 04**. Below the programs shown on the programming canvas are some additional programs. Students may or may not have seen them already. Ask students to add these programming blocks to the hat block so the time and speed variables will be used.

Ask students to run the program. Have them push the curling rock like they did in **Step 03.**

Discuss with students what they see on the graph. Now, what information can they gather from the variables - speed and time?

Remind students how speed is calculated. Speed is distance divided by time. The distance of one circumference, which is one white to black to white cycle of the pointer on the wheel, is 17.5 centimeters. The elapsed time it takes to move one circumference (distance) is the clock time when the white color is seen by the color sensor the second time minus the clock time when the white is seen by the color sensor the first time. (For example, the color sensor detects the white color at 1.05 seconds and 3.25 seconds. The elapsed time is 3.25 - 1.05, or 2.2 seconds.)

Elapsed time is used to calculate speed. The distance, 17.5 cm divided by the elapsed time determines the speed in centimeters per second. Students need to be clear about the units for each of the variables.

Ask students to verify the calculations made by the computer. What is the speed? Remind students that the units are cm/s. Typically, speed would be calculated in miles per hour or meters per second, but for the curling rock, centimeters per second is practical.

Ask students if every team get the same speed. Probably the speed varies. Ask students why the speed varies between teams. Ask students to think about the factors that could make the speed vary.

Ask students to think about how they could vary the speed. Have students change the speed so the curling rock moves slower and then change the speed to make it move faster. Have students verify the results by running the program when they push the curling rock.

Challenge students to think about how changing the size of the wheels could change the speed.

5. Evaluate

Teacher Assessment

Evaluate the students' understanding of how the data on a graph can be analyzed. Evaluate the students' understanding of how the circumference of the wheels could change the speed, distance, and the initial kinetic energy of the curling rock.

Self-Assessment

Have students answer the following in their journals:

- How could data and analysis be helpful with the challenge?
- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on your time management today.
- Using a scale of 1-3, rate yourself on your materials (parts) management today.

LEGO® Education SPIKE™ Prime

Mini-Challenge: Distance Game

Grade 6-8 | **90 minutes** | **Intermediate**

Mini-Challenge: Distance Game

Graph and analyze data created data.

Questions to Investigate

How do engineers change the distance a vehicle will go?

Materials Needed

- SPIKE Prime Set
- Device with SPIKE App installed
- Student journal
- 1 target for each team
- Tape
- Tape measures
- Curling Rock model

Prepare

- Check to make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.
- Students should have completed the Graphing, Speed, and Distance lesson prior to working on the Mini-Challenge: Distance Game. The students will need the curling rock model built prior to beginning this lesson.

1. Engage

Students will explore ways to make a vehicle go different distances.

Have students push their curling rocks in a location (like a hallway) where they can go a long distance. Have them try two or three times.

Ignite a discussion. Ask students, what could you do to make the curling rock go even further (e.g., push harder, add weight, make more aerodynamic).

KEY OBJECTIVES

Students will:

- Use data analysis as strategy for winning a game.
- Create and program multiple variables for gathering data.
- Create a program to determine initial kinetic energy.

STANDARDS

CSTA

2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.
2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation and review of programs.
2-DA-08 Collect data using computational tools and transform the data to make it more useful and reliable.
3A-AP-13 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
3-AP-16 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.

VOCABULARY

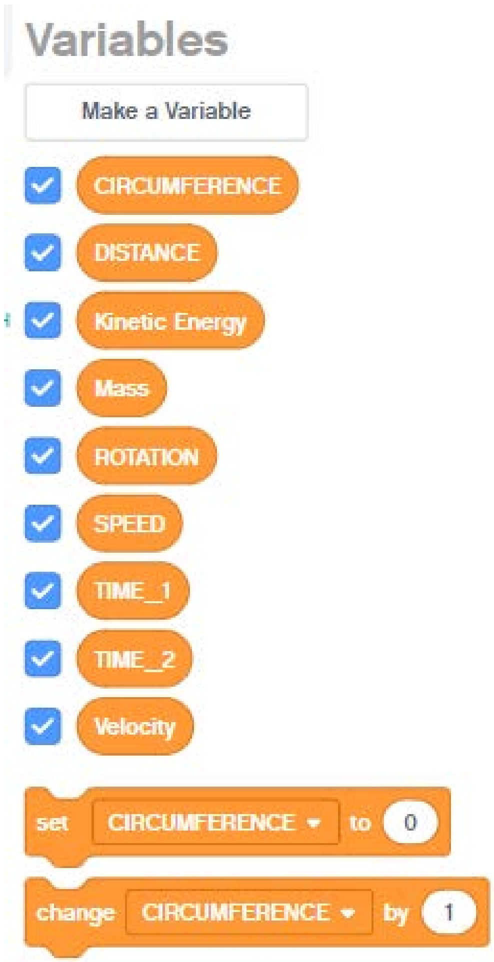
Kinetic energy
Variable
Mass
Velocity

2. Explore

Students will determine the initial kinetic energy of the vehicle when they give it a push.

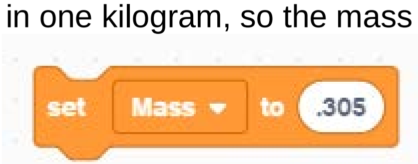
Have students to the **Aim for It** lesson in **Training Trackers** to locate the building instructions and the programming.

Ask students to complete **Step 05**. Create a variable called Kinetic Energy. Create a variable called Mass. Create a variable called Velocity.



Remind students initial kinetic energy is measured as ½ mass times velocity squared.

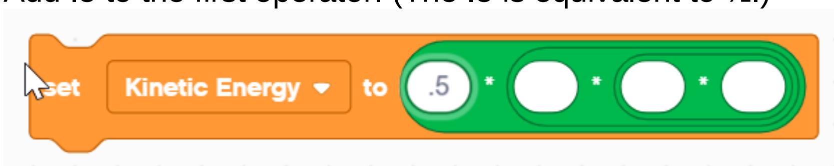
Ask students to add a "set Mass to" block. Change the "0" to ".305". Ask students why they are setting the mass to .305. (Explain to students that the mass of the curling rock is given as 305 grams. However, the formula uses mass in kilograms. There are 1000 grams in one kilogram, so the mass is .305 kilograms.)



Ask students to add a “set Kinetic Energy to” block and three operator blocks that are multipliers.

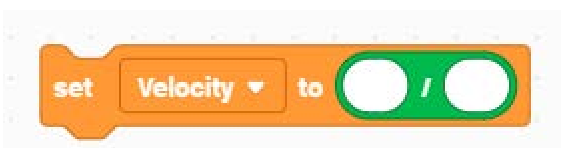
Help students set up the formula for Initial Kinetic Energy: KE = ½ mass times velocity squared. Direct students to:

- Add .5 to the first operator. (The .5 is equivalent to ½.)

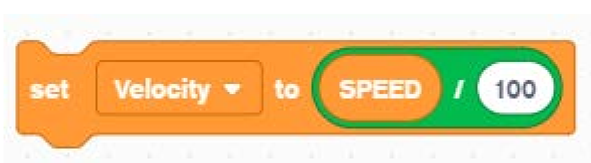


- Add the mass. Use the Mass variable block as an operator.

- The velocity of the curling rock is equivalent to the speed because there is only one direction (vector) of movement. However, they need to convert speed (centimeters per second) to velocity (meters per second). Create a set velocity to block with the operator that is division.

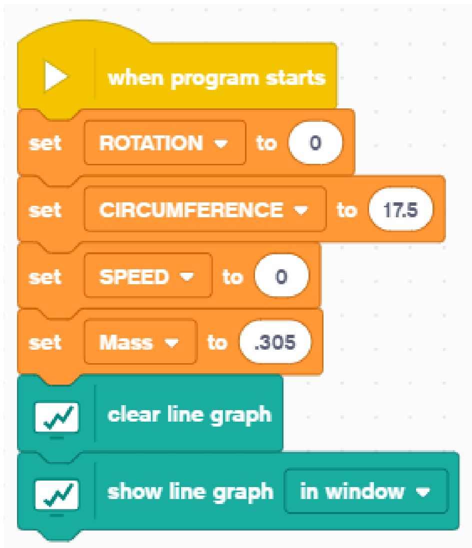


- Students need to convert the speed to the same measurement of velocity. First add the speed variable as the first operator. How many centimeters are in one meter? There 100 centimeters in one meter, so use 100 as the second operator.

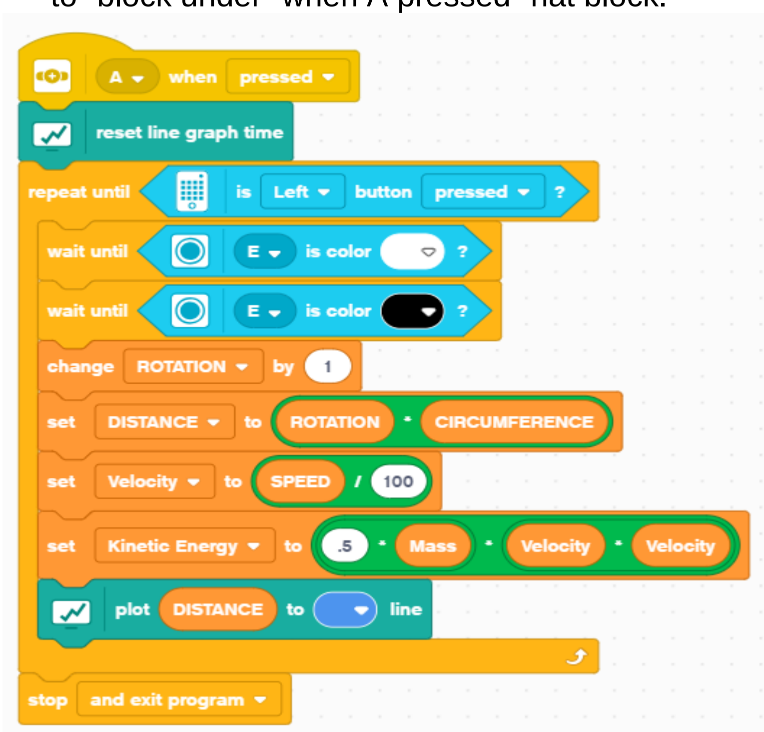


- In order to square the velocity, you must code the block to multiple the velocity by itself.

- Place the “set Mass to” block under the “set speed to” block under the “when program starts” hat block.



- Place the “set Kinetic Energy to” and “set Velocity to” blocks under “set distance to” block under “when A pressed” hat block.



Ask students to verify their programming with another team and then run the program.

3. Explain

Ask students:

- What value did you get for Initial Kinetic Energy? (The units are joules.)
- Think about the formula. How could you change the amount of Initial Kinetic Energy? (add mass)
- Try adding mass to the curling rock and then run the program. If students are able to measure the mass added, remind them to change the value of Mass in the code. Have students explain how the result changes when the mass of the curling rock changes.
- What is the relationship of an increased Initial Kinetic Energy to distance traveled?
- What are two factors that affect distance traveled?

4. Elaborate

Students play a game using the factors that change the distance a vehicle moves on a given surface.

Have students watch the video on **Step 06**.
Set up the playing field. Place one target 2 meters away from the starting line.
Have students follow these instructions:

- Have each partner try to push the curling rock so it stops directly on the target. They can add or subtract weight from the curling rock and adjust the push force.
- Measure the distance from the target for each try. (Measure from the front of the curling rock to the center of the target.) Write the measurements in your journals.
- After each partner has had three trials, move the target to 3 meters away.
- Each partner gets three trials, measuring the distance from the target for each try.

Ask each partner to add together the distances from the target for each trial. Which one was the closest, meaning which total number was the smallest? Did students need to add or subtract mass to get to the farthest target?

Have students move the target 4 meters away. Ask them to repeat the trials and measurement, writing the results in their journals.

5. Evaluate

Teacher Assessment

Evaluate the students' understanding of how data can be used to make changes to a vehicle's distance.

Evaluate the students' understanding of how increasing Initial Kinetic Energy can change the distance a vehicle travels.

Self-Assessment

Have students answer the following in their journals:

- How determining using mathematics could be helpful with any of the culminating activities.
- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on your time management today.
- Using a scale of 1-3, rate yourself on your materials (parts) management today.

LEGO® Education SPIKE™ Prime

Connecting to Career: Finance and Business Management & Administration

Grades 6-8 | **90 minutes** | **Beginner**

Connecting to Careers: Finance and Business Management & Administration

In this lesson series, students will have the opportunity to explore and research careers.

Prepare

Prior to starting the lesson, prepare the following:

- Set aside enough LEGO® bricks for students' models.
- Make sure you have enough devices and access to the Internet for student use during this lesson.
- Make copies of the handouts (if desired) or place into a digital platform for student use.
- Prepare images of different jobs. There should be a variety of images to show different jobs within the 16 career clusters. (Examples: Accountant, Stock Broker, Financial Planner, Auditor, CEO, Sales Manager, Pension Fund Manager, Estate Administrator, Bank Teller.).

1. Engage

Ignite a discussion with students: Present the images of the jobs in the area of Business Management & Administration and Finance to students—as a class, decide how you would categorize the jobs. Ask questions like:

- What jobs belong together?
- What kind of similar skills do these different jobs use?
- What kind of environments are associated with these jobs?
- Do any of these jobs interact or rely on one another? If so, how?

KEY OBJECTIVES

Students will:

- Articulate their personal interests and goals.
- Relate their personal interests and goals into possible career pathways.
- Explore various careers in career pathways.

STANDARDS

Career Ready Practice 10- Plan education and career path aligned to personal goals. (CCTC)

VOCABULARY

Career
Career Cluster
Qualifications
Skills
Education
Knowledge

Ask students to think about what interests them. What kind of job(s) would they like to have in the future?

2. Explore

Students will be assigned two career clusters each time career connections lessons are taught. By the end of the course, students will have explored all 16 career clusters.

There are 16 career clusters:

- Agriculture, Food & Natural Resources
- Architecture & Construction
- Arts, A/V Technology & Communications
- Business Management & Administration
- Education & Training
- Finance
- Government & Public Administration
- Health Science
- Hospitality & Tourism
- Human Services
- Information Technology
- Law, Public Safety, Corrections & Security
- Manufacturing
- Marketing
- Science, Technology, Engineering & Mathematics
- Transportation, Distribution & Logistics

The two career pathways we are studying today are:

- **Business Management & Administration**
- **Finance**

Ask students to brainstorm as many jobs as you can within each career cluster. Have one person write down the jobs as they are named.

Ask students to think about what they studied and learned in the lessons in Unit 6. (drones, variables, data analysis, graphing, energy, velocity, strategy) Ask students how any of these can relate to business management, administration, or finance. (Data analysis, graphing, strategy, variables are used in all three areas. Some business use drones; one key area of business in the energy sector, and so forth.)

Ask students what skills they used in these lessons and how those skills might help someone seeking a career.

In small groups of 4, students complete online research to find out more about each career clusters. Allow time to get information about the jobs that included, especially jobs they had not heard of or had not thought of in their brainstorm.

Students should also research:

- Skills needed
- Forecast of future job openings (and current)
- Education level, certifications, licenses, apprenticeships, etc. are required

In their group, ask students to create a visual representation (build a model) of one of the two career clusters. They will build one or more physical models with LEGO® bricks. Each person can build a model, or the group can create one large model that represents all ideas about one of the career clusters.

Each group will be responsible for a quick one-minute presentation of their LEGO® model(s) to explain how it represents their career cluster.

3. Explain

When the students have finished building, allow each group to present their model.
Ask students:

- To tell about your LEGO® build.
- What kind of interests would these career clusters have?
- What are some of examples of jobs in these career clusters?
- Can you explain the difference between a job and career cluster?
- How are these career clusters similar to one another?

4. Elaborate

Ask students if there were any jobs in today's career cluster that were interesting to them.

Ask students how the lessons relate to the career clusters of Business Management & Administration and Finance. What types of businesses could use drones? Do you think drones would be more expensive than other methods of taking aerial photos, delivering small packages or mail, or surveillance? How would someone in these careers get and analyze information in order to help make a decision?

Have students think about the game pushing a curling rock. What types of businesses would be interested in the results and the analysis of force, distance, and mass? (vehicles such as cars, trucks, trains, planes) Why? (fuel efficiency, cost savings)

5. Evaluate

Evaluate the students' skills development by observing if they:

- Articulate their personal interests and goals.

- Relate their personal interests and goals into possible career pathways.
- Explore various careers in career pathways.

LEGO® Education SPIKE™ Prime

Intro to Arrays (Lists) & Compound Conditionals

Grade 6-8 | **90 minutes** | **Intermediate**

Intro to Arrays (Lists) & Compound Conditionals

Create and utilize arrays (lists).

Questions to Investigate

- Why would an engineer use the Boolean operators "and" and "or" when using sensors?
- How can an array be used to help understand data that has been gathered?

Materials Needed

- SPIKE Prime sets
- Devices with SPIKE App installed
- Student journals
- Sticky notes
- Small paper bag

Prepare

- Make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.

1. Engage

Explain to students that an array is a list of items. It is like a single column with many rows. Each row contains an item. The group of items is called an array. In Scratch-based programming, an array is called a List.

For example, ask students to think about playing a video game and in it you have different homes you move to or from. The code in the game may choose a house for you at random from the list. Perhaps, the code in the game chooses an order of the houses you will live in – maybe the more points you earn in the game, the "better or bigger" the house you can get.

KEY OBJECTIVES

Students will:

- Create and utilize arrays (lists).
- Code with compound conditionals using lists (arrays).
- Read pseudocode and compare with a program.

STANDARDS

CSTA

2-CS-01 Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices

2-CS-02 Design projects that combine hardware and software components to collect and exchange data.

2-DA-09 Refine computational models based on the data they have generated.

2- AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms.

2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.

2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP 16 Incorporate existing code, media, and libraries into original programs, and give attribution

2-AP-18 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.

VOCABULARY

Array

List

Have students share examples of video games that are similar to your example. Encourage the students to discuss ideas to further understand the example.

Create an Array

Have teams take all the 2x4 bricks from the set. Each team needs to have 5 small pieces of paper or sticky notes with one number from 1-5 written on each.

Have each team place the 5 bricks into a small paper bag. Partner 1 will hold the bag of bricks in way that Partner 2 cannot see the bricks. Partner 2 should choose one brick at random. Place that brick on the table. Explain that this brick is located in the first position on the list.

Tell partner 2 to choose another brick at random. Place that brick on the table after the first brick. It is now located in the second position on the list.

Ask students to continue the pattern until they have all five bricks on the table. Explain to each team that they have created an array of items that they could call color. Ask the students to write their array "color" in their journals. List the 5 brick colors in order. Number the colors in order, from 1-5.

Ask students to create another stack. Tell partner 2 to place the papers 1-5 in the paper bag and partner 1 to put all the 2x4 bricks into a pile that both partners can access. Have teams create a stack of 5 bricks by doing the following:

- Repeat until the stack contains 5 bricks.
    - Partner 2 holds the bag containing the papers numbered 1-5 so Partner 1 cannot see them.
    - Partner 1 chooses a piece of paper at random.
    - Partner 2 hands Partner 1 the correct color.
    - Partner 1 verifies the color and adds it to the stack.
    - Partner 2 places the numbered paper back into the paper bag.

Discuss the experience as a group. Each team created a stack of bricks based on the random selection of numbers, each corresponding to a color. Did any team have repeated colors? Why could it happen? (Because we are replacing the number.)

Ask the students to write pseudocode for choosing a random number from an array to make a stack of 4 bricks from a list of 10 colors. They do **not** need to pull physical bricks and they can choose any ten colors.

When students have completed writing pseudocode, have each team trade pseudocode with two other teams. Have them read the pseudocode and give feedback.

Ask students what they expect the program to do. Do they expect four different colors to be chosen every time? (No, because we are not eliminating any colors when one is chosen.) How many of the same color would be possible? (All four could be the same color.)

Have students put away the bricks.

2. Explore

Students will create an array using the color sensor and several colors of bricks. Then they determine the number used by the software to identify each color.

Ask students to complete the following steps:

- Open the **SPIKE App** and choose **New Project +, Word Blocks**, **Create.**
- Connect the hub.
- Scroll down on the block palette until they see Variables.
- Click **Make a List**. Then, name the list "Colors" and click ok.

Show students that they will see a new list (this is the array) on the upper right side of the screen. At this point there are no values in the list.

Students will want to program the color sensor to put values into the list. Discuss how they could write code to do that.

Look at the following program with the students. Read the program. Ask students to explain what the program is asking to be done.

When program starts Repeat ten times Add the color identified by the color sensor in port A to the list colors Wait three seconds

Note: When the program starts, the color sensor in Port A will read a color and add it to the list (array) named "Colors". It will wait three seconds and repeat reading and adding a color – 9 more times. The add block is new for students. It is located under Variables and is shown as add "thing" to Colors.

Ask students to line up the five colors of 2x4 of bricks from the set in any order. (Bricks are magenta, green, yellow, blue, red.) Ask students to write code for adding 5 colors to the list (array).

Tell students to attach the color sensor to the hub and then run the program.

Discuss what values are in the array "Colors" as a group.

Explain to students that the array is filled with numbers that represent the colors.
Ask students:

- How can we determine the number that corresponds to each color?

Explain to students there are several ways, but notice that the icon for the color sensor is shown next to the hub icon at the top of the programming area. The numbers below the color sensor shows live data. When you put the magenta brick in front of the color sensor, you see it change to 1 and the circle becomes magenta.

Note: Magenta 1, Blue 3, Green 5, Yellow 7, Red 9

Allow students time to investigate the program. Have students determine the color represented by each number.

3. Explain

Discuss with students how the program works and demonstrates an array.
Ask students to explain:

- What an array (list) is
- How the program works
- What values are in the list (array) named colors
- How they determined what numerical value is assigned to each color

4. Elaborate

Students will program the hub to identify a color.

Using the lights on the hub, ask students to identify the location of **red** in the list called color. Tell students they will write a program that will identify through lights on the hub where **red** is located within the list (array) called color (9 is the color code within the SPIKE App for red).

Provide students with the pseudocode for the program:

- When program starts
- Wait 1 second
- If item 1 of color list = 9
  - Then write the word first using the hub lights
  - Else
  - If item 2 of color list = 9
    - Then write the word second using the hub lights
    - Else
    - If item 3 of color list = 9

- Then write the word third using the hub lights
- Else
- If item 4 of color list = 9
  - Then write the word fourth using the hub lights
  - Else
  - If item 5 of color list = 9
    - Then write the word fifth using the hub lights
    - Else
    - Play beep 60 for .2 seconds
- Play sound Dun Dun Dunnn until done.

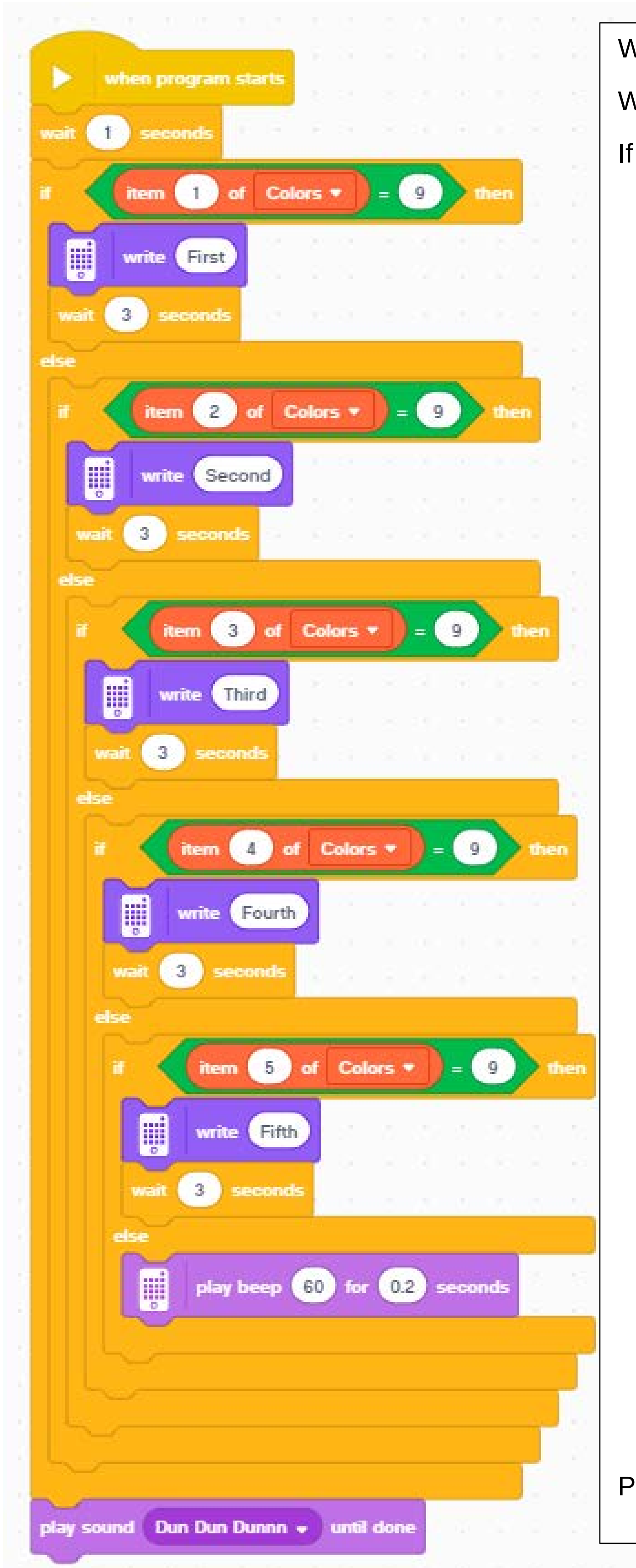
Have students indicate the blocks needed to create the code from the programming palette. Have students write the program. Have teams help each other if they get stuck.

Ask students to test their programs.

Ask students to explain what happened (They should have seen the position for a red brick located in the in the array).

Have students move the number 9 to a different location. Remind students they only need one red brick at this point. Ask students to test their program.

Note: If students need additional support, you can show them this program.



When program starts

Wait 1 second

If item 1 of color list = 9

Then write the word first using the hub lights

Else

If item 2 of color list = 9

Then write the word second using the hub lights

Else

If item 3 of color list = 9

Then write the word third using the hub lights

Else

If item 4 of color list = 9

Then write the word fourth using the hub lights

Else

If item 5 of color list = 9

Then write the word fifth using the hub lights

Else

Play beep 60 for .2 seconds

Play sound Dun Dun Dunnn until done.

After students have completed coding, ask students where the compound conditional statements are located in their programs (When an if statement is embedded within an if statement, a compound conditional exists).

> **Optional:** Ask students who finish quickly to add their voices in recordings to say First, Second, Third, and so forth.

5. Evaluate

Teacher Assessment

Evaluate students' understanding of an array and how it can be used in programming.
Evaluate students' understanding of a compound conditional.
Ask students:

- What challenges they encountered in trying to create a complex program with compound conditionals and arrays (lists).
- If they asked other teams for help. If so, what did they learn?

Self-Assessment

Have students answer the following in their journals:

- What did you learn about arrays or compound conditionals that could be helpful with the culminating activities?
- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on your time management today.
- Using a scale of 1-3, rate yourself on your materials (parts) management today.

LEGO® Education SPIKE™ Prime

Comparing Arrays (Lists)

Grade 6-8 | **135 minutes** | **Intermediate**

Comparing Arrays (Lists)

Compare arrays (lists) and use multiple operators within a conditional statement.

Questions to Investigate

- How can we use and/or operators with a list? Why would we use and/or operators with a list?
- How can an array be used to help understand data that has been gathered?

Materials needed

- SPIKE Prime sets
- Devices with SPIKE App installed
- Student journals
- Sticky notes

Prepare

- Make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.

1. Engage

Have each team take out 10 bricks – 2 of each color of the 2x4 bricks: red, green, yellow, blue, and magenta. Each partner should have one of each color brick.

Ask each partner puts his/her bricks in a stack (array or list) that the other partner can't see. Discuss how they could compare the stacks.

Ask students how they could write a program that would compare the two stacks and determine if the stacks are the same. Have students work together to write pseudocode to compare the two stacks and determine if they are the same. Ask them to follow the pseudocode using each of their stacks.

Ask students:

- How did it work?

KEY OBJECTIVES

Students will:

- Create and compare arrays (lists).
- Use conditionals with multiple operators.
- Use Boolean operators while coding.

CSTA STANDARDS

2-CS-01 Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices
2-CS-02 Design projects that combine hardware and software components to collect and exchange data.
2-DA-09 Refine computational models based on the data they have generated.
2- AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms.
2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
2-AP 16 Incorporate existing code, media, and libraries into original programs, and give attribution
2-AP-18 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.

VOCABULARY

Conditional
Array

- Did your pseudocode determine correctly if your brick stacks were the same?
- If your code doesn't work, debug your pseudocode.

Have students place their brick stacks in the same order and follow the pseudocode again.

Ask students:

- Did your pseudocode determine correctly that your brick stacks were the same? (If your code doesn't work, debug your pseudocode.)

Tell students to have another team check their pseudocode by trying to follow the program using stacks that are the same and stacks that are different.

Note: If students are struggling, discuss the following pseudocode:

- Read the color of the first brick in stack 1.
- Read the color of the first brick in stack 2.
- The color of the first brick in stack 1 = the color of the first brick in stack 2.
    - If true, read the color of the second brick on stack 1.
    - Read the color of the second brick on stack 2.
    - The color of the second brick in stack 1 = the color of the second brick in stack 2.
        - If true, read the color of the third brick on stack 1.
        - Read the color of the third brick on stack 1.
        - Read the color of the third brick on stack 2.
        - The color of the third brick in stack 1 = the color of the third brick in stack 2.
            - If true, play win sound.
            - If false, play bonk sound.
    - If false, play bonk sound.

Ask students to put the bricks away.

2. Explore

Students will play a game using the color sensor and creating arrays (lists). They will compare the arrays (lists) to see if they are the same.

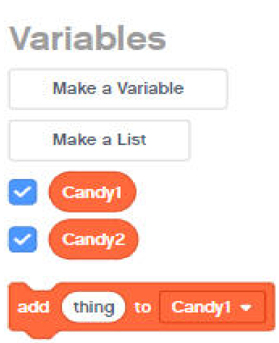
Player 1

Have students complete the following:

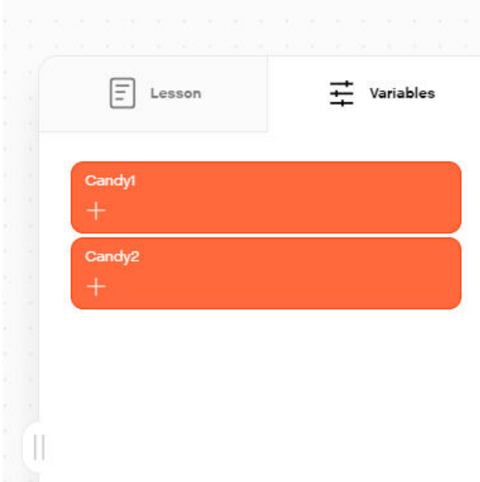
- Find the **Brain Game lesson** in the **Life Hacks unit**.
- Build the Game Master in **Step 02**.
- Read **Step 03**. Students may need to zoom out to see both programs. Each program has comments. They should read through the code and determine what they think will happen with each block. Discuss as a group.

- The two programs for **Step 03** are shown here:

- Place a check mark in the boxes for the reporter blocks Candy1 and Candy2, if they are not already checked.



- Next to the word Lesson on the right side of the screen find Variables.



If the section Variables has not automatically opened, have students click Variables and they will see the empty arrays named Candy1 and Candy2. Ask students what they think will be placed in each of these arrays. Discuss together as a group.

Have students complete **Step 03** and play the program.
Ask students:

- What did it tell you about Candy1?
- Do you know where the red brick is located?
- Do you know which is the top of the candy stick (first brick) and which is the bottom (last brick)?

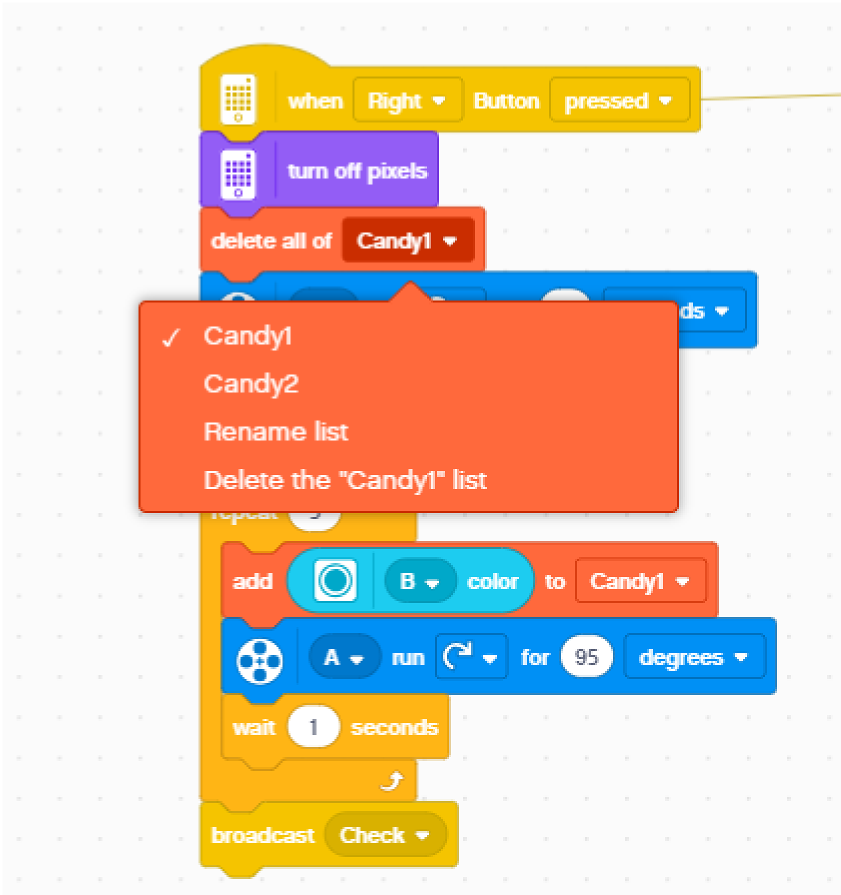
Candy Stick Direction
Tell students they will determine which end of the candy stick is the bottom.

Have students place their red brick at one end and observe which end of the candy stick is placed first into the Game Master Head. Have students write in their journals if the red brick entered the Game Master's head first or last.

Ask students to play the program. Did they hear dance magic? If they heard "oops", have students place the candy stick in the opposite direction and try again.

- **Note**: They should hear the dance magic sound and know which end is considered the bottom by the Game Master.

Player 2
Have students complete **Step 04**. Tell students to duplicate the first program. They need to change the duplicated program so the right button is pressed to start the stack and change the array name to Candy2.



Change "left" to "right" on the hat.

Use the pull-down menu to change delete all of "Candy1" to "Candy2."

Use the pull-down menu to change add color from "Candy1" to "Candy2"

Ask students:

- What will you need to change for this program?
- Will you need to change the broadcast message block and message block?

Have students make all changes and then run their programs using the second candy stick. They should be able to identify the location of the red brick.

3. Explain

Discuss how the program works and what they learned about arrays.
Ask students:

- How does the program work?
- What are the contents of the two arrays?
- How can you prove which is the top and which is the bottom of the candy stick?

4. Elaborate

Students will play a game to compare colors on the sticks.

Same Position

Have students complete **Step 05**. Ask students if they can program the Game Master to play the "Win" sound if the red bricks on both candy sticks are in the same position. Recommend that students write pseudocode before starting to code and use comments to explain parts of the programming.

Code Breaking Game
Have students read **Step 06** and program the Game Master to play a sound if all the colors on both candy sticks are the same. Students need to write pseudocode before starting to write code. Allow teams to help each other if needed.

> **Note:** Consider having a short discussion with the class so students can share ideas prior to having them start writing pseudocode or programming for **Step 06**.

Ask students to explain how they created their program after they demonstrate success.

Have student take apart the models and put the elements into the trays.

5. Evaluate

Teacher Assessment

- Evaluate the students' understanding of "and" and "or" in Boolean operators..
- Evaluate the students' understanding of an array and how it can be used in programming.
- Ask students what challenges they encountered in trying to create a complex program with multiple steps and conditionals.

Self-Assessment

Have students answer the following in their journals:

- Did you ask other teams for help? If so, what did you learn?
- What characteristics of a good teammate did you display today?
- Rate yourself on a scale of 1-3 on your time management today.
- Rate yourself on a scale of 1-3 on your materials (parts) management today.

LEGO® Education SPIKE™ Prime

Conditionals and Simplifying Code

Grade 6-8 | 90 minutes | Intermediate

Conditionals and Simplifying Code

Practice simplifying code by making a series of coding blocks into a subcomponent utilizing a My Block.

Questions to Investigate

- What are ways to simplify code?
- What are subcomponents and why are they helpful?

Materials Needed

- SPIKE Prime sets
- Devices with SPIKE App installed
- Student journals
- Colored tape or strips of paper for red, blue, yellow and black that trigger the color sensor (Black electrical tape and blue painters tape work well)

Prepare

- Check to make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.

1. Engage

Simplifying Code: My Block

Ask students to take out one blue frame element from the set and one minifigure. The frame represents a piece of code. It has several subcomponents.

Tell students to start at a corner then walk their fingers along the straight edge to the next corner. Next, students should turn their hand so that their fingers will walk along the straight edge again. Ask students if they can see the pattern of straight, turn, straight, turn. Explain that the frame represents walking in a rectangle.

KEY OBJECTIVES

Students will:

- Use conditionals to make a robot follow a line.
- Debug programs using sub-components.
- Utilize My Blocks when creating subprograms within a comprehensive program.

STANDARDS

CSTA

2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms.
2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.
2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution
2-AP-18 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.
3A-AP-16 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.
3A-AP-17 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

VOCABULARY

My Block
Subcomponent

Walk the Frame
Have students work in teams. Tell students to choose one partner to be the robot and the other partner to be the computer code. The robot will be controlled by the computer. To simplify the code, the computer uses "frame." Ask the computer partner to hold up the frame (which is the code) and robot partner to physically walk in the room so his/her steps create a small rectangle (i.e., follows the code "frame" to create the rectangle).

Write the Code
Ask students to write the pseudocode for the subcomponent called "frame."
One example might be:

- Walk forward
- Turn right
- Walk forward
- Turn right
- Walk forward
- Turn right
- Walk forward
- Turn right

A simplified example might be:

- Repeat four times
    - Walk forward
    - Turn right

However, the simplest way to write the code is to write the word "frame" – which has meaning to the computer and robot - walk in a rectangle.

Explain to students that in Scratch-based programming a **My Block**, which allows one block to take the place of many lines of code, can be used. A **My Block** can be a subcomponent of an entire program.

Add Subcomponents
Have students place two elements beside the frame. (They are not attached.)

Ask each team to determine what physical movement the two elements represent and what the program would be for that movement. One example might be the biscuit – which means move your hand to your mouth and take a fake bite and chew.

> **Note:** A biscuit is either a magenta or black piece with 5 holes in the shape of a plus sign with 2 holes on two sides.

Another example might be a 1x4 beam which would mean jump 4 times.

Ask students to write the pseudocode for the new elements. Ask both partners to follow the program written by all three elements. Then, students should change the order and follow the new code.

Have teams share their code with at least one other team.

Add Sensors

Have students use a new element to represent a distance sensor. Discuss what the action will be when the distance sensor senses an object 2 feet away (Example: stop, clap your hands, back up slowly).

Ask students how they could create a **My Block** that would help with programming when using a sensor. Have students write pseudocode in their journals for a My Block that uses a distance sensor to cause an action when the sensor senses an object.

Allow students to share their pseudocode and discuss.

Tell students to place all the elements into the correct locations in their sets.

2. Explore

Students will learn how to stop at a line and follow a line. Then, students will use My Block (Elaborate) to simplify the code.

Stop at a Line

Tell students to find the **Training Camp 3** in the **Competition Unit** and build the Driving Base and add the Color Sensor Module.

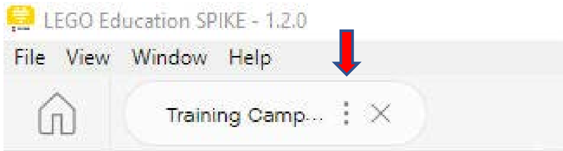
Tell students to complete **Step 03** and stop. Explain to them that this program tells the robot to stop at a black line. Allow students time to try this program.

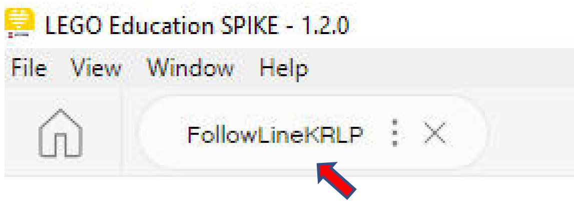
Ask students to change their programs so the robot will stop at a black **or** a red line. Make sure students have a blue or yellow line in front of the black or red line so they know the robot is not triggered by a different color. The robot should not stop until the correct color is sensed. Remind students to debug their programs if the robot stops at a different color line.

> **Hint:** Students have live data shown on screen from your sensors. Roll the robot over the colored lines and watch the sensor readings. Does the color sensor sense the colors? The sensor is made to register official LEGO colors. Therefore, not any red or blue will be sensed properly.

Follow a Line

Ask students to complete **Step 04**, which will tell the robot to follow a black line. Have students rename the program by clicking the three vertical dots next to the program name at the top of the screen by deleting the current name and renaming the program “FollowLine” followed by both partners’ initials.





Have students run the program. Troubleshoot as needed. Students need to observe how the robot behaves.

3. Explain

Discuss the program and how the sensor works with students.
Ask students:

- What happened when they ran the program?
- Why would a color sensor would skip over lines until it senses the correct color?
- How does a robot follow a colored line?

4. Elaborate

Students will use **My Block** to simplify the program for following a line.

Discuss with students how you could use a **My Block**. Ask students what are ideas for names and functionality of a **My Block**.

- For example, 90 degree right turn could be 90RTurn
- For example, Stop at a red line could be StopRed
- For example, Follow a black line could be FollowBlack
- Under each of these you could write the pseudocode words for what the program will do. (For example, StopRed - The robot will stop when is senses red.)

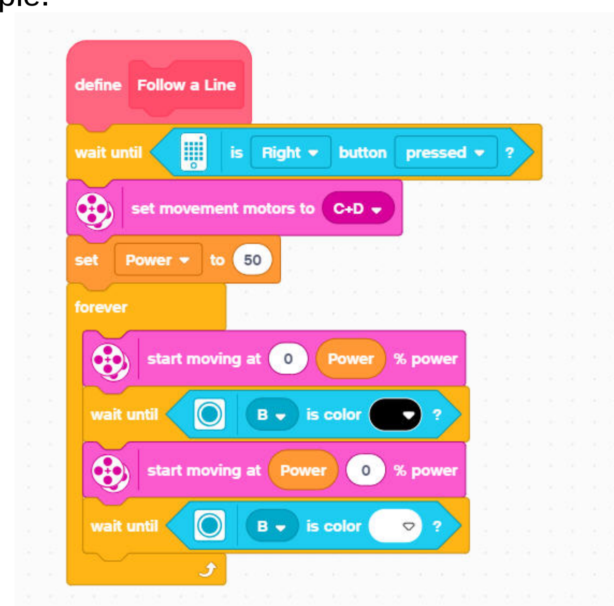
Create a My Block for Following a Line

Student create a **My Block** and program their robots to follow a line, play a sound, and show a light pattern on the hub.

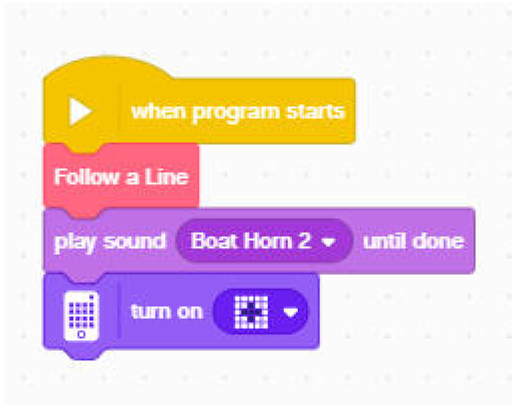
A red hat block called define Follow a Line now appears at the top of the screen under the hub and sensors. Move it down so you can clearly see it.

Ask students to place the code for following a line under the red hat block defining Follow a Line.

Remind students that you cannot have two hat blocks. Discuss what should replace the orange hat block when right button pressed. If students struggle, show them this example.



Have students choose the hat block “when program starts” and place the **My Block** named Follow a Line, found in the **My Blocks** palette, under the hat block. Students should add a “play sound until done” block and a block to show a pattern on the hub. If students struggle, show them this example.



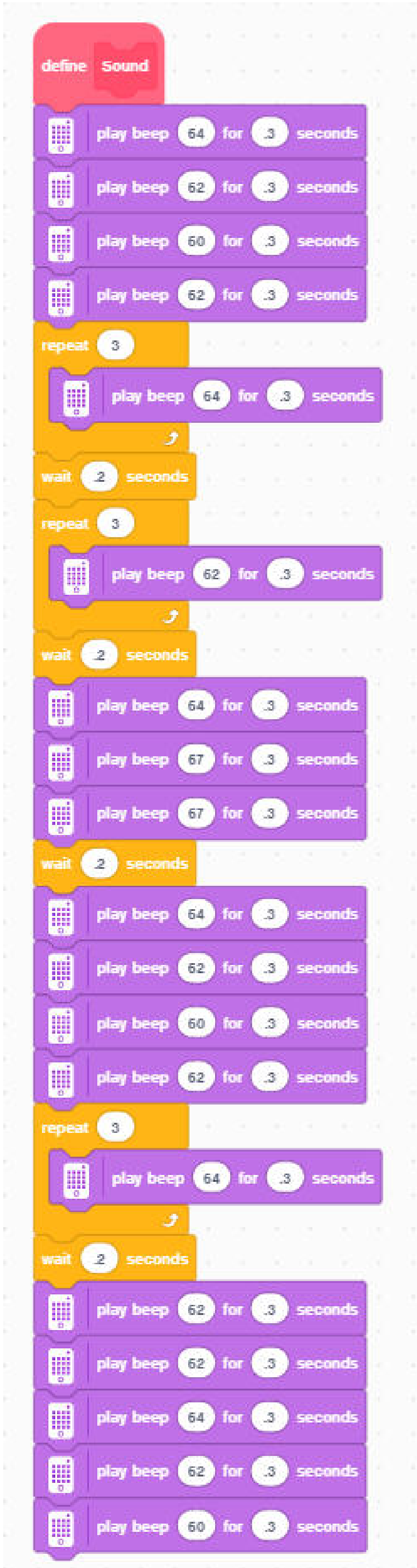
Ask students to test their programs. Remind students to debug and modify the line following instructions as needed. Remind students that they can ask other team members for assistance.

Allow students to iterate debugging the program until their program works well.

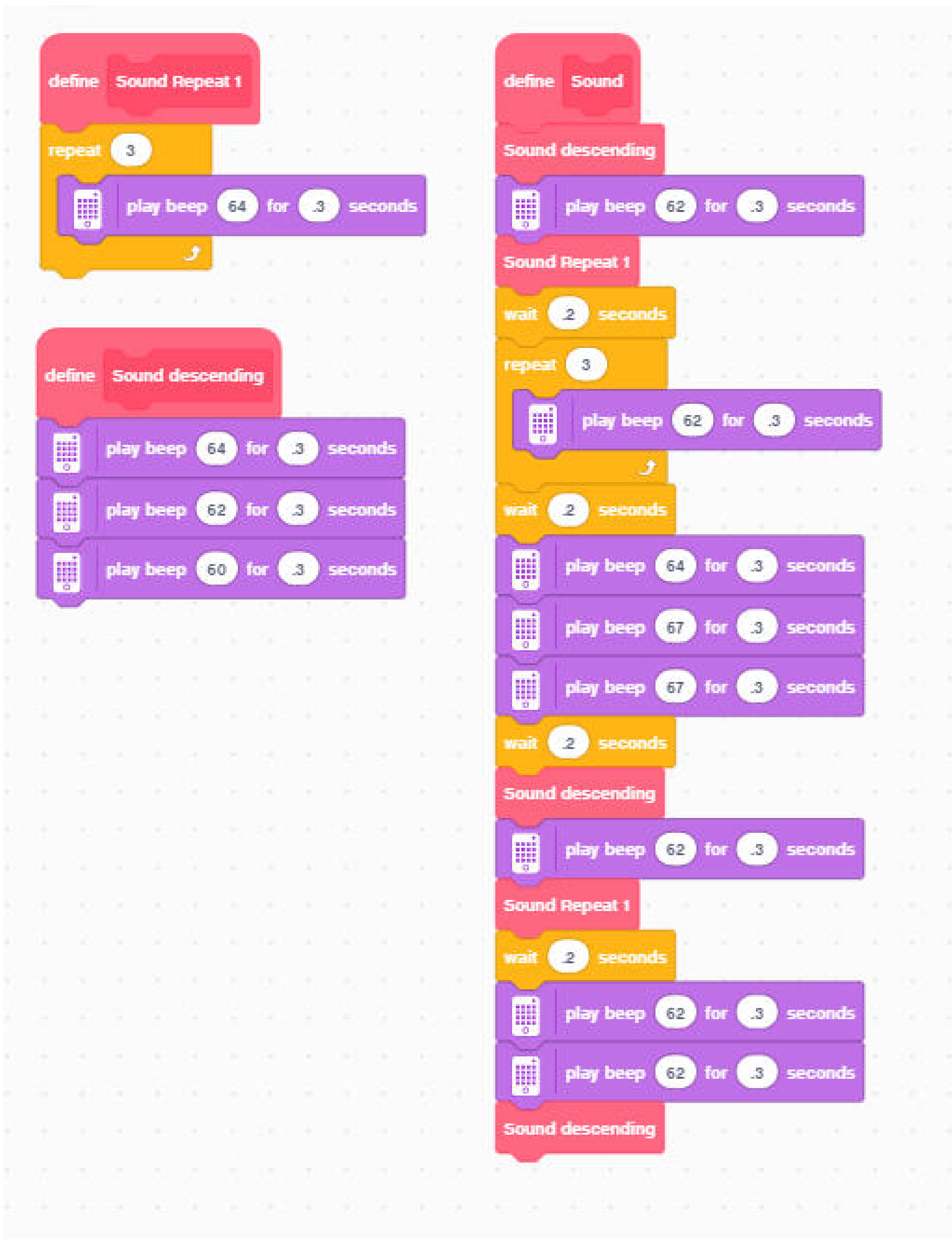
Create a My Block: Sound

Students create a **My Block** for a series of sounds.

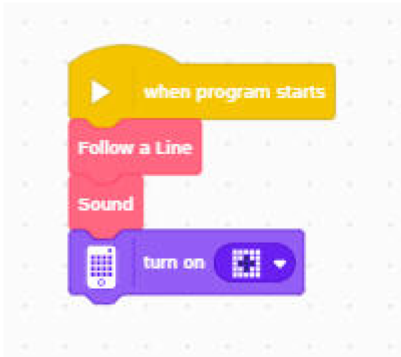
Ask students to create another **My Block** and name it “Sound.” Explain that this **My Block** should play a series of at least 10 sounds or notes. Ask students to write in their journals what they want to play. If students struggle, you can share the following example:


define Sound
play beep 64 for .3 seconds
play beep 62 for .3 seconds
play beep 60 for .3 seconds
play beep 62 for .3 seconds
repeat 3
play beep 64 for .3 seconds
wait .2 seconds
repeat 3
play beep 62 for .3 seconds
wait .2 seconds
play beep 64 for .3 seconds
play beep 67 for .3 seconds
play beep 67 for .3 seconds
wait .2 seconds
play beep 64 for .3 seconds
play beep 62 for .3 seconds
play beep 60 for .3 seconds
play beep 62 for .3 seconds
repeat 3
play beep 64 for .3 seconds
wait .2 seconds
play beep 62 for .3 seconds
play beep 62 for .3 seconds
play beep 64 for .3 seconds
play beep 62 for .3 seconds
play beep 60 for .3 seconds

Discuss with students that if some sounds repeat more than once (like a chorus) another **My Block** (ex. Sound Repeat 1) can be created that can be use within the Definition of Sound. Show students the following example:



Explain to students that they need to replace the “play sound until” block with the **My Block** called Sound. Point out to students how few blocks are needed to run a more complex program that contains many commands inside the **My Blocks**.



Ask students to debug the program until it works well. They may wish to add wait time if the song needs a pause between sounds to keep the beat.

> **Note:** To remove a Definition of a **My Block** from the programming area, you must remove **all** uses of the **My Block** first, or the program will freeze. To correct, restart the program.

Follow a Line and Play Sounds

Have students program the robot to follow a line for 10 seconds and play sounds for the entire time. Ask students to stop everything at the end of 10 seconds.

> **Note:** A **My Block** is only available to be used in the program in which it is created. You cannot copy and paste between programs.

Ask students **not** to take apart the robots because they will be used again.

5. Evaluate

Teacher Assessment

Evaluate the students’ understanding of how to simplify a program.

Ask students what a subcomponent is and how subcomponents are used.

Evaluate the students’ understanding of how **My Blocks** can make a program more efficient by having subcomponents that are used in writing a complete program.

Self-Assessment

Have students answer the following in their journals:

- How could **My Blocks** be useful when programming for the end of unit challenge?
- Reflecting on what you learned from others, what tips and ideas did you use? Did it spur other ideas of your own?
- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on your time management today.
- Using a scale of 1-3, rate yourself on your materials (parts) management today.

LEGO® Education SPIKE™ Prime

Conditionals & Boolean Expressions

Grade 6-8 | **90-135 minutes** | **Intermediate**

Conditionals & Boolean Expressions

Practice writing code using Boolean expressions with multiple conditions and subcomponents.

Questions to Investigate

- Why would engineers and computer programmers use operators or Boolean expressions?
- How can operators control the response of a machine?

Materials Needed

- SPIKE Prime sets
- Devices with SPIKE App installed
- Student journals
- Colored tape or strips of paper for red, blue, yellow and black that trigger the color sensor. Black gaffer tape and blue painters tape work well.

Prepare

- Make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.
- Students should have completed the **Conditionals and Simplifying Code** lesson before beginning this lesson. They will be using (remixing) the code that was created in that lesson.

1. Engage

Have students discuss what they think of when they hear Boolean expression – it may be something they have used in previous units. They should be thinking of words like “or, “and”, “not”.

KEY OBJECTIVES

Students will:

- Use Boolean expressions and My Blocks when writing code.
- Evaluate and write Boolean expressions with multiple conditions
- Debug programs using subcomponents.

STANDARDS

CSTA

2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms.
2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.
2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution
2-AP-17 Systematically test and refine programs using a range of test cases.
2-AP-19 Document programs in order to make them easier to follow, test, and debug.
3A-AP-17 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

VOCABULARY

Boolean expression
My Block
Subcomponent
Evaluate

Tell students an expression is something you can evaluate and determine a value. In math you would evaluate to get a number. In math or in computer science, expressions can contain variables. Expressions can also evaluate to true or false.

Evaluating a Boolean Expression with One Condition

Explain to students that when you evaluate a condition as true or as false, a specific action can occur. For example, the color sensor senses red; if true, turn the motor off. If false, keep the motor on. Notice the expression is not a question, it is a statement.

Tell students they are going to imitate evaluating conditions. Have all students sit down. If students have red on any of their clothing, stand up, else keep sitting.

Ask students to rephrase the example of having red on clothing as a Boolean expression. Example: My clothing has red on it. (Which leads you to say, “True or False.”)

Have students create another example of a Boolean expression with one condition and have the class act out true and false.

Evaluating a Boolean Expression with Two Conditions

Have all students sit. If students have red on any of their clothing **or** they have brown eyes, stand up, else keep sitting.
Ask students to rephrase this example as a Boolean expression. Example: My clothing has red on it, **or** I have brown eyes (Which leads each person to say, “True or False”). Explain to students that you must meet only one of the conditions, but you could evaluate both conditions as true.

Tell students a Boolean expression evaluates to true or false. The criteria is met or it is not.

Have all students sit. Say “My hair is black, **and** I have blue on my clothes. If both conditions evaluate to true, then stand up, else sit down.” Remind students in this example **both** conditions **must be met** for the condition to be true.

Ask students:

- What is the variable in the first condition, “My hair is black”? (The color of hair is the variable. Depending on who is answering the question, an evaluation could lead to true or false.)
- What is the variable in the second condition? (The color on clothes. Therefore, you can get different statements of true or false depending on who is answering the question.)

Remind students that **both** conditions **must be true** in order to evaluate the condition as true.

Have all students sit. Ask the students to evaluate the following Boolean expression: I do **not** have red hair **and** I am **not** near a wall. If true stand up, else sit down.

Remind students that they must evaluate both conditions as true to stand up. Repeat the Boolean expression a couple of times – this is the first condition that has contained "not." If students struggle with this concept, go slowly through each condition.

Adding a Conditional to Represent Using a Sensor

Tell students they are going to look at sensors and data to evaluate a condition as true or false. For example, they might use the distance sensor to measure the distance to the wall as less than 15 cm. If you evaluate as true, decrease power to 10%. If you evaluate the condition as false, continue at 50% power.

Have students write a conditional statement using the distance sensor that can be evaluated as true or false.

Boolean Expressions and Sensors

Based on their exploration, each team will write three examples in their journals of how they could use Boolean expressions using sensors. Have their partners check their examples. Remind students that expressions are statements not questions.

Ask students to write in their journals a conditional statement:

- Using the color sensor that can be evaluated as true or false.
- Using the force sensor that can be evaluated as true or false.
- Using the gyro sensor that can be evaluated as true or false.

Have students share their ideas with the class. You, as the teacher, may need to help students correctly word their examples.

2. Explore

Students will explore Boolean expressions and how to use the Boolean operators "and", "or" and "not" to combine Boolean expression to form new Boolean expressions. They will use the Driving base with the ultrasonic sensor attached.

Ask students to open the SPIKE App. They will need a driving base with the ultrasonic sensor attached. If they do not have a driving base, they should find the building instruction in **Training Camp 3: Reacting to Lines** in the **Competition Unit** and build the driving base with the Color Sensor Module.

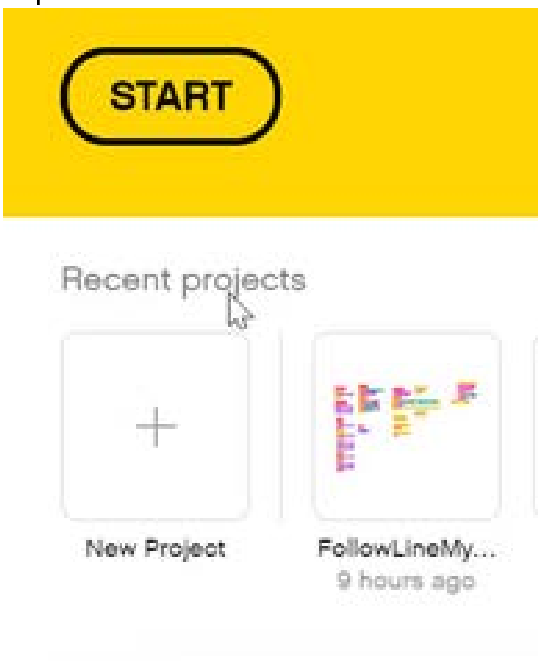
Ask students to add the distance sensor to the top of the hub, so the "eyes" are facing up. They can add distance sensor by using only two black connector pegs. They should connect it to Port A.

Have students click the House to go to the HOME page.

Follow a Line with Boolean Operators

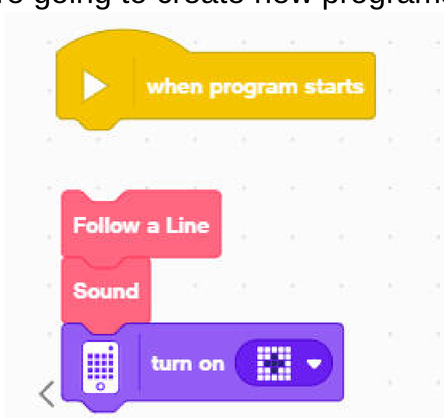
Have students look at Recent Projects. They should find the program "FollowLineMyBlock" followed by your initials that they coded previously. Ask students to open the project. Do **not** delete any programming.

An example of the screen is shown below:



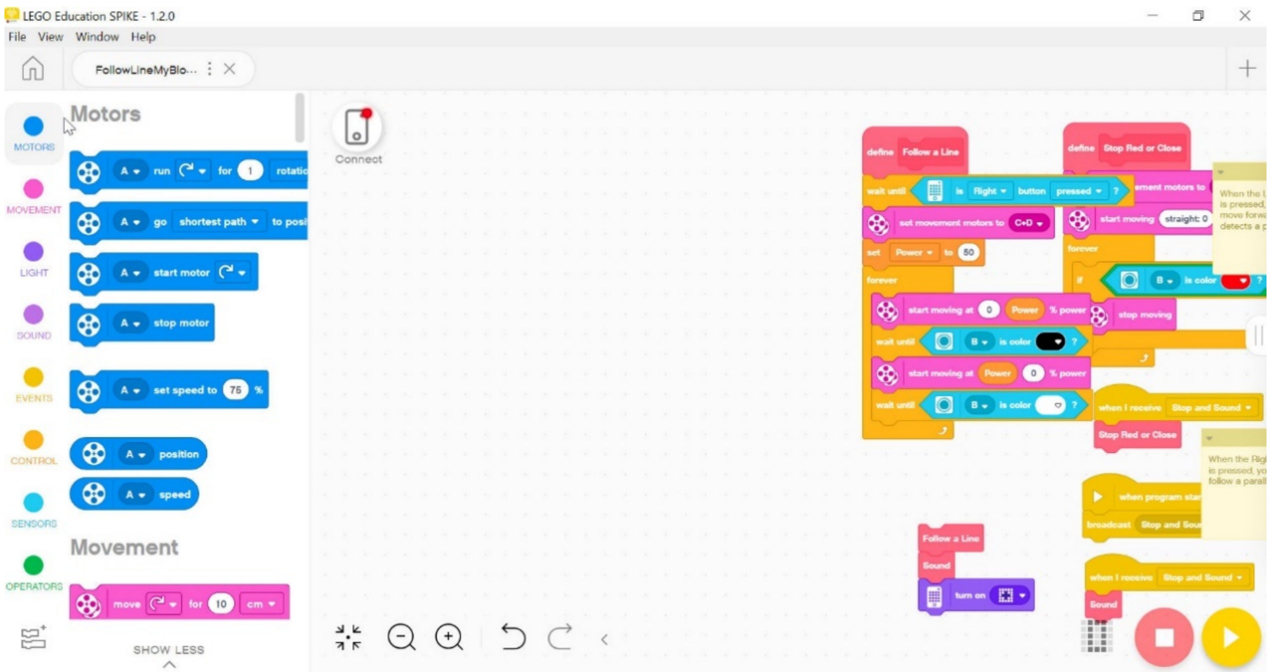
When the project opens, students will see the programs they created previously. Remind students **not** to delete any programs.

Have students find the code that start with a hat block "when program starts". Detach the hat block from those programs. They do not want those programs to run because they are going to create new programs.



This shows a hat block detached from the program. If the program blocks are too close the hat block, they will snap together.

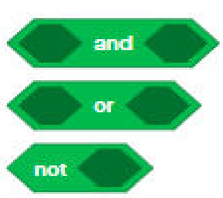
Ask students to move the blocks to the right on the screen, so they are available, but out of the way. **Do not move code left** – the blocks will be deleted. The screen shot below shows programs moved to the right. The area on the center left will be used to create new programs.



Note: If students have not completed the lesson using **My Blocks**, they should complete the Simplifying Code lesson before continuing.

On the palette, have students choose Operators. Point out to students that they have multiple operators from which to choose. Remember that "+", "-", and "<" are all operators. When they evaluate an expression that combines numbers with the "+" operator, the result is numeric. When they evaluate an expression that combines numbers with "<", the result is a Boolean expression.

Focus students on the green hexagons with "and", "or" "not" containing blank hexagonal shapes or fields. These will be needed for the next part of the program they will add. Explain to students that the green operators containing "and" and "or" require two conditions. The green operator containing "not" requires one condition.



Operators with blank fields

Ask students to write pseudocode first, then write code so the robot stops moving if the color sensor senses red **or** if the distance sensor senses an object less than 15 cm away.

Explain to students that in computer operators, the word "**or**" means it sees red or it senses an object less than 15 cm away or **both**. Either condition can be met or both conditions can be met. Students will need to add a Boolean operator.

> **Note:** In English, we might say "You can have chocolate or vanilla ice cream" meaning you choose one. In programming the "or" means either or both! That's better, right – if you like ice cream!

If the color sensor does not sense red or the distance sensor senses the distance to an object is greater than 15 cm, the robot should continue moving. Test your robot by placing your hand over the ultrasonic sensor or letting it move over a red line. Either action should stop the motors. Both conditions can apply, not just one.

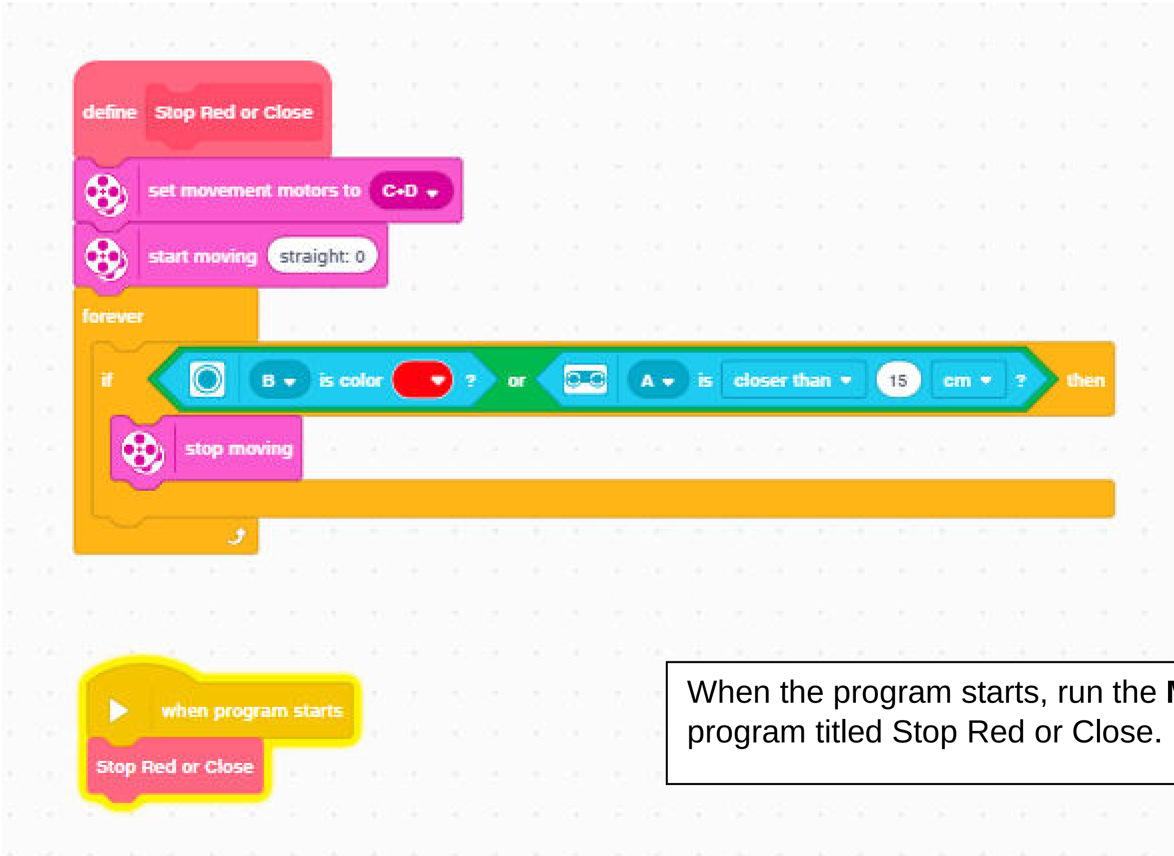
> **Note:** If students are struggling, you can show them this example:

```
Repeat forever
    If the color sensor senses red or
    If the distance sensor senses an object greater than 15 cm away
        Stop moving
    Else, keep moving
```

Ask students to create a **My Block** for the correct code. Remind students to name the **My Block** appropriately so they will remember what code the block represents.

If students struggle, shown them this example.

> Define the **My Block** called Stop Red or Close
>
> Motors C&D will start moving straight
>
> If the color sensor senses red or the distance sensor senses an object less than 15 cm away, stop the motors
>
> Else keep going straight
>
> Do this until one of the conditions is met.



When the program starts, run the **My Block** program titled Stop Red or Close.

3. Explain

Ask students to explain:

- What the Boolean operator "and" does.
- What the Boolean operator "or" does.
- The difference between how "or" in English works compared to "or" in computer science using a Boolean operator.
- How the Boolean expression they used in their code works.

4. Elaborate

Tell students they will be adding music/sound. The music should play regardless of whether the robot stops or not. They will create a new stack.
Have students create a **My Block** or use the **My Block** called "Sound" created previously.

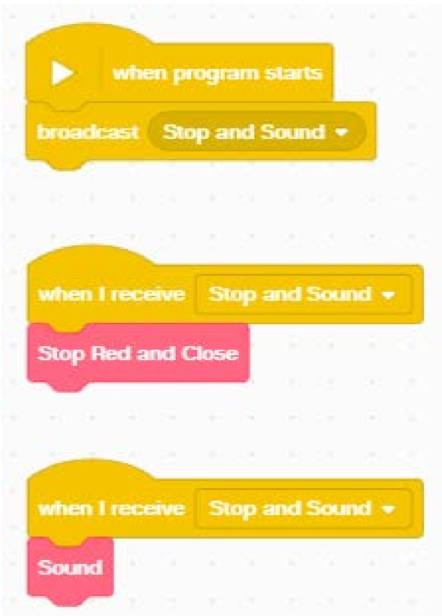
> **Note:** Because a **My Block** cannot be copied from one program to the next, students will have to return to the previous program if they wish to use a previously created **My Block**. Students should iterate with the debugging process until their robot works correctly.

All student programs should have two **My Blocks** that play simultaneously. Each **My Block** has subcomponents.

Note: A **My Block** can even contain another **My Block** if it can make programming more efficient.

Their newest **My Block** should contain a Boolean operator.

If students struggle, show them this example.



When the program starts, broadcast the message Stop and Sound.

When the message Stop and Sound is received, run the **My Block** program Stop Red or Close

and run the **My Block** program Sound at the same time.

Optional Extension:
Have students add other actions. For example, add lights formations, when the color sensor sees different colors.

5. Evaluate

Teacher Assessment

Ask students what challenges they encountered in trying to write the programs using Boolean logic.
Ask students to reflect in their journals on how evaluating Boolean expressions are alike and different from typical English speech.

Self-Assessment

Have students answer the following in their journals:

- What was one thing you enjoyed today?
- How could Boolean operators be helpful with the end of unit challenge?
- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on your time management today.
- Using a scale of 1-3, rate yourself on your materials (parts) management today.

LEGO® Education SPIKE™ Prime

Mini-Challenge: Security Alarm Using Operators

Grade 6-8 | **90 minutes** | **Intermediate**

Mini-Challenge: Security Alarm Using Operators

Students will investigate how to use operators to make a security alarm using sensors.

Questions to Investigate

- How can operators be used with sensors to create a more complex conditional statement?

Materials Needed

- SPIKE Prime sets ready for student use. Prior to the first lesson, please visit the following website for help with set up, kit organization and SPIKE App https://education.lego.com/en-us/start/spike-prime/intro
- Devices with the SPIKE App installed.
- Student journals

Prepare

- Ensure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.

1. Engage

Engage students in a conversation about security.

Ask students to think about places that they keep secure (i.e., phones, computers, doors, etc.). Have students share how these areas are kept secure.

Launch a discussion about physical security options and ways to detect "break ins". Prompt students to think about ideas like detecting movement or setting passwords.

2. Explore

Students will investigate using two steps for more security.

KEY OBJECTIVES

Students will:

- Program an alarm using Boolean operators.
- Use pseudocode to help write code that must meet multiple requirements.
- Iterate and debug coding to create a working security device.

STANDARDS

CSTA

2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms.
2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.
2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution
2-AP-17 Systematically test and refine programs using a range of test cases.
2-AP-19 Document programs in order to make them easier to follow, test, and debug.
3A-AP-17 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

VOCABULARY

Boolean operator

Direct students to the **BUILD** section in the SPIKE App. Here students can access the building instructions for the **Quality Check Robot** model. Ask students to build the model. The building instructions are also available at https://education.lego.com/en-us/support/spike-prime/building-instructions.

Ask students to create a security device that alerts you to movement and then requires a color-coded password. Discuss the two sensors included in the model and how these might be used to create a physical alert device.

Ask students to write a pseudocode program that explains how their program should work. The program must include:

- The distance sensor to detect motion
- The color sensor as a pass code
- At least one Boolean operator

Note: Students may need to be reminded about the operators. Have them open Operators in the palette and look at their choices. Ask them what operators make the most sense when using the distance sensor and when using the color sensor. Have students explain how the Boolean operators "and", "or", and "not" work as well as the "<", "=", and ">" symbols.

Ask students to use their pseudocode to create their program. Remind students to use code comments in their program to indicate the expected actions.

Allow students time to test and modify their program as needed.

3. Explain

Discuss various program examples together as a class to determine what is effective. Have students share their devices and programs freely with each other.
Ask students questions like:

- How can sensors be used to create a security device?
- How did the logic operator work in your program?

4. Elaborate

Challenge students to use a different logic operator in their security device. For example, if they used "and" perhaps now they try "not" or they can add a second operator.

Ask students to write a new pseudocode to change the way the program will work by using a different logic operator. Students can add a second logic operator to their program or replace the one in their current program.

Allow students time to modify their program according to the new pseudocode. Students should test and refine their program.

5. Evaluate

Teacher Assessment

Discuss the programs with students.
Ask students questions like:

- What happened when you tried to use sensors in your security device?
- What operators did you use and why?
- How secure do you think your device is?

Self-Assessment

Have students answer the following in their journals:

- What did you learn today about using operators?
- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on your time management today.
- Using a scale of 1-3, rate yourself on your materials (parts) management today.

LEGO® Education SPIKE™ Prime

Connecting to Careers: Hospitality & Tourism and Government & Public Administration

Grades 6-8 | 90 minutes | Beginner

Connecting to Careers: Hospitality & Tourism and Government & Public Administration

In this lesson series, students will have the opportunity to explore and research careers.

Prepare

Prior to starting the lesson, prepare the following:

- Set aside enough LEGO® bricks for students' models.
- Make sure you have enough devices and access to the Internet for student use during this lesson.
- Prepare images of different jobs. There should be a variety of images to show different jobs within the 16 career clusters. (Examples: Front Desk Associate, Tour Guide, Mayor, City Manager, Caterer, City Council Member, Custodian, Housekeeping Staff).

1. Engage

Ignite a discussion with students:

- What are some of the jobs adults you know have?
- What do you think they like about their jobs? Is there such a thing as the perfect job?

Present the images of the jobs in the area of Government & Public Administration and Hospitality & Tourism to students—as a class, decide how you would categorize the jobs.

Ask questions like:

- What kind of environments are associated with these jobs?
- Do any of these jobs interact or rely on one another? If so, how?

KEY OBJECTIVES

Students will:

- Articulate their personal interests and goals.
- Relate their personal interests and goals into possible career pathways.
- Explore various careers in career pathways.

STANDARDS

Career Ready Practice 10- Plan education and career path aligned to personal goals. (CCTC)

VOCABULARY

Career
Career Cluster
Qualifications
Skills
Education
Knowledge

Ask students to think about what interests them. What kind of job(s) would they like to have in the future?

2. Explore

Students will be assigned two career clusters each time career connections lessons are taught. By the end of the course, students will have explored all 16 career clusters.

There are 16 career clusters:

- Agriculture, Food & Natural Resources
- Architecture & Construction
- Arts, A/V Technology & Communications
- Business Management & Administration
- Education & Training
- Finance
- Government & Public Administration
- Health Science
- Hospitality & Tourism
- Human Services
- Information Technology
- Law, Public Safety, Corrections & Security
- Manufacturing
- Marketing
- Science, Technology, Engineering & Mathematics
- Transportation, Distribution & Logistics

The two career pathways we are studying today are:

- **Government & Public Administration**
- **Hospitality & Tourism**

Ask students to think about the lessons in Unit 7, especially the Mini-Challenge. Discuss how the lessons relate to the career pathways. (Both career pathways need to have security devices to help keep people safe; there are programs in both career pathways that require evaluation of criteria or constraints to qualify for jobs, benefits, special offers, and so forth.)

Ask students to brainstorm as many jobs as you can within each career cluster. Have one person write down the jobs as they are named. Have students make a list of three top skills for 10 of the jobs named and create an graphic organizer to show where skills overlap between jobs. Based the on graphic organizer, what skills seem to be most important?

In small groups of 4, students complete online research to find out more about each career clusters. Allow time to get information about the jobs that included, especially jobs they had not heard of or had not thought of in their brainstorm.

Students should also research:

- Skills needed
- Forecast of future job openings (and current)
- Education levels, certifications, licenses, apprenticeships, etc. are required

In their groups, students create a visual representation (build a model) of one of the two career clusters. They will build one or more physical models with LEGO® bricks. Each person can build a model or the group can create one large model that represents all ideas about one of the career clusters.

Each group will be responsible for a quick one-minute presentation of their LEGO® model(s) to explain how it represents their career cluster.

3. Explain

When the students have finished building, allow each group to present their model.
Ask students:

- To tell about your LEGO® build.
- What kind of interests would these career clusters have?
- What are some of examples of jobs in these career clusters?
- What jobs are entry level and what jobs require experience in a field?
- How are these career clusters similar to one another?

4. Elaborate

Ask students if there were any jobs in today's career cluster that were interesting to them.

Discuss the following with students:

Have a discussion around each of the following:

- How could a Security System relate to one or more of the career pathways? (For example, governments have a lot of data as well as physical objects and buildings that need to be secured.)
- Hotels and restaurants want guests to feel secure – physically as well as the data like credit card information. If people do not feel secure, they will not return and the businesses will fail. What skills did students use in Unit 7 that would help them in either of these areas?
- Programs such as frequent flyer, returning restaurant visits, etc., usually have criteria to earn miles, points, etc. Frequently there are multiple criteria that are

used to determine if points are earned and how many points are gained. It may be that a certain dollar level purchase must be made and/or certain items must be purchased. For example, a customer may have to purchase a ticket for more than $100, check a bag, and pay for a seat assignment in order to earn miles. Or in order to get a buy-one-get-one free sandwich, a customer must purchase one of three specific sandwiches and buy a drink. Ask students how these relate to evaluating Boolean statements.

5. Evaluate

Evaluate the students' skills development by observing if they:

- Articulate their personal interests and goals.
- Relate projects into possible career pathways.
- Explore various careers in career pathways.

LEGO® Education SPIKE™ Prime

Game with Variables

Grade 6-8 | **45-90 minutes** | **Intermediate**

Game with Variables

Create a game for two players that will keep the score.

Questions to Investigate

- How do engineers and programmers work together to create games?

Materials Needed

- SPIKE Prime Sets
- Device with SPIKE App installed
- Student journals

Prepare

- Check to make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.

1. Engage

Ignite a discussion with students about games they have played – card games, board games, video games, sports, and so forth. Ask students to explain:

- What game(s) do they like to play and why?
- What makes a game fun?

2. Explore

Tell students they will be designing a game for two people to play. Encourage students to be creative and to have fun.

Have students find the **Goal!** lesson in the **Extra Resources** section of **Start**. Ask them to complete **Steps 01-03** and see what the program does. Each team should modify the models and the program. Remind students to have fun and be creative. Brainstorm ideas and try something new.

When the models are complete and the programs work well, have students write the rules for their game. Prompt students to think about what each player does and how a player can score points. Students should name their game.

KEY OBJECTIVES

Students will:

- Create a game for two players.
- Keep score by using variables.

STANDARDS

CSTA

2-CS-02 Design projects that combine hardware and software components to collect and exchange data.
2-CS-03 Systematically identify and fix problems with computing devices and their components.
2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.
2-AP-15 Seek and incorporate feedback from team members and users to refine a solution that meets user needs.
2-AP-17 Systematically test and refine programs using a range of test cases.
3A-AP-13 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
3A-AP-16 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.

VOCABULARY

Variable

3. Explain

Allow students to share their games.
Ask students:

- What is the name of their game?
- How do their game pieces work together?
- How does a player score a goal?
- When is the game over?

4. Elaborate

Have students add a way to keep score by using the buttons on the front of the hub. The display in lights should show who scored each time the button is pressed. Students should add variable blocks to keep track of the score on screen.

As teams complete their projects, have them pair up and play both games. Students may clarify the rules as needed, to make it a fair game. Consider the first time the game is played as an editing check on the rules and program.

All games should be played by at least 3 different teams. It is preferable for all teams to play all games.

5. Evaluate

Evaluate the students' ability to use variables to keep score.
Evaluate the students' ability to communicate the rules of the game.

Have students answer the following in their journals:

- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on your time management today.
- Using a scale of 1-3, rate yourself on your materials (parts) management today.

LEGO® Education SPIKE™ Prime

Compound Conditionals

Grade 6-8 | 90 minutes | Advanced

Compound Conditionals

Utilize compound conditionals to mimic passwords used to unlock a box.

Questions to Investigate

- Why would an engineer need to keep information or materials safe?
- Why are some ways that information or materials are kept safe?

Materials Needed

- SPIKE Prime sets
- Devices with SPIKE App installed
- Student journals

Prepare

- Make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.

1. Engage

Have students work with a partner. Partner A should write 3 whole numbers between 0 and 9 in his/her journal without Partner B seeing what they are writing. Partner B should guess the first number written. Partner A sees how many guesses it takes and writes the number of guesses underneath the guessed number. Next, Partner B guesses the second number. Again, Partner A writes how many guesses it takes. Finally, Partner B guesses the third number. Partner A writes how many guesses it takes.

Switch roles. Partner B writes 3 numbers in his/her journal. Partner A guesses the numbers and then Partner B writes how many guesses it takes.

Ask students how many possible answers there are for each number. There are 10 possible answers– 0,1,2,3,4,5,6,7,8,9. How many possible answers are there for a 3-digit passcode? (10 x 10 x 10) [numbers 000 to 999]

KEY OBJECTIVES

Students will:

- Identify ways to create a more secure entrance into a device.
- Use conditional statements and compound conditionals to mimic passwords that unlock a box.

CSTA

2-CS-01 Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices
2-CS-02 Design projects that combine hardware and software components to collect and exchange data.
2-CS-03 Systematically identify and fix problems with computing devices and their components.
2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms.
2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution
2-AP-17 Systematically test and refine programs using a range of test cases.
2-AP-19 Document programs in order to make them easier to follow, test, and debug.
3A-AP-13 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
3A-AP-21 Evaluate and refine computational artifacts to make them more usable and accessible
2-IC-20 Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.
2-DA-09 Refine computational models based on the data they have generated.

VOCABULARY

Compound conditional
Complexity

LEGO education

Ask students how many guesses of the 10 possible it took for each number to be found. Have them create a decimal number based on how many guesses it took to find a number (An example would be 6 guesses required of 10 possible guesses = .6 for the first number). Ask students to talk with their partners and suggest other ideas for making a three-unit password that would be more difficult to guess. What would be a good number of required items in a passcode?

Discuss as a class, ways to increase the difficulty (Examples might be to add alphabet characters – 26 instead of 10 possible answers; adding special characters to the numbers; differentiating between capital and lowercase letters – 52 possibilities; limiting the number of chances before being locked out, or having a time limit for a correct answer).

Ask students to consider adding gears for degrees moved, color recognition, or other items only found in robotics to increase the level of difficulty for a passcode. What would they add and why?

2. Explore

Students will build and program a lock box and then increase the difficulty of getting into it.

Have students find the **Keep it Safe** lesson in **Kickstart a Business**. Build the Safe-Deposit Box. Students should complete **Steps 01-03**. Ask them to look closely at the code. What needs to be done to make the safe unlock?

Allow students time to investigate the model and program.

Challenge students to make their safe more secure.

Allow students time to complete **Step 04** and investigate the program.

Discuss the new program. Ask students:

- How was the complexity increased?
- Was the box still easy to open?
- What must happen before the box is unlocked?

Challenge students to complete **Step 05**. Students need to add two more actions that must be done in order. Have students use the three vertical dots at the bottom of the block palette to open the entire array of coding blocks. The students do not have to move only the dial.

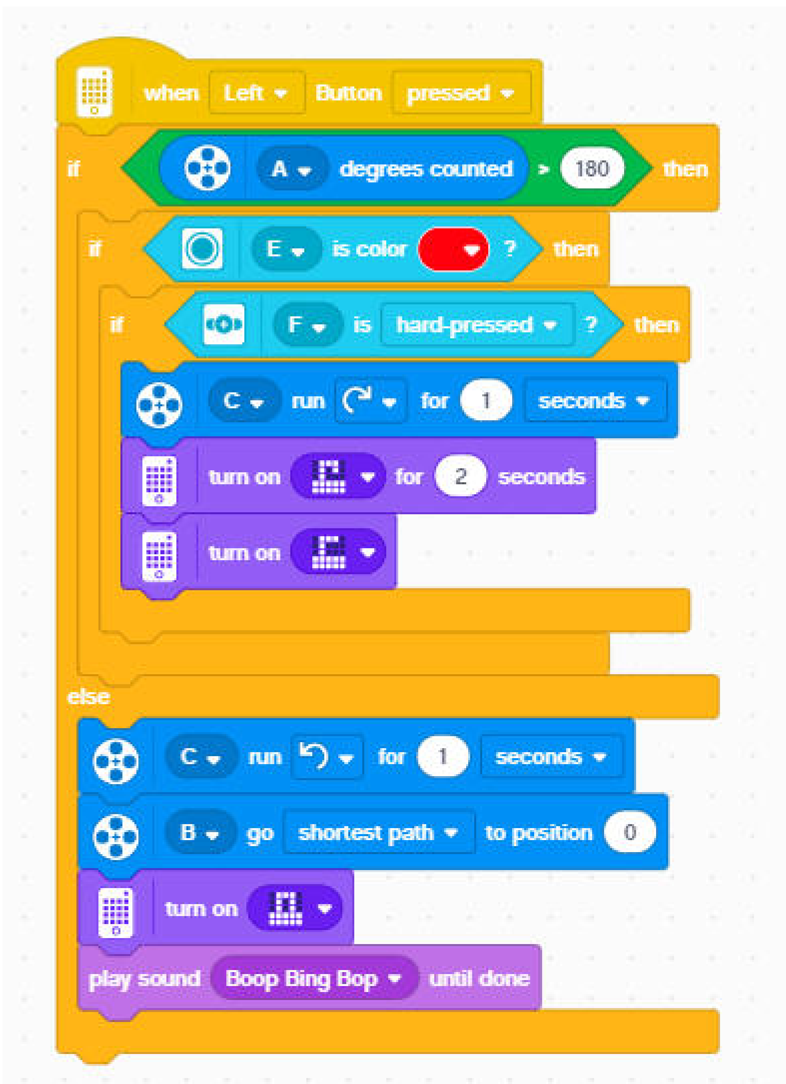
Tell students they can use a color sensor, force sensor, distance sensor, motor rotation sensor, button pushes, and so forth as part of a code that requires multiple actions to be taken in a specific order before the box will unlock. Students should write pseudocode first and then use comments while writing code.

3. Explain

Discuss with students how the safe works.
Ask students:

- How does the box open?
- Was the safe easy to open?
- How secure is the safe?
- What are some things that could be added or changed that would make it more secure?

If students are struggling, you can show them this example of code.



When the left button is pressed

If motor A turns more than 180 degrees and

If the color red is detected and

If the force sensor is hard pressed

Then run motor C clockwise for 1 second and

Turn on the hub lights for 2 seconds

Else

Run motor C counterclockwise for 1 second and

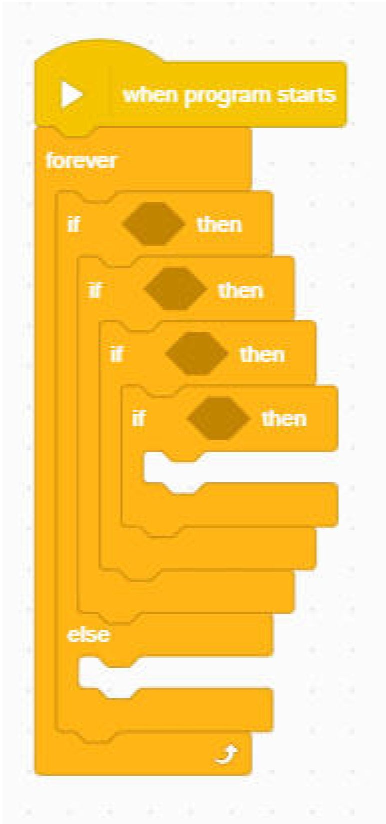
Run motor B to position 0

Turn on hub lights and

Play Sound

Discuss if/then and if/then/else statements. If they nested these statements together, they could create a string of actions that must be taken before the box will unlock. **Nested** means that one if/then statement contains an if/then statement.

If students struggle, show them this example. Note – the operators are empty.



4. Elaborate

Allow students time to create and expand their programs. Add another step or two to open the safe.

Discuss the more secure programs together as a group.
Ask students:

- How was the complexity increased?
- Was the safe still as easy to open?
- Could the process become so complicated that even the owner would not want to open it?
- How do you create a balance between security and simplicity?

5. Evaluate

Evaluate the students' understanding of how to make a passcode more secure.
Ask students to explain how a compound conditional works.

Have students answer the following in their journals:

- What was the most challenging aspect of creating a secure device?
- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on your time management today.
- Using a scale of 1-3, rate yourself on your materials (parts) management today.

LEGO® Education SPIKE™ Prime

Compounding Conditionals

Grade 6-8	90 minutes	Advanced

Compound Conditionals

Utilize compound conditionals to mimic passwords for unlocking a device.

Questions to Investigate

- Why would an engineer need to keep information or materials safe?
- Why are some ways that information or materials are kept safe?

Materials Needed

- SPIKE Prime sets
- Devices with SPIKE app installed
- Student journals

Prepare

- Make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.

1. Engage

Tell students they are going to review security and then make their safes even more secure.

Briefly show students a stack of six 2x4 bricks from the set. Tell students you have assigned a number to each brick 0 to 9 (The numbers are **not** on the bricks, you just tell them). Explain to students that they cannot take notes. Be sure not to write the numbers out for students.

For example, the set could look like this:

- Yellow 3
- Red 1
- Green 7
- Blue 2
- Magenta 0
- Green 9

KEY OBJECTIVES

Students will:

- Identify ways to create a more secure entrance into a device.
- Build and program a device to keep an object very secure.
- Use compound conditionals to mimic multi-factor authentication.

CSTA

2-CS-01 Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices

2-CS-02 Design projects that combine hardware and software components to collect and exchange data.

2-CS-03 Systematically identify and fix problems with computing devices and their components.

2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms.

2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.

2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution

2-AP-17 Systematically test and refine programs using a range of test cases.

2-AP-19 Document programs in order to make them easier to follow, test, and debug.

3A-AP-13 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.

3A-AP-21 Evaluate and refine computational artifacts to make them more usable and accessible

2-IC-20 Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.

2-DA-09 Refine computational models based on the data they have generated.

VOCABULARY

Compound conditional

Move the bricks out of sight of the students.

Ask students to name the brick color **or** the number you have assigned starting from the bottom brick. Allow each student one guess until everyone has had a turn or the bricks have been put in the correct order. Allow as many additional turns as needed. As they are guessing, have one student record with tick marks the number of guesses needed to get the color or the number correct for each brick.

At the end, show students how many guesses it took to get all 6 bricks placed in the correct order. Discuss the number as a group.

A Second Chance

Shuffle the bricks and make a new stack. Change the numbers that go with each color. For example, the set might look like this:

- Blue 3
- Red 4
- Yellow 5
- Magenta 9
- Blue 6
- Green 9

Ask students to name the brick color **and** the number you have assigned starting from the bottom brick. Each guess must contain a color and a number. Allow each student one guess until everyone has had a turn or the bricks have been put in the correct order. Allow as many additional turns as needed. Have one student record with tick marks the number of guesses needed to get the color and the number correct for each brick. How many guesses did it take to get all 6 bricks and numbers placed in the correct order?

At the end, show students how many guesses it took to get all 6 bricks placed in the correct order. How did it compare with the number of guesses to guess only the color sequence?

Ask students to explain how the guessing of color or color and number relate to “and” and “or” conditional statements.

Ask students to explain why the number of guesses for an “and” statement is likely more than for and “or” statement. Discuss how this activity mimics security and password protection.

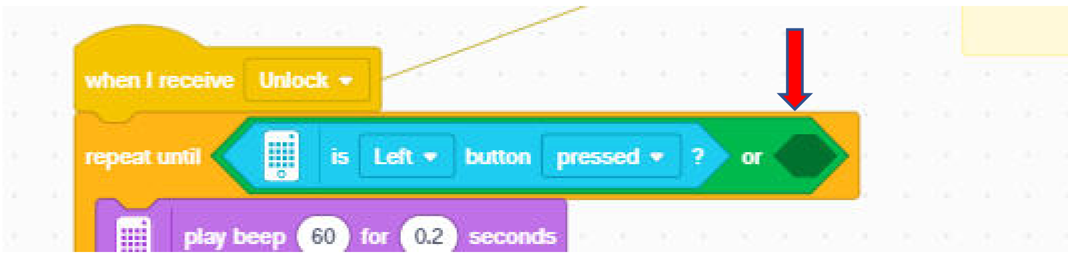
2. Explore

Students will increase the security on their safe box.

If students have the Safe-Deposit Box from the **Keep it Safe** lesson in **Kickstart a Business**, they can use it in this lesson. They should add the arm to the box which requires only **Steps 44-48**.

If students do not have the Safe-Deposit Box from the **Keep it Safe** lesson in **Kickstart a Business**, they should find the **Keep it Really Safe** lesson in **Kickstart a Business**. Build the safe-deposit box.

Have students complete **Step 03**. Ask students to look at the sample code. They should notice that the second operator in the repeat until is blank (Example is shown below with red arrow pointing to black).



Discuss the program with students. Ask students:

- If leaving a field blank is a good idea.
- What could they add to make a better program?
- Was the level of complexity to open the safe increased or decreased? Why?
- What are the conditions?
- What happens when one the conditions is not met?

Add a Layer of Security
Have students complete **Step 04**. Review with students what is happening in the code.

Add Another Layer of Security

Have students complete **Step 05**. Ask students to change the program to require **both** conditions to be true (use “and” instead of “or.”) before the lock will open.

3. Explain
Discuss the layers of security and how the programs work.

Ask students:
- To compare the security of the two programs in **Steps 04 and 05**. Which program is more secure and why?
- What is a compound conditional? (Two nested conditionals)

- Why isn't the current program a compound conditional, even though there are two conditions? (Two conditionals do not make a compound conditional. Two nested conditionals are required to make a compound conditional.)
- To compare the program for **Step 04** with the program for **Step 05**. Which program makes a more secure deposit box? Why?
- How did they use a Boolean expression?

4. Elaborate

Students will create a very secure locked box.

Ask students to complete **Step 06**. Have students pretend they are spies. They need to leave secret messages inside for their contact. Ask student to write pseudocode instructions that could be texted to the agent on how to enter the box. Ask students to consider adding a sensor, adding a series of gear movements, button presses, and so forth.

Have students build and program the model, adding elements as desired. Students may remove the device that covers the gear if they wish. When they have a working model and excellent pseudocode instructions, students will share their model.

Have teams exchange pseudocode and models to check if the instructions could be followed accurately. Can most teams open the model within three attempts?

Teams may modify the pseudocode, the model, and the code as needed. Students should help one another with coding and model construction. They should use the comment feature to give credit to others for their help.

Challenge students to make the entry into the safe as difficult as they can and still be able to open it 90% of the time.

5. Evaluate

Evaluate the students' understanding of how to make a passcode more secure.
Ask students to explain what conditionals they used in their final super safe box.
Ask students to explain examples of very secure safe boxes they encountered.

Have students answer the following in their journals:

- What ideas from using compound conditionals could be useful when solving the culminating activity challenges?
- What part of this lesson did you enjoy the most? Why?
- What characteristics of a good teammate did you display today?
- Using a scale of 1-3, rate yourself on your time management today.
- Using a scale of 1-3, rate yourself on your materials (parts) management today.

LEGO® Education SPIKE™ Prime

Mini-Challenge: Break Out Room

Grade 6-8 | **135 minutes** | **Advanced**

Mini-Challenge: Break Out Room

Students will create and program their own breakout security system.

Questions to Investigate

- How can conditions be set to create a secure "room" from which people can break out?

Materials Needed

- SPIKE Prime sets ready for student use. Prior to the first lesson, please visit the following website for help with set up, kit organization and SPIKE App https://education.lego.com/en-us/start/spike-prime/intro
- Devices with the SPIKE App installed.
- Student journals

Prepare

Ensure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.

1. Engage

Engage students in a conversation about escape rooms.

Ask students if they have ever participated in or heard of a breakout or escape room. Consider showing images or videos to students. Explain that in these games, participants use clues and complete puzzles to figure out how to get out of the locked room.

Brainstorm

Brainstorm different ideas of puzzles that could be used in a breakout room.

As a group, brainstorm several ideas for creating a secure device that could be used in a breakout or escape room.

KEY OBJECTIVES

Students will:

- Write code that includes conditions that must be met in a game format
- Create a game that requires a series of robot responses in sequence

STANDARDS

CSTA

2-CS-01 Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices

2-CS-02 Design projects that combine hardware and software components to collect and exchange data.

2-CS-03 Systematically identify and fix problems with computing devices and their components.

2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms.

2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.

2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.

2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution

2-AP-17 Systematically test and refine programs using a range of test cases.

2-AP-19 Document programs in order to make them easier to follow, test, and debug.

3A-AP-16 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.

3A-AP-17 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

2-DA-09 Refine computational models based on the data they have generated.

VOCABULARY

Constraint

Ask students to then work in their smaller groups to create a short survey to ask other students what they would like in a security system or something to breakout of or escape from. Students can share examples and ask for other ideas. Students should include 3 questions about what would be fun, what would be challenging, and what themes are appealing. Students want to ensure user interest is high in their escape device and also that the idea is age-appropriate.

Allow students time to create and complete their survey within the class or arrange to have students survey another class.

2. Explore

Students will design, build, and program a puzzle game to use in a breakout room.

Design and Choose the Right Idea

Students should consider the input from their surveys and start to design their escape room or security device.

Students will design, build, and program a device that could be used in a breakout or escape room. The constraints are:

- It must use the light matrix
- It must use at least one motor
- It must use at least one sensor
- It must include at least one Boolean operator
- It must include at least two steps to crack the code.
- It must include a timer feature

Students should create a sketch of their building idea and a flowchart or pseudocode of their programming idea.

Test and Iterate

Allow time for students to test and analyze their idea as they go, making improvements where needed. Students should test and evaluate their designs against the design criteria. They should refine their flowcharts as they make modifications to their idea.

Ensure students use sketches and photos of their models to record their design journey. Students should use comments to document their programs.

Allow students to receive feedback on their designs as time allows. This can be from other groups or the teacher.

3. Explain

Students should share their design idea and explain how it will work. Conduct an initial sharing session with students.

Ask students questions like:

- How do you plan to program your model to create a security device with at least two steps?
- What decisions did you have to make while creating your design?
- What type of conditional statement will you choose?

4. Elaborate

Encourage students to incorporate any new ideas they got from the sharing session. Allow students time to build their model and write the code. They should start by writing pseudocode. Students will need time to modify the model and debug their initial code. Remind students to add comments to their code to help with troubleshooting and for clarification. Remind students that using My Blocks may be helpful.

Students should finalize their design and code.

Have half the class share their game with another team. Allow teams to modify their games based on the experience. Next, have the other half of the class share their games. Again, allow teams to modify their games based on the experience. Try to allow enough time for playing games so that teams can experience at least four different games.

Ignite a discussion on what features of games were fun, interesting, challenging. Ask students to give examples of games they played that they enjoyed.

5. Evaluate

Teacher Assessment

Evaluate students' understanding of Boolean operators.
Evaluate students' understanding of how software and hardware work together.

Ask students questions like:

- What was difficult about this challenge?
- What was your approach to solving this challenge?
- What type of logic operators did you include and why?

Self-Assessment

Have students answer the following in their journals:

- What did you learn today about creating your own design when it must be based on given criteria?
- What characteristics of a good teammate did I display today?
- Using a scale of 1-3, rate yourself on your time management today.
- Using a scale of 1-3, rate yourself on your materials (parts) management today.

Connecting to Careers: Architecture & Construction and Science, Technology, Engineering, and Mathematics

Grades 6-8 | 90 minutes | Beginner

Connecting to Careers: Architecture & Construction and Science, Technology, Engineering, & Mathematics

In this lesson series, students will have the opportunity to explore and research careers.

Prepare

Prior to starting the lesson, prepare the following:

- Set aside enough LEGO® bricks for students' models.
- Make sure you have enough devices and access to the Internet for student use during this lesson.
- Make copies of the handouts (if desired) or place into a digital platform for student use.
- Prepare images of different jobs. There should be a variety of images to show different jobs within the 16 career clusters (Examples: Architect, Electrical Engineer, Electrician, Plumber, Cabinet Maker, Biomedical Engineer, Statistician, Chemist, Maintenance Technician, Carpet Installer, Roofer, Cement and Concrete Finisher).

1. Engage

Ignite a discussion with students about the lessons completed in Unit 8 and what jobs come to mind.

Present the images of the jobs in the area of Architecture & Construction and Science, Technology, Engineering & Mathematics to students—as a class, decide how you would categorize the jobs. Ask questions like:

- What jobs belong together?

KEY OBJECTIVES

Students will:

- Articulate their personal interests and goals.
- Relate their personal interests and goals into possible career pathways.
- Explore various careers in career pathways.

STANDARDS

Career Ready Practice 10- Plan education and career path aligned to personal goals. (CCTC)

VOCABULARY

Career
Career Cluster
Qualifications
Skills
Education
Knowledge

- What kind of similar skills do these different jobs use?
- What kind of environments are associated with these jobs?
- Do any of these jobs interact or rely on one another? If so, how?

2. Explore

Note: These are the final two career clusters to be researched.

There are 16 career clusters:

- Agriculture, Food & Natural Resources
- Architecture & Construction
- Arts, A/V Technology & Communications
- Business Management & Administration
- Education & Training
- Finance
- Government & Public Administration
- Health Science
- Hospitality & Tourism
- Human Services
- Information Technology
- Law, Public Safety, Corrections & Security
- Manufacturing
- Marketing
- Science, Technology, Engineering & Mathematics
- Transportation, Distribution & Logistics

The two career pathways we are studying today are:

- **Architecture & Construction**
- **Science, Technology, Engineering & Mathematics**

Ask students to brainstorm as many jobs as you can within each career cluster. Have one person write down the jobs as they are named.

Ask students to think about the lessons in Unit 8 and how they relate to the career pathways being studied. What connections can students draw between the careers and the skills, models, coding, and so forth utilized in Unit 8? (Construction of a secure box, mathematics relates to the password and possible number of attempts and potential correct answers, science – how your hands and eyes can work together and the speed at which they can complete tasks, technology – how coding could affect the security level, engineering – model creation)

Ask students which area they would be most interested in or if they find multiple careers interesting based on the lessons in Unit 8.

Have students create a Venn Diagram between the two career pathways. Ask students to write skills that overlap between the two pathways and skills that are only required for either pathway. Discuss as a class their ideas.

In small groups of 4, students complete online research to find out more about each career clusters. Allow time to get information about the jobs that included, especially jobs they had not heard of or had not thought of in their brainstorm.

Students should also research:

- Skills needed
- Forecast of future job openings (and current)
- Education level, certifications, licenses, apprenticeships, etc. are required

In their groups, students create a visual representation (build a model) of one of the two career clusters. They will build one or more physical models with LEGO® bricks. Each person can build a model, or the group can create one large model that represents all ideas about one of the career clusters.

Each group will be responsible for a quick one-minute presentation of their LEGO® model(s) to explain how it represents their career cluster.

3. Explain

When the students have finished building, allow each group to present their model. Ask students questions like:

- Tell us about your LEGO® build.
- What kind of interests would these career clusters have?
- What are some of examples of jobs in these career clusters?
- Can you explain the difference between a job and career cluster?
- How are these career clusters similar to one another?

4. Elaborate

Ask students if there were any jobs in today's career cluster that were interesting to them.

In their group, students create a visual representation (build a model) of one of the two career clusters. They will build one or more physical models with LEGO® bricks. Each person can build a model, or the group can create one large model that represents all ideas about one of the career clusters.

Each group will be responsible for a quick one-minute presentation of their LEGO® model(s) to explain how it represents their career cluster.

Students should give examples of how these career pathways relate to the most recent projects using conditional statements and compound conditionals.

Quickly review the 16 Career Pathways with the class. Have each student complete the My Career Plans worksheet. (This is the same worksheet they completed during the first Connecting with Careers lesson.) They will utilize this information and the information on careers in their journals in the next part of Connecting with Careers. In the next lesson, this worksheet will be compared with their first version of career plans completed at the beginning of the course to see changes and growth.

5. Evaluate

Evaluate the students' skills development by observing if they:

- Articulate their personal interests and goals.
- Relate their personal interests and goals into possible career pathways.
- Explore various careers in career pathways.

Name: ______________________________________ Date: ________________

My Career Plans

WHO AM I?

Things I Am Interested In:	My Favorite Subjects in School:
1.	1.
2.	2.
3.	3.

Things I Am Good At:	Things I Do Not Like to Do:
1.	1.
2.	2.
3.	3.

I need to improve on ____________________ because ______________________.
1.
2.
3.

LEGO education

Culminating Activity: Careers

Grades 6-8 | 90 minutes | Intermediate

Culminating Activity: Careers

In this lesson series, students will have the opportunity to explore and research careers that relate to a real-world situation around attendees at a convention.

Prepare

Prior to starting the lesson, prepare the following:

- Make sure you have enough devices and access to the Internet for student use during this lesson.
- Make copies of the handouts (if desired) or place into a digital platform for student use.
- Chart paper and markers
- Chart paper or slide with the 16 career clusters listed
- LEGO® bricks

1. Engage

Ignite a discussion with students. Get them thinking about traveling somewhere to learn about a topic.

Ask students questions like:

- What do you think of when you hear the word *conference* or *convention*?
- What do people do when they attend a convention?
- How do they get to the convention site?

Consider showing images of conventions and people attending conventions. Discuss what is seen in the images and what careers might be represented.

2. Explore

Tell students that throughout the school year, they have explored different jobs within the career clusters. Today, they will work on a culminating activity for those career clusters based on a convention. Students will work together in small

KEY OBJECTIVES

Students will:

- Determine the careers that impact a real-world situation.
- Articulate their personal interests.
- Relate their personal interests about careers.
- Explore various careers in the career pathways.

STANDARDS

Career Ready Practice 10- Plan education and career path aligned to personal goals. (CCTC)

VOCABULARY

Career

groups and then come together as a whole class. They will constantly refine their ideas based on feedback from others.

Explain to students that a large convention of people is coming to town. Discuss what careers will impact the people who attend the convention.

Have students work in groups of four. Have students build a model of places that conference/convention attendees would use while they are in the city. Ask students to discuss what activities the people would do in each of the places. Ask students to discuss what careers would impact the attendees at each location?

Give each group a piece of chart paper and markers. Tell students that careers in all 16 pathways will affect the experience of the people attending the convention Assign each group an equal number of the career pathways. Allow students time for research their pathways and identify ways they will be represented at the convention through building their model.

3. Explain

Allow students to discuss as a class areas that they should consider. Have each group share their models and explain the place and the activity. Ask students questions like:

- Who (in terms of jobs) might the attendees meet at each location? Why?
- What types of interactions might occur?
- Who (in terms of jobs) are behind the scene – meaning that the work they do (did) is not going on simultaneously with the visit by the attendee, but was done prior to their coming?

4. Elaborate

Ask students to create a web map or other graphic organizer that shows careers that will impact the conventioneers from each of the career pathways assigned to them.

Hang the charts around the room. Give each group another piece of chart paper.

Ignite a discussion and see how many additional jobs can be added by asking questions like the following:

- How do the roads, traffic, airports, train stations, buses, and taxis affect the people?
- How do the temperature of the rooms, the availability of food and drinks, music/sound/lighting affect the people?
- How do security, cleanliness, comfort of chairs, beds, working elevators, clean restaurants, and safe sidewalks affect the people?

- How does heath care availability, weather conditions, cell phone service, ability to access money or a way to pay for items affect the people?
- How does the purity of water or flushing toilets affect the people?

Have students continually add jobs to their chart paper (even outside their assigned career pathways).

Place all the chart papers on tables around the room. Each student should have a marker. Ask students to read each paper and circle jobs that they might consider or find interesting.

5. Evaluate

Evaluate the students' skills development by observing if they:

- Can determine jobs how affect others.
- Relate their personal interests and goals into potential career pathways.
- Explore various careers in career pathways.

LEGO® Education SPIKE Prime

My Career Interests

Grades 6-8 | 120 minutes | Beginner

My Career Interests

In this lesson, students will have the opportunity to explore and research careers that relate to their personal interests and skills.

- **Prepare**

Prior to starting the lesson, prepare the following:

- Make sure you have enough devices and access to the Internet for student use during this lesson.
- Make copies of the handouts (if desired) or place into a digital platform for student use.
- LEGO® bricks
- SPIKE Prime set with the hubs charged
- Chart paper and markers
- Chart paper or slide listing the 16 career clusters
- Chart paper with the following down the side:
    - Oral communication
    - Written communication
    - Collaboration
    - Creativity
    - Critical thinking
    - Problem solving
    - Work ethic
    - Leadership
    - Teamwork
    - Organization
    - Self-management

KEY OBJECTIVES

Students will:

- Articulate their personal interests and goals.
- Relate their personal interests and goals into possible career pathways.
- Explore various careers in career pathways.
- Compare their responses to career interests at the beginning of the course with their responses at the end of the course.

STANDARDS

Career Ready Practice 10- Plan education and career path aligned to personal goals. (CCTC)

1. Engage

Ignite a discussion with students about what skills are needed to buy and operate a car.

Ask students to consider questions like:

- How many of them would like to have their own car? Do any of them already own a car?
- What do you need to be able to read and pass a driving exam?
- What do you need to know what the contract of sale and insurance states?
- What skills do you need to have the self-discipline to follow the driving laws?
- How can you be creative and solve problems when the typical way to get somewhere is blocked?
- What skills to be able to do basic math so they can make payments and know how much insurance, maintenance, and gas will cost?

Have students create a list of skills needed to buy and operate a car. Ask students to reflect on how those skills relate to skills needed in many careers.

2. Explore

Tell students that throughout the school year, they have explored different jobs within the career clusters. Various skills are required for each career cluster. However, some skills recurred over and over.

Have students work in groups of four. Give each group a piece of chart paper and markers. Ask students to determine at least 5 skills that nearly all jobs require. Remind students that they have information in their journals. If they think of a job that doesn't require that skill, have them write it in pencil next to the skill. If students seem stuck, prompt them with ideas.

Give students access to LEGO® bricks and the SPIKE Prime set. Ask students to build models that represent the skills (minimum 5) that nearly all jobs require. As they build, if students think of more skills, have them add the skills to the chart paper.

3. Explain

Ask students to share their builds and explain the skills that are represented. Allow students to discuss the careers that use these skills.

Have students share some of the skills on their chart paper. If they have a job that they believe does not need the skill, have the class consider if that is true. Ask students if they can find a reason that the skill is actually needed?

Show students the chart paper listing skills. Tell students these are the skills employers look for most frequently when hiring for **any** position. Ask students to review their lists and compare them to your list. Discuss any skills that students did not include and any they had in their lists that are not on yours.

4. Elaborate

Have students find their two My Career Plans – Who Am I? worksheets. Each student completed a worksheet at the beginning of the course and a new version in the prior lesson.

Have students compare the My Career Plans worksheet from the beginning of the course to the one they recently completed. Ask students to think about ways in which they have grown and changed. Add an entry in their journal that explains what has changed or (if they have the exact same answers) why nothing has changed.

Ask students to discuss with a partner some of their interests and hopes for the future with regard to a career. Have students share skills they see in each other that may not be listed on the worksheets.

5. Evaluate

Evaluate the students' skills development by observing if they:

- Articulate their personal interests and goals.
- Relate their personal interests and goals into potential career pathways.
- Explore various careers in career pathways.

Name: ______________________________________ Date: ________________

My Career Plans

WHO AM I?

Things I Am Interested In:	My Favorite Subjects in School:
1.	1.
2.	2.
3.	3.

Things I Am Good At:	Things I Do Not Like to Do:
1.	1.
2.	2.
3.	3.

Skills I bring to an employer – Rate yourself 1- 10 for each skill.	
Oral Communication	
Written Communication	
Collaboration & Teamwork	
Creativity	
Critical Thinking	
Problem Solving	
Work Ethic	
Leadership	
Organization	
Self-management/discipline	

I need to improve on ____________________ because ______________________.
1.
2.
3.

My Career Reflection and Plan

Grades 6-8 | 120 minutes | Intermediate

My Career Reflection and Plan

Students create a personal plan for moving into a specific career.

Prepare

Prior to starting the lesson, prepare the following:

- Make sure you have enough devices and access to the Internet for student use during this lesson.
- Make copies of the handouts (if desired) or place into a digital platform for student use.
- LEGO® bricks
- Chart paper and markers
- Student work on chart paper from My Career Connections 1

KEY OBJECTIVES

Students will:

- Create a personal plan for moving into a specific career.
- Articulate their personal interests and goals.
- Relate their personal interests and goals into possible career pathways.
- Explore various careers in career pathways.

STANDARDS

Career Ready Practice 10- Plan education and career path aligned to personal goals. (CCTC)

VOCABULARY

Career
Career Cluster
Qualifications
Skills
Education
Knowledge

1. Engage

Ignite a discussion with students about making a plan. Discuss ideas of why you need a plan and what makes a good plan.

Use the car buying scenario with students. Explain to students that you are really interested in buying a new car. But you need to make a plan in order to know what car to purchase and to be sure you have enough money. Explain to the students that we are going to work together to outline the specific steps we need to take before we buy a car.

On chart paper, write out the specific actions to do prior to buying a car. (For example, deciding on what features you want/need, how much insurance will cost for different cars, what the gas mileage will be – and how much you plan to drive and how much gas you will have to buy, what you can afford for a monthly payment, how much cars cost that are new and used)

Explain to the students that before you can accomplish the goal of buying a car, you must strategically think about what has to happen before and during the task in order for you to be successful in buying a car that you can afford and will work well for you. This analogy of buying a car is very similar to creating an action plan to be successful in future career endeavors.

2. Explore

Remind students that throughout the school year, they have explored different jobs within the career clusters. Today, they are going to discuss their individual goals and interests and how they might align with possible jobs in specific career clusters in the future.

Provide students with a copy of the "My Career Reflection" handout. Allow students the opportunity to research any needed information. Students should also have access to the My Career Plan – Who Am I? worksheets completed previously. Students complete the My Career Reflection worksheet.

Based on their My Career Reflection handout, have students select one of their desired jobs. Provide each student with LEGO® bricks to create a physical build one desired job based on his or her interests and skills. Students should create a model and be prepared to tell about it and their career interest.

3. Explain

When the students have finished building, allow students to share and explain their model.

Ask students:

- To tell us about your LEGO® model.
- What interests you about this job?
- What are some related jobs in this career cluster?
- Why are you most interested in this job?
- What would your day-to-day responsibilities include?
- How does this job relate to your personal interests and goals?
- What personal skills are required in this career?

4. Elaborate

Have students reflect on their specific job selected. Ask students to answer these reflection questions:

CONNECT	What is your job responsible for "connecting"? (i.e., connecting people, connecting ideas, connecting physical materials, connecting equipment, etc.)

CREATE	What is your job responsible for “creating”? (i.e., creating solutions, creating physical products, creating ideas, creating programs, etc.).
CHANGE	What is your job is responsible for “changing” or improving? (i.e., changing quality of life, changing efficiency, changing perspectives, etc.

Have students write in their journals:

- What courses they plan to take in high school that reflect their career interests
- What they will do (differently) in regard to effort at improving skills that are not subject specific – like creativity, collaboration, communication, interpersonal skills, and so forth
- What part-time jobs or volunteer work or clubs would be helpful in learning more about their chosen field
- Who they know that might be of assistance to learn more about the career or where they might be able to meet potential mentors or coaches

Provide students with LEGO® bricks. Have students build models showing themselves in the career of their choice, ten years from now, successfully enjoying the career. Add bricks to show what steps were necessary for the student to obtain that future success.

Have students share their models and explain what the future will look like in their chosen career and how they were able to make that dream come true.

6. Evaluate

Evaluate the students’ skills development by observing if they can:

- Articulate their personal interests and goals.
- Relate their personal interests and goals into potential career pathways.
- Make a plan for moving into a career.

Name: ______________________________ Date: ______________

MY CAREER REFLECTION

MY ACTION PLAN

My interests lead me to these jobs: (at least 3)

The jobs are in the following career cluster(s).

____

____

____

My top three job choices are____ because_____.
1.
2.
3.

What 3 subjects do I need to focus on in school based on each job above?
1.
2.
3.

What 3 clubs and/or electives could I participate in to help me toward each of these jobs?
1.
2.
3.

What specific qualifications would I need to have for this job?

QUALIFICATIONS	
Skills Needed	
Training Needed	
Minimum Education Level	

Based on the qualifications, would I need to get an industry certification? YES NO

If yes, what kind? ____

Based on the qualifications, would I need to attend college? YES NO

If yes, what kind of degree(s) would I need? ____________________________

If yes, what major should I choose? ________________________________

If yes, what college/universities have specific programs for this job/career cluster?

____

____

____

Where are the opportunities for each of these jobs within the region I live in?

____

____

____

LEGO® Education SPIKE Prime

Physical Computing Culminating Activity

Grade 6-8 | **500 minutes** | **Advanced**

Physical Computing Culminating Activity

Students choose one real-world problem. They work together in small groups to solve the problem, then build and program a model as part of the solution. Students present their solutions to the class.

Questions to Investigate

- How can you share an idea to solve a real-world problem using a programmed model?

Materials Needed

- SPIKE Prime sets
- Devices with SPIKE App installed
- Student journals
- Materials for the challenge solutions may include items used in previous lessons such as tape, colored strips, empty plastic bottles and so forth.
- Students may wish to bring in items to use

Prepare

- Make sure SPIKE Prime hubs are charged, especially if connecting through Bluetooth.
- Access to the Challenges and the Rubric.
- Internet access for research.

KEY OBJECTIVES

Students will:

- Solve a real-world problem and design, build, and program a model as part of the solution.
- Iterate design ideas and debug program prior to the final presentation.
- Present the solution, answering questions, and showcase the model.
- Evaluate your solution, your team, and yourself.

STANDARDS

CSTA

2-AP-10
2-AP-12
2-AP-13
2-AP-16
2-AP-17
2-AP-18
2-AP-19
2-CS-01
2-CS-02
2-CS-03
2-DA-09
2-IC-20
2-IC-21
2-IC-22
3A-AP-13
3A-AP-21

1. Engage

Students should read the Challenges. They are located in the Student Guide.

Challenge 1

Natural Resources - Environmental Clean Up

Students research problems in the environment such as, but not limited to, recycling, environmental cleanup, or reducing the use of limited resources. Students choose an issue and a location for an environmental problem to solve. They create a model and present how it works to solve the problem. Students determine how well their solution works and the social impact of their solution.

We want each student to choose his/her own specific challenge and the solution. However, if students are really stuck, you may use this example:

> You are scientists and engineers who form a team to take white plastic items from the ocean. The size containers you target are soda and water bottles and plastic bags. When the sensor sees something white in the range of a grabber, the model makes a sound to alert the operator. The lights on the hub keep a count of the materials that have been captured.

Challenge 2

Transportation

Students choose A, B, C, or D in this area.

A. Transportation – Delivering to Specific Locations

Students research problems of transportation in a specific location – could be inside a building or outside in a limited area. Students choose an issue and a location for their transportation problem to solve. They create a model and present how it works to solve the problem. Students determine how well their solution works and the social impact of their solution.

B. Transportation of Materials On the Moon

Students are part of a team devising new transportation vehicles for moving materials on the moon. The vehicle will need to be able to move in, out, and around craters, be seen from towers, and carry a load of either people or materials. Students determine how well their solution works and the environmental impact of their solution.

C. Transportation of Materials In the Ocean

The ocean, a place for exploration. The students work for a company that has won the bid to create a vehicle that can handle the pressure of the depths, go into shark infested waters, or navigate sensitive ecosystems. The vehicle will also have to grab items to bring to the surface for study, show video real-time, and not bump into anything, especially coral reefs, other equipment in the ocean, or the bottom of the ocean. Students determine how well their solution works and the environmental impact of their solution.

D. Autonomous Transportation of Humans on Earth

Students have landed a job on the autonomous vehicle team! This team is creating an autonomous vehicle that can safely carry people from place to place, obeying all traffic laws, parking safely, and letting people know when the battery needs to be charged. Remember you must discuss and take into consideration people getting in and out of the vehicle, decisions that could involve evaluating choices for crashes, and visibility on the road. Students determine how well their solution works and the environmental impact of their solution.

We want each student to choose his/her own specific challenge and the solution. However, if students are really stuck, you may use this example:

> You are engineers and planners for a hospital. You must ensure that the autonomous vehicles deliver the specific materials to the correct locations. Using sensors on a wheeled vehicle, the model must take vials of blood to the laboratory, x-rays to reviewing doctors, escort visually impaired persons to the surgery center, and family members to the gift shop. Your machine needs to transport all materials safely, not run into objects or people, and show the location on the brick of where it is headed.

Challenge 3

Health Science and STEM: Game for Mental Agility

Students research problems of mental agility by helping people keep the brain active and learning new things. Students choose ways to help people stay mentally agile and create a game that incorporates multiple tasks. The game can be any type, but must meet the requirements in the rubric. They create a model and present how it works to solve the problem. Students determine how well their solution works and the social impact of their solution.

We want each student to choose his/her own specific challenge and the solution. However, if students are really stuck, you may use this example:

> Students are game designers and scientists who are working on an escape room puzzle challenge that people can do from home. Players have a set of robotic materials and download the instructions for a challenge. Players build the model and then begin the challenge. A timer will indicate the amount of time remaining. The clues allow the players to unlock the box in the time allowed – if they can figure out the steps. A minimum of 6 steps is required.

Challenge 4

Manufacturing

A novel item is being created by the company for which the students work. They will need to create an item, be able to gather and stack the items, place a stack in a given location for warehouse/storage, then take from storage to shipping.

We want each student to choose his/her own specific challenge and the solution. However, if students are really stuck, you may use this example:

> Students build models from LEGO® bricks – first they gather the materials using a robot to move materials from Raw Materials area to a Production area of the company. The students can automate production or make by hand stacks of bricks. Then, students automate the system to recognize specific products and take them to the warehouse/storage area and, as

needed, from the storage area to a shipping area. A customer order could be for a stack of red bricks or blue bricks or yellow bricks which must be produced and have all available in the warehouse. Then when an order arrives, only the correct product moves to shipping.

Challenge 5

Government and Public Administration and Law, Public Safety, Corrections & Security - Helping People in Times of Natural Disaster

What disasters occur in your area - earthquakes, tornadoes, hurricanes, large scale fires, blizzards? Students choose one type of disaster and determine how to create an alert system to help keep people out of harm's way and a system to find people that are trapped or to clean up areas affected by the disaster.

We want each student to choose his/her own specific challenge and the solution. However, if students are really stuck, you may use this example:

> A fire has been started by lightning and due to high, shifting winds and very dry conditions, the local populations are endangered. Students create a system to help keep people out of harms' way and a vehicle to find and rescue trapped individuals.

Challenge 6

Agriculture

There are lots of tasks that require manual labor in agriculture. Think about one of these tasks and create a way to automate it. Each student should choose his/her own specific challenge and the solution.

We want each student to choose his/her own specific challenge and the solution. However, if students are really stuck, you may use this example:

> After hay bales have been created, farmers take forklifts and trackers to move the bales onto trucks or to locations near barns. Could you create an autonomous vehicle to handle this job?

Review of Challenges and Discussion

Have a small class discussion on each challenge – 5 minutes per challenge – to get some ideas flowing.

Go through the rubric with students – as an overview.

- Explain there are 6 areas that will be assessed. The grading system is built around an average student earning a 2, below average performance is 1 and above average performance is a 3.
- Every student has the opportunity to get a 3 on each area, it just depends on the effort, collaboration and thoughtfulness put into the project.

Tell students there are 4 phases in the challenge. They are going to begin Phase 1 today. It will require a Teacher check and approval before students move to Phase 2. All teams work at their own pace. Time management is imperative.

Challenge Phase 1

Teacher check and approval as soon as team is ready.

- Research several topics
- Identify two or three possible problems to solve within the challenge
- Select one problem to solve within the challenge
- Brainstorm solutions for the problem chosen
- Consider human/computer interactions – impact on society – ethical; physical haptics/design of their proposed solution
- Determine the solution is doable within time frame allowed
- Write a clear statement of the problem, proposed idea for a solution, and the impact on society in their journals.
- **Teacher check and approval prior to going forward**

Challenge Phase 2

Peer review should be done as soon as a team is ready for it.

- Model plan including where using sensors and motors
- Model plan including where to use subcomponents
- Model plan including how and when to use sounds and lights
- Sketches of where to use components and overall design
- Description of solution and model concept
- Pseudocode for the solution
- **Peer review - based on rubric**
- **Teacher review**

Challenge Phase 3

Create the model and the program.

While working on solution, the teacher should ask to see:

- Previous/similar design/lessons where they have used
- Models and initial programming
- Pseudocode for possible program (determine software/hardware interactivity)
- Considerations of human/computer interactions – impact on society – ethical; physical haptics design
- Iterations of design and programing

Challenge Phase 4

Presentations & Assessment

Presentation of Completed Solution to the Class

- Present completed solution to class
- Teacher and team check against rubric
- Peer review
- Team self-assessment
- Individual self-assessment

- Clean up including all models taken apart and a full inventory of all sets

2. Explore

Begin Phase 1

Students begin Phase 1. Allow students to do 20-40 minutes of research, as needed. Have students who think they are interested in one of the challenges group together, so you have three groups. Have students brainstorm problems and write ideas they find interesting into their journals.

Ask students to choose some of the problems that were most interesting to them and brainstorm ideas. Students should write the ideas from brainstorming in their journals. Have students tell their focus group which problems and possible solutions they are most interested in. Then, have the students try to make their teams of two.

> **Note**: If you have students who don't know what they are interested in, have them visit another group or two for 10 minutes each and then choose a group to join.

Have students work in pairs or teams of 4 (maximum) on a challenge. They do not have to work with the same partner they have been, because interest (passion) should determine who works on which challenge. Students should be able to choose their partners. No one should work alone. Each set of two students should be given a SPIKE Prime set. Groups of three may need 2 sets.

Have students refer to the challenges, challenge phase 1, and the rubrics throughout the process. Suggest they keep both documents available as reference.

Have the students start the process of writing a clear statement of the problem and their first idea for a solution. They should discuss and take notes in their journals about the computer/human interaction of their solution. Discuss their ideas. Ask questions like:

- Is there bias?
- Could anyone use the solution?
- What are the limitations?
- What are the consequences?

3. Explain

Monitor students as they complete Phase 1. Have students signal you they are ready for the Teacher Check and approval. Send the teams who are ready to Phase 2.

.

Challenge Phase 2

After the solution ideas have been determined, complete a peer review of all projects. Each project should receive at least two responses from classmates and students should incorporate the feedback into their final project.

Student teams create a model plan including

- Sketches
- Optional: very basic model (not a complete one)
- Plans for including use of sensors, motors, subcomponents, sound, lights and so forth
- Pseudocode for making the model work
- Description of solution

Peer Review

Peer review should be done as soon as a team is ready for it.

Peer review requires both teams to write in their journals – feedback given and received. This is proof of helping another team with useful information and that some of the feedback received that was incorporated into the final solution. (Not all feedback must be incorporated.)

Review the following questions in order to help teams when thinking about the solution:

- Did you understand what the solution is supposed to do? **(Clarify)**
- What features does the solution have? Does the project seem as if it will work as expected? **(Features)**
- How engaging is the solution? Is it interactive, original, fun, interesting? Who would use it/interact with it? **(Appeal)**

Ask students to verify the model plan includes:

- Use of sensors and motors
- Use of subcomponents
- Use of sounds and lights
- Sketches of where to use components and overall design
- A description of the solution and model concept
- Pseudocode for the solution

You may wish to help students with giving feedback by sharing these questions:

- What is something that is missing, doesn't seem to work, or could be improved?
- What is something that is confusing or could be done differently?
- What is something that works well or you really like about the solution?

Students should write feedback received into their journals as well as take notes in their journals on feedback they gave.

4. Elaborate

Students complete Phase 3.

Students incorporate the feedback into their plans. They complete the model and the programming. Students iterate until the model and the program are complete and debugged. They should write notes in their journals about the iterative process and how they improved their models and their program.

Students complete writing the solution. They determine how they will present their ideas and what part each person will have in the presentation.

Students should practice the presentation to make it as professional as possible, remembering the lessons learned throughout the course. Students should refer to the rubrics to ensure they have included all the required aspects to their solution and the presentation.

While working on solution, the teacher should ask to see:

- Previous/similar design/lessons where they have used
- Models and initial programming
- Pseudocode for possible program (determine software/hardware interactivity)
- Considerations of human/computer interactions – impact on society – ethical; physical haptics design
- Iterations of design and programing

Present

Students begin Phase 4.

Challenge Phase 4

Presentation of Completed Solution to the Class

Students present their challenge, solution and working model in professional manner to the class and any guests. All team members should be part of the presentation. Students should refer to the rubrics before the first presentation.

Prior to the presentations, ask students questions like:

- What will you be looking for in the presentations?
- What will be the focus of your presentation?
- What must be included in a presentation?

Students may wish to have a copy of the rubric before, during, or after each presentation. Encourage students to take notes in their journals during or immediately after a presentation.

Allow students time to look at the peer review and write notes after each presentation. If permitted, students may take pictures of the models if that will assist in the review.

5. Evaluate

Each presentation will be evaluated by the teacher, peers, team, and individual.

- Teacher and team check against rubric
- Peer review
- Team self-assessment
- Individual self-assessment

Peer Review

After the solution ideas have been determined, students should have a peer review of all projects. Each project should receive at least two responses from classmates and incorporate the feedback into their final project.

You may wish to ask students the following questions to help with feedback:

- Was something missing, did not seem to work, or could have been improved?
- Was something confusing or could have be done differently for clarification?
- What worked well or made the idea memorable or was something you really liked about the solution?

Review the following questions in order to help teams when thinking about the solution:

- Did you understand what the solution is supposed to do? **(Clarify)**
- What features did the solution have? Did the project work as expected? **(Features)**
- How engaging was the solution? Was it interactive, original, fun, interesting? Who would use it/interact with it? **(Appeal)**

Team Assessment

Students should rate their team effectiveness.

3 – Our team worked like a well together. We gave and accepted graciously feedback from each other and from other teams. We made sure each person played a significant role in solving the problem and creating the solution.
2 – Our team worked fairly well together. We had a bit trouble now and then giving and receiving feedback from each other and from other teams. We generally made sure people were involved or engaged.
1 – Our team struggled. We had a hard time giving and receiving feedback, and the work was not shared equally. Some member(s) monopolized the time together and insisted their ideas were best.

Self-Assessment

Each student should rate his/her own behaviors during the challenge.

3 – I gave and accepted graciously feedback to my partner and other teams. I allowed my partner to have time to build, program, explain, and document.
2 – I gave feedback graciously to others but had a bit of difficulty accepting feedback from others. I didn't always give my partner equal time with building, programming, explaining or documenting.
1 – I gave little feedback to others; I accepted little feedback from others.

Inventory

After all the assessments are complete, students should take apart the models and put the elements into the correct compartments. Each team should take a **complete inventory** of the set. Have students inform you of any missing pieces. There should be a location for students to take extra items they find in their sets which may be used to complete other sets.

Culminating Project Rubric

Both the teacher and the team should complete the rubric, assessing the solution, and presentation.

Segment	**3**	**2**	**1**
Developed a Plan and Diligently Worked Toward a Successful Solution. (Fostering an Inclusive Computing Culture)	Created a well-documented plan and the team worked effectively toward a solution. Documentation contains all the following: initial ideas from brainstorming, chosen topic, notes from research, initial idea for model and how it will work, daily notes on progress, issues that need to be resolved during process.	Created a plan and worked toward the solution. Documentation contains most of the following: initial ideas from brainstorming, chosen topic, notes from research, initial idea for model and how it will work, daily notes on progress, issues that need to be resolved during process.	Did not have a complete plan and team worked occasionally toward a solution. Documentation contains a little of the following: initial ideas from brainstorming, chosen topic, notes from research, initial idea for model and how it will work, daily notes on progress, issues that need to be resolved during process.
Programming (Testing and Refining Computational Artifacts)			
Includes purposeful use of lights.	Uses lights in an original way	Uses lights directly as done in a lesson.	Does not use lights.
Includes purposeful use of sounds.	Uses sound in an original way.	Uses sound directly as done in a lesson.	Does not use sound.

Written efficiently Includes subcomponents and My Blocks	Uses several subcomponents and My Blocks	Uses one My Block as subcomponent	Does not use subcomponents or My Blocks
Uses sequences and loops	Uses multiple sequences and nested loops	Uses a sequence and a loop	Does not use a loop
Uses conditional statements	Uses multiple conditional statements	Uses one conditional statement	Does not use a conditional statement
Can justify the appropriate tools and techniques	Clear explanation of why programming was chosen and used	Explanation is satisfactory, but needs refinement.	Explanation is difficult to follow and shows a lack of understanding.
Can explain the program and its functions	Clear explanation of how programming was chosen and used	Explanation is satisfactory, but needs refinement.	Explanation is difficult to follow and shows a lack of understanding.
Engineering Design (Testing and Refining Computational Artifacts)			
Model is fully functional with purposeful use of motors	Purposeful use of motors	Motor used	No motor used
Model is fully functional with purposeful use of sensors	Purposeful use of multiple sensors	Sensor used	No sensor used

Can explain how the team iterated possible solutions as they used the design process	Clear explanation of how the model idea was chosen and the iterations needed to get to the final version	Some explanation of how the model idea was chosen, but only a few examples of iterations between initial idea and final version	No clear explanation of how model idea was chosen and no iteration notes between idea and final model
Can explain how feedback was incorporated into the design	Clear explanation of how final model was tested with feedback noted and how feedback was used to improve	Some explanation of how final model was tested with feedback notes, but vague information on how feedback was incorporated	Little to no explanation of feedback received or how it was incorporated
Documentation (Creating Computational Artifacts)			
Followed the process for solving a problem	Identified the problem; brainstormed several possible solutions; tested models; iterated model design and programming; analyzed the designs and programming; reviewed feedback; chose the best solution based on criteria	Identified the problem, thought of a solution, built a model and made changes, programmed and debugged the solution	Identified a problem, made and programmed a model

Created computational artifacts	Chose appropriate tools; decomposed the problem; coded with the design in mind; debugged the program; improved on the original design	Choose appropriate tools, built and coded the model, debugged the program, made the model work	Made a model and programmed it
Selected a variety of formats to communicate ideas	Used screen shots, video, photos, journaling, and models to communicate throughout the project	Used photos and journaling to communicate during but mostly at the end of the project	Made some journal entries and took a few photos at the end of the project
Collaboration and Feedback (Collaborating Around Computing)			
Used the feedback form for sharing ideas with at least 2 other groups	Useful feedback, both positive and inspiring were provided to other teams	Feedback was given, but minimal	No feedback given
Gave an example of useful feedback shared with another team	Feedback shows thought and was helpful to another team and could have led to improvement.	Feedback was given but was minimal.	No feedback given
Completed the B/Y/V feedback form for self-assessment	Feedback was honest and thoughtful, providing both positive comments and areas where improvement could be made.	Feedback was minimal with little thought being made for what could be done in the future and what was done well.	No feedback given

Completed the B/Y/V feedback form for team assessment	Feedback was honest and thoughtful, providing both positive comments and areas where improvement could be made.	Feedback was minimal with little thought being made for what could be done in the future and what was done well.	No feedback given
Presentation (Communicating About Computing, Collaborating Around Computing, Recognizing and Defining Computational Problems)			
All team members participated (Collaborating Around Computing)	All members participated fully.	All or most participated, but dominated by one person.	Only one person presented or completed a majority of the work or no presentation occurred.
All team members answered questions (Collaborating Around Computing)	All members participated fully.	All or most participated, but dominated by one person.	Only one person presented or no presentation occurred.
The problem was clearly identified (Recognizing and Defining Computational Problems)	The problem was clearly identified and explanation of how the problem was determined through brainstorming and narrowing of options.	The problem was identified.	The problem was never clearly identified.

The solution solved the problem (Collaborating Around Computing)	The solution solved the problem in an original manner through a process of iteration that was explained.	The final solution solved the problem.	The solution did not solve the problem.
The physical model was programmed and demonstrated as part of the solution	The model demonstrated the solution and an explanation of how each part of the solution, including the model and the programming, work together to create the final solution.	The model demonstrated the solution.	The model did not demonstrate a solution.
The audience was taken into consideration when presenting	The audience was taken into consideration and explanations were clear and thoughtful, giving necessary background, processes, and explanations.	The audience was generally taken into consideration, but some aspects such as adequate background were missing.	The audience was tangential to the presentation. Clear explanations for someone not familiar with the issue were missing.