

1. Multilevel Programming (Cruise Control)

Content Area: **Technology**
Course(s):
Time Period: **Cycle**
Length: **8 days**
Status: **Published**

General Overview, Course Description or Course Philosophy

This 22-day cycle course for 8th grade students serves as a continuation of the robotics curriculum started in 7th grade. Through problem solving activities and project based collaboration, students will use experimentation, testing, and analysis to develop foundational skills in the areas of algorithm construction, scripting, engineering design, and prototype testing. The course culminates in students designing and completing their own robot using knowledge that they've acquired over the marking period, with the goal of solving a given challenge.

OBJECTIVES, ESSENTIAL QUESTIONS, ENDURING UNDERSTANDINGS

Students will be able to design a cruise control program to assist drivers by making their driving experience less stressful.

CONTENT AREA STANDARDS

SCI.MS-ETS1-1	Define the criteria and constraints of a design problem 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.
SCI.MS-ETS1-2	Evaluate competing design solutions using a systematic process to determine how well they meet the criteria and constraints of the problem.
SCI.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.
SCI.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.
CS.9-12.8.1.12.AP.1	Design algorithms to solve computational problems using a combination of original and existing algorithms.
CS.9-12.8.1.12.AP.2	Create generalized computational solutions using collections instead of repeatedly using simple variables.
CS.9-12.8.1.12.AP.3	Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
CS.9-12.8.1.12.AP.4	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue.
CS.9-12.8.1.12.AP.5	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

CS.9-12.8.1.12.AP.6	Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
CS.9-12.8.1.12.AP.7	Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback from users.
CS.9-12.8.1.12.AP.8	Evaluate and refine computational artifacts to make them more usable and accessible.
CS.9-12.8.1.12.AP.9	Collaboratively document and present design decisions in the development of complex programs.

RELATED STANDARDS (Technology, 21st Century Life & Careers, ELA Companion Standards are Required)

LA.RST.6-8.3	Follow precisely a multistep procedure when carrying out experiments, taking measurements, or performing technical tasks.
LA.RST.6-8.4	Determine the meaning of symbols, key terms, and other domain-specific words and phrases as they are used in a specific scientific or technical context relevant to grades 6-8 texts and topics.
LA.RST.6-8.7	Integrate quantitative or technical information expressed in words in a text with a version of that information expressed visually (e.g., in a flowchart, diagram, model, graph, or table).
LA.WHST.6-8.2	Write informative/explanatory texts, including the narration of historical events, scientific procedures/experiments, or technical processes.

STUDENT LEARNING TARGETS

Refer to the 'Declarative Knowledge' and 'Procedural Knowledge' sections.

Declarative Knowledge

Students will understand that:

- The use of algorithms and programming are used to solve problems in our local community and society.
- Algorithmic thinking is the ability to create an ordered series of steps with the purpose of solving a problem.
- Evaluating and debugging the ability to verify whether or not a prototype works as intended, and if not, the ability to identify what needs to be improved. It is also the process a computer programmer goes through in order to find and correct mistakes within a program.

Procedural Knowledge

Students will be able to:

- Use the Variable Block to store information
- Develop multi-level programs
- Create function blocks (My Blocks)

EVIDENCE OF LEARNING

Refer to the 'Formative Assessments' and 'Summative Assessments' sections.

Formative Assessments

Daily checklist

Project checklist

Summative Assessments

Performance of challenge task

Challenge checklist

RESOURCES (Instructional, Supplemental, Intervention Materials)

Mindstorms EV3 Programming app (on computer)

– Touch Sensor

INTERDISCIPLINARY CONNECTIONS

Mathematics

Communication

Science, technology, Engineering, and Math

ACCOMMODATIONS & MODIFICATIONS FOR SUBGROUPS

See link to Accommodations & Modifications document in course folder.