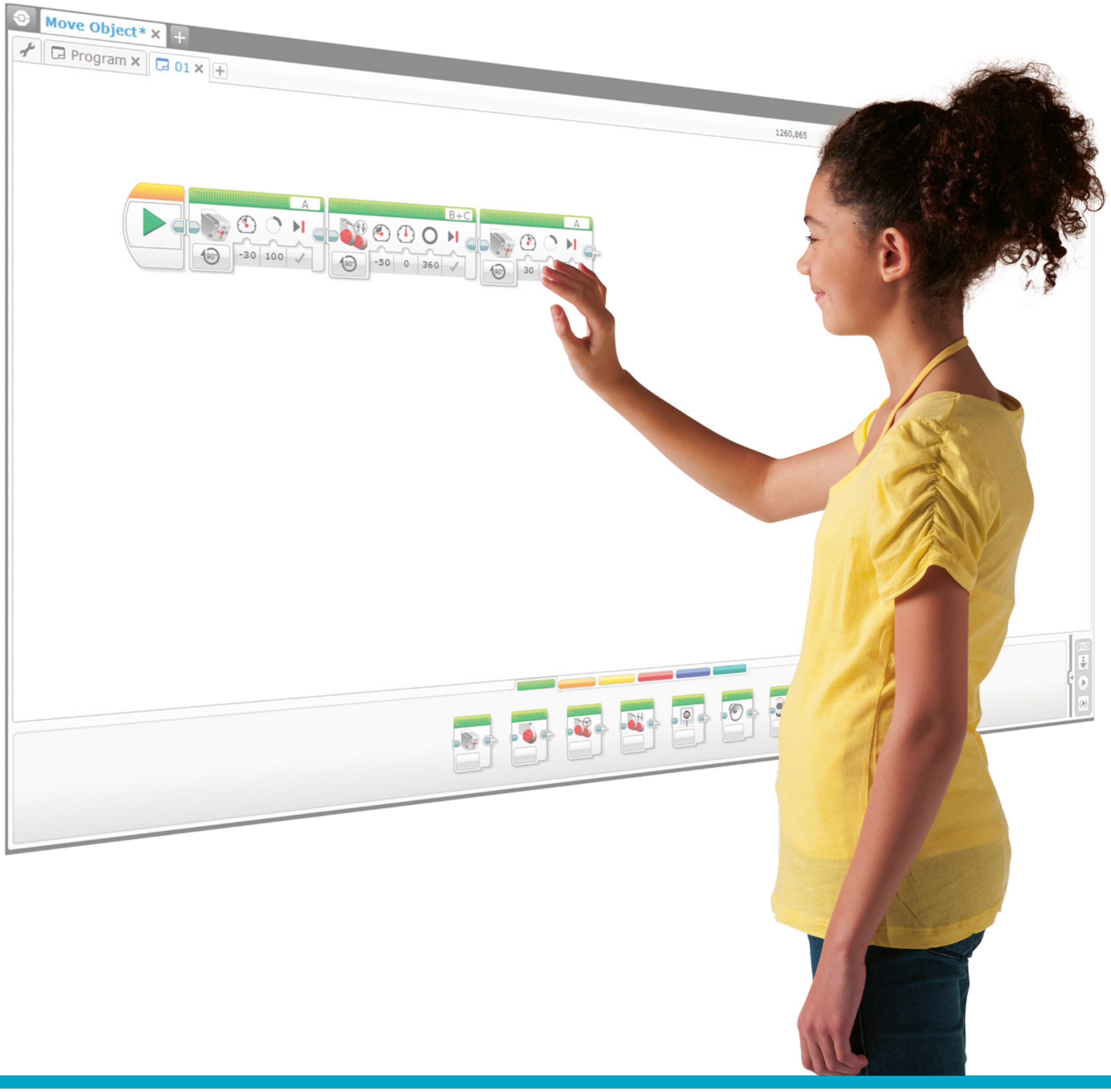
Curriculum Links

In the US and many other countries, education and industry leaders are calling for the inclusion of more programming (coding) experiences in K-12 classrooms. The Computer Science Teachers Association (CSTA) established a set of Computer Science Standards in 2011 and revisited them in 2016.

With this in mind, LEGO® Education has produced this set of lessons using LEGO MINDSTORMS Education EV3 to help students in grades 5-8 tackle this abstract subject.



Lesson	Learning Objectives	Programming concept	Major EV3 Programming Blocks Covered
Autonomous Parking	– Understand that algorithms are capable of carrying out a series of instructions – Explore the concept of Outputs, comparing different ways in which a wheeled robot can move	– Design cars that can park themselves safely without driver intervention	– Move Tank – Wait – Touch Sensor – Brick Status Light
Reversing Safely	– Extend the use of algorithms are capable of carrying out a series of instructions – Extend understanding of outputs	– Design features for a car that will improve safety as it reverses	– Move Steering – Wait – Brick Button – Sound – Display
Automatic Headlights	– Explore the concept of Inputs and the way to control them – Explore the concept of a Wait for function	– Design car features that will improve nighttime driving safety	– Wait – Color Sensor – Loop – Loop Interrupt
Line Detection	– Explore the concept of the Loop – Understand the concept of a switch and how to use it for true and false operations	– Design ways to improve driving safety by helping to prevent drivers from falling asleep and causing an accident	– Wait – Color Sensor – Loop – Switch – Loop Interrupt
Object Detection	– Extend understanding of the Loop	– Design ways to avoid accidents between vehicles and objects in the road	– Wait – Ultrasonic Sensor – Loop – Switch
Unlocking a Car	– Understand simple Boolean logic (such as AND, OR and NOT) and some of its uses in circuits and programming – Use several inputs in combination	– Design a way to use passcodes to protect cars from thieves	– Ultrasonic Sensor – Brick Buttons – Logic – Switch – Loop
Cruise Control	– Use the Variable Block to store information – Develop multi-level programs – Create function blocks (My Blocks)	– Design a cruise control program to assist drivers by making their driving experience less stressful	– Touch Sensor – Loop – Switch – Variable – Math – My Blocks
Roaming Vehicles	– Make appropriate use of data structures such as lists, tables and arrays – Extend Boolean logic and some of its uses in circuits and programming – Use the Variable Block to store information – Use the Array Operations Block	– Design an autonomous car that is safe enough to drive on the streets	– Variable – Brick Buttons – Loop – Array Operations – My Blocks
Project	– Integrate and reinvest what they have learned in the previous coding lessons – Design, use, and evaluate solutions to a real-world problems and physical systems	– Design an autonomous car that can safely cross an intersection	– All of the above

Curriculum Grid

NGSS ● = addresses standard ◐ = partially addresses standard		Autonomous Parking	Reversing Safely	Automatic Headlights	Color and Line Detection	Object Detection	Unlocking a Car	Cruise Control	Roaming Vehicles
MS.Engineering Design									
MS-ETS1-1.	Define the criteria and constraints of a design problem, asking questions and defining problems with sufficient precision to ensure a successful solution, taking into account relevant scientific principles and potential impacts on people and the natural environment that may limit possible solutions	●	●	●	●	●	●	●	●
MS-ETS1-2.	Evaluate competing design solutions using a systematic process to determine how well they meet the criteria and constraints of the problem	●	●	●	●	●	●	●	●
MS-ETS1-3.	Analyze data from tests to determine similarities and differences among several design solutions to identify the best characteristics of each that can be combined into a new solution to better meet the criteria for success	●	●	●	●	●	●	●	●
MS-ETS1-4.	Develop a model to generate data for iterative testing and modification of a proposed object, tool, or process such that an optimal design can be achieved			●	●	●	●	●	●

Science and Engineering Practices in the NGSS									
Practice 1	Asking questions (for science) and defining problems (for engineering)	●	●	●	●	●	●	●	●
Practice 2	Developing and using models	●	●	●	●	●	●	●	●
Practice 3	Planning and carrying out investigations								
Practice 4	Analyzing and interpreting data								
Practice 5	Using mathematics and computational thinking	●	●	●	●	●	●	●	●
Practice 6	Constructing explanations (for science) and designing solutions (for engineering)	●	●	●	●	●	●	●	●
Practice 7	Engaging in argument from evidence								
Practice 8	Obtaining, evaluating, and communicating information	●	●	●	●	●	●	●	●

Curriculum Grid

CSTA Standards (2016) ● = addresses standard ◐ = partially addresses standard		Autonomous Parking	Reversing Safely	Automatic Headlights	Color and Line Detection	Object Detection	Unlocking a Car	Cruise Control	Roaming Vehicles
Algorithms and Programming									
2-A-2-1	Solicit and integrate peer feedback as appropriate to develop or refine a program	●	●	●	●	●	●	●	●
2-A-7-2	Compare different algorithms that may be used to solve the same problem in terms of their speed, clarity, and size (e.g., different algorithms solve the same problem, but one might be faster than the other). (Clarification: students are not expected to quantify these differences)	●	●	●	●	●	●	●	●
2-A-7-3	Provide proper attribution when code is borrowed or built upon	●	●	●	●	●	●	●	●
2-A-7-4	Interpret the flow of execution of algorithms and predict their outcomes. (Clarification: algorithms can be expressed using natural language, flow and control diagrams, comments within code, and pseudocode)	●	●	●	●	●	●	●	●
2-A-5-5	Design, develop, and present computational artifacts such as mobile applications that address social problems both independently and collaboratively	●	●	●	●	●	●	●	●
2-A-5-6	Develop programs, both independently and collaboratively, that include sequences with nested loops and multiple branches. (Clarification: at this level, students may use block-based and/or text-based programming languages)			●	●	●	●	●	●
2-A-5-7	Create variables that represent different types of data, and manipulate their values						●	●	●
2-A-4-8	Define and use procedures that hide the complexity of a task and can be reused to solve similar tasks. (Clarification: students use and modify, but do not necessarily create, procedures with parameters)								●
2-A-3-9	Decompose a problem into parts and create solutions for each part	●	●	●	●	●	●	●	●
2-A-6-10	Use an iterative design process (e.g., define the problem, generate ideas, build, test, and improve solutions) to solve problems, both independently and collaboratively	●	●	●	●	●	●	●	●
Computing Systems									
2-C-7-11	Justify the hardware and software chosen to accomplish a task (e.g., comparison of the features of a tablet vs. desktop, selecting which sensors and platform to use in building a robot or developing a mobile app)								
2-C-4-12	Analyze the relationship between a device's computational components and its capabilities. (Clarification: computing systems include not only computers, but also cars, microwaves, smartphones, traffic lights, and flash drives)	●	●	●	●	●	●	●	●
2-C-6-13	Use a systematic process to identify the source of a problem within individual and connected devices (e.g., follow a troubleshooting flow diagram, make changes to software to see if hardware will work, restart device, check connections, swap in working components)	◐	◐	◐	◐	◐	◐	◐	◐

Curriculum Grid

CSTA Standards (2016) ⬢ = addresses standard ◐ = partially addresses standard		Autonomous Parking	Reversing Safely	Automatic Headlights	Color and Line Detection	Object Detection	Unlocking a Car	Cruise Control	Roaming Vehicles
Data and Analysis									
2-D-7-14	Describe how different formats of stored data represent tradeoffs between quality and size. (Clarification: compare examples of music, text, and/or image formats)								
2-D-7-15	Explain the processes used to collect, transform, and analyze data to solve a problem using computational tools (e.g., use an app or spreadsheet form to collect data, decide which data to use or ignore, and choose a visualization method)	◐	◐	◐	◐	◐	◐	◐	◐
2-D-5-16	Revise computational models to more accurately reflect real-world systems (e.g., ecosystems, epidemics, spread of ideas)	⬢	⬢	⬢	⬢	⬢	⬢	⬢	⬢
2-D-4-17	Represent data using different encoding schemes (e.g., binary, Unicode, Morse code, shorthand, student-created codes)								
Impacts of Computing									
2-I-7-18	Summarize negative and positive impacts of using data and information to categorize people, predict behavior, and make recommendations based on those predictions (e.g., customizing search results or targeted advertising based on previous browsing history can save search time and limit options at the same time)								
2-I-7-19	Explain how computer science fosters innovation and enhances nearly all careers and disciplines	◐	◐	◐	◐	◐	◐	◐	◐
2-I-1-20	Provide examples of how computational artifacts and devices impact health and wellbeing, both positively and negatively	⬢	⬢	⬢	⬢	⬢	⬢	⬢	⬢
2-I-1-21	Describe ways in which the Internet impacts global communication and collaborating								
2-I-1-22	Describe ethical issues that relate to computing devices and networks (e.g., equity of access, security, hacking, intellectual property, copyright, Creative Commons licensing, and plagiarism)								
2-I-6-23	Redesign a computational artifact to remove barriers to universal access (e.g., using captions on images, high contrast colors, and/or larger font sizes)								◐
Networks and the Internet									
2-N-7-24	Summarize security risks associated with weak passwords, lack of encryption, insecure transactions, and persistence of data						◐		
2-N-7-25	Simulate how information is transmitted as packets through multiple devices over the Internet and networks						◐		