MILLSTONE TOWNSHIP SCHOOL DISTRICT
ELECTIVE CURRICULUM
GRADE: 7
(Updated October 2021)

Unit of Study: Programming I

Unit Overview:
8.2 Technology Education, Engineering, Design, and Computational Thinking - Programming:
- All students will develop an understanding of the nature and impact of technology, engineering, technological design, computational thinking and the designed world as they relate to the individual, global society, and the environment.
- Computational thinking builds and enhances problem solving, allowing students to move beyond using knowledge to creating knowledge.

Pacing

1 Marking Period - Approximated 45 days

Enduring Understandings:	Essential Questions:
- Computer programming is largely concerned with the analysis and manipulation of data sets through the use of computational algorithms. - There is an infinite variety of data and algorithms. - There is more than one way to solve a problem. - A well-designed program, will be easier to code, demonstrate fewer bugs, and require less maintenance in the future. - Well written code is efficient, easy for a future user to understand, and re-usable.	- What is an algorithm? - What is a program? - What are a function and variable? - What are a loop and a nested loop? - How many ways can you solve a problem? - What skills does a computer programmer need to be successful in the real world? - What steps can help you to fix or debug a program or solve a problem?
Objectives/Teaching Points:	**2020 NJSLS:**

- Engage in problem solving using code and following the rules of programming.
- Utilize both functions and variables when writing code in a program.
- Explain and define vocabulary and key concepts needed to write code: "algorithm", "program," "loop", "nested loops", and "debugging".
- Work together using "crowdsourcing" to solve and design a program.
- Explore creative labs to develop unique programs/ games/songs using code.
- Advance through progressively complex levels of mazes using various skills learned in this course.
- Create programs that utilize the skills, functions, and variables covered in this course.
- Explore HTML and CSS coding tags to design a single web page.
- Utilize HTML codes to create a new web page.
- Utilize CSS codes to enhance style of a web page.

- 8.1.8.CS.2: Design a system that combines hardware and software components to process data.
- 8.1.8.AP.3: Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
- 8.1.8.AP.4: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
- 8.1.8.AP.5: Create procedures with parameters to organize code and make it easier to reuse.
- 8.2.8.ED.2: Identify the steps in the design process that could be used to solve a problem
- 8.1.8.AP.8: Systematically test and refine programs using a range of test cases and users.
- 8.1.8.AP.9: Document programs in order to make them easier to follow, test, and debug.

ISTE Standards (formerly NETS)

1.b - Create original works as a means of personal or group expression.
1.c - Use models and simulation to explore complex systems and issues.
2.d - Contribute to project teams to solve problems.
4.b - Plan and manage activities to develop a solution or complete a project.
4.d - Use multiple processes and diverse perspectives to explore alternative solutions.

CSTA K-12 Computer Science Standards

- CPP.L1:6-05. Construct a program as a set of step-by-step instructions to be acted out.
- CT.L1:3-03 - Understand how to arrange information into useful order without using a computer.

	- CT.L1:6-01 - Understand and use the basic steps in algorithmic problem-solving. - CT.L1:6-02 - Develop a simple understanding of an algorithm using computer-free exercises. - CT.L2-07 - Represent data in a variety of ways: text, sounds, pictures, & numbers. **Common Core Mathematical Practices** 1. Make sense of problems and persevere in solving them. 2. Reason abstractly and quantitatively. 3. Construct viable arguments and critique the reasoning of others. 6. Attend to precision. 7. Look for and make use of structure. 8. Look for and express regularity in repeated reasoning.

2020 NJSLS Career Readiness, Life Literacies, & Key Skills:

CRP1- Act as a responsible and contributing citizen and employee.
CRP2- Apply appropriate academic and technical skills.
CRP4- Communicate clearly and effectively and with reason.
CRP5- Consider the environmental, social and economic impacts of decisions.
CRP6- Demonstrate creativity and innovation.
CRP7- Employ valid and reliable research strategies.
CRP8- Utilize critical thinking to make sense of problems and persevere in solving them.
CRP9- Model integrity, ethical leadership and effective management.
CRP11- Use technology to enhance productivity.
CRP12- Work productively in teams while using cultural global competence.
9.4.8.CI.1 Assess data gathered on varying perspectives on causes of climate change (e.g., cross-cultural, gender-specific, generational), and determine how the data can best be used to design multiple potential solutions (RI.7.9, 6.DP.N.5, 7.1.NH.IPERS.6, 8.2.8.ETW.4).
9.4.8.CI.2 Repurpose an existing resource in an innovative way (e.g. 8.2.8.NT.3).
9.4.8.CI.3 Examine challenges that may exist in the adoption of new ideas (e.g., 2.1.8.SSH, 6.1.8.CivicsPD.2)
9.4.8.CI.4 Explore the role of creativity and innovation in career pathways and industries.
9.4.8.CT.1 Evaluate diverse solutions proposed by a variety of individuals, organizations, and/or agencies to a local or global problem, such as climate change, and use critical thinking skills to predict which one(s) are likely to be effective (e.g., MS-ETS1-2).
9.4.8.TL.2 Gather data and digitally represent information to communicate a real-world problem (e.g., MS-ESS3-4, 6.1.8.EconET.1, 6.1.8.CivicsPR.4).

9.4.8.TL.3 Select appropriate tools to organize and present information digitally.

Interdisciplinary Connections:

Language Arts: Students will read complex information text in order to follow directions for assignments.
NJSLSA.R10. Read and comprehend complex literary and informational texts independently and proficiently with scaffolding as needed.

VPA-Media Arts:
- 1.2.8.Cr1a: Generate a variety of ideas, goals, and solutions for media artworks using creative processes such as sketching, brainstorming, improvising, and prototyping with increased proficiency, divergent thinking, and opportunity for student choice.
- 1.2.8.Cr1b: Organize and design artistic ideas for media arts productions
- 1.2.8.Cr.1c: Critique plans, prototypes, and production processes considering purposeful and expressive intent.
- 1.2.8.Cr2a: Organize and design artistic ideas for media arts productions.
- 1.2.8.Cr2b: Critique plans, prototypes, and production processes considering purposeful and expressive intent.
- 1.2.8.Cr3a: Experiment with and implement multiple approaches that integrate content and stylistic conventions.
- 1.2.8.Pr4a: Experiment with and integrate multiple forms, approaches and content to coordinate, produce, and implement media artworks that convey purpose and meaning (ex: narratives, video games, interdisciplinary projects, multimedia theatre).
- 1.2.8.Pr5a: Develop and demonstrate a variety of artistic, design, technical, and soft skills (ex: Self initiative, problem solving, collaborative communication) through performing various roles in producing media artworks
- 1.2.8.Pr5b: Develop and demonstrate creativity and adaptability through processes such as testing constraints and divergent solutions, within and through media arts productions
- 1.2.8.Pr5c: Develop and demonstrate creativity and adaptability in standard and experimental ways, to construct, achieve assigned purpose, and communicate intent in media artworks.
- 1.2.8.Pr6a: Analyze and design various presentation formats and tasks in the presentation and/or distribution of media artworks.
- 1.2.8.Pr6b: Analyze benefits and impacts from presenting media artworks.
- 1.2.8.Re9a: Evaluate media art works and production processes at decisive stages, using identified criteria, and considering context and artistic goals.

Learning Experiences:	Assessments:
The following learning experiences will	**Formative:** - Teacher observation - Exit slips - Checklists - Student self-assessment

help students explore the big ideas and essential questions:

- Complete activities to introduce key vocabulary terms.
- Advance through mazes and draw an image using artist tools based on skills learned from code.org.
- Use Scratch software to design personal game or animation.
- Work with and explore how to use Arduino boards that read inputs -such as a light on a sensor, a finger on a button, and turn it into an output - activating a motor, or turning on an LED.
- Use programming and Arduino boards to do something unique.
- Use HTML codes and CSS style codes to design a web page based on a personal theme/topic.

Summative:
  Skills for Assessment
- Grading Rubrics for Scratch Game and Arduino boards (dollhouse challenge).
- Assessment worksheets from code.org based on understanding of vocabulary and coding.
- Assessments provided use code.org for online understanding of programming skills.
- Grading Rubric for web page using HTML and CSS coding.

Alternative:
  Other Evidence and Student Self-Assessment
- Follows directions, safety concerns, and classroom procedures
- Demonstrates creativity within code writing.
- Experiments with a variety of tools available in Scratch and Arduinos.
- Seeks to explore options teacher not required/demonstrated to enhance overall project.
- Explore HTML and CSS codes not shown in Khan Academy for further understanding.

Benchmark:
- Typing skills monthly assessment

Ideas for Differentiation

- Varying coding software.
- Small group instruction to review skills taught based on students abilities.
- Pairing up students - based on learning needs with a stronger "coder".
- Talking it through out loud the programming steps with a partner to solve more difficult programming problems.

- Debugging by running it through and make changes to see what happens when having a problem with comprehension.
- List of what to do if you need help and get stuck in a program.
- Choosing topics of interest for when designing a web page.
- Use Khan Academy HTML and CSS programming courses at students' pacing.
- Peer teaching coding skills when needed.

Based on the needs of the students, there may be a need for additional teaching points, extending beyond or substituting in for those outlined in the curriculum map.

English Language Learners:
- Speak and display terminology and movement
- Teacher modeling
- Peer modeling
- Develop and post routines
- Word walls

IEP/504 Learners:
- Utilize modifications and accommodations delineated in the student's IEP
- Work with paraprofessional
- Use multi-sensory teaching approaches. Provide helpful visual, auditory, and tactile reinforcement of ideas.
- Work with a partner
- Provide concrete examples and relate all new strategies to previously learned strategies.
- Solidify and refine concepts through repetition.
- Change requirements to reduce activity time
- Chunk tasks into sections to assist with organization and work completion
- Provide graphic organizers and sentence starters as needed

Students at Risk of Failure:
- Using visual demonstrations, illustrations, and models
- Give directions/instructions verbally and in simple written format.
- Chunk tasks into sections to make workload less overwhelming
- Peer Support
- Increase one on one time
- Teachers may modify instructions by modeling what the student is expected to do
- Instructions left on the board/easel for the student to see during the time of the lesson.
- Review behavior expectations and make adjustments for personal space or other behaviors as needed.
- Oral prompts can be given

Gifted and Talented Learners:
- Curriculum compacting
- Inquiry-based instruction
- Independent study
- Higher order thinking skills
- Adjusting the pace of lessons
- Interest based content

- Real world scenarios
- Student Driven Instruction
- Student choice selecting application(s) needed to complete tasks

Suggested Resources:

Student Materials:
Technology:
- Google Classroom and Documents
- Internet
- Code.org web site and classroom
- Scratch software / online
- Arduino boards and input devices
- Arduino software/ online
- Khan Academy - HTML and CSS basics

Teaching Materials:

Worksheets: teacher made and from code.org and Khan Academy

Arduino boards and sensors

Google Classroom – Documents

Teacher Resources: