Montague Township School Curriculum Guide Coding and HTML- STEM Course (Middle School: Mixed Grade Levels)

Course Description: Coding and HTML is a choice Middle School elective designed to introduce STEM students to understanding and writing code. Students will work with a variety of contemporary peripheral devices. Students will develop a working proficiency in the language of code and HTML. This course has been designed to address problem solving through hands-on exploration of design ideas and solutions. It will incorporate group learning, authentic assessment methods, and the use of technology as a means of learning throughout the course. Real world problem solving will be an important part of the Coding and HTML curriculum. The computer and its inherent peripherals will be used in the final production of all projects.

Rationale: This trimester course will provide students with multiple opportunities to expand their critical thinking and problem solving abilities. The course will also introduce new subject matters and possible future careers in the various STEM-related fields. The course incorporates basic design concepts in the application of problem-solving and original creation of code. This course is aligned with the New Jersey Student Learning Standards for Science, New Jersey Career Readiness, Life Literacies, and Key Skills and the New Jersey Student Learning Standards for Technological Literacy (NJSLS for Technological Literacy).

Format/Mapping/Sequence: The format in which the curriculum is written follows the parameters of Understanding by Design. Each course curriculum document is written as a series of units containing established goals, enduring understandings, essential questions, and the necessary skills and knowledge a student must attain in a school year. In addition, each document stipulates resources, required tasks, and assessments. Teachers are expected to design lessons that will meet the requirements stipulated in this document; however, they are provided flexibility in how they choose to meet these demands. As well, the order of the units is a suggestion and a teacher may introduce the units as he/she feels best meets the needs of the class, keeping in mind the scaffolding of skill development suggested.

Pacing: Coding and HTML is a choice elective course that meets every other day over the course of a trimester, 40-minutes per session, and is currently open to students in grades 5-8. There are 2 student-centered units, each designated to focus on a specific program or skill. Each of the units provides a suggested time frame, taking into consideration the time needed to differentiate for a variety of learners.

Resources: In each unit, both electronic and print resources are provided. It is the intention that teachers will be able to access the curriculum document on the district website as well as be able to add to the resources lists periodically throughout the school year. A valuable site that should be referenced in planning <u>https://www.nj.gov/education/standards/</u>

Student Outcomes: (Link to New Jersey Student Learning Standards)

In accordance with district policy as mandated by the New Jersey Administrative Code and the New Student Learning Standards, the following are proficiencies required for the successful completion of the above named course.

Interdisciplinary Connections

Social Studies:

Use a variety of sources to analyze and explain events as related to class discussions, readings and research papers.

Analyze the extent to which specific time periods affect and influence culture, society, and civilizations as it pertains to specific novels, texts, and writing assignments. Current events

Visual and Performing Arts

Create projects that include well-structured plots and subplots, clear thematic intent, original characters, and elements appropriate to a variety of genres.

English Language Arts

Critique writing, Open ended writing responses, Current art events, Guest speakers Career options are discussed as they arise throughout the course

Technology

Students learn how to research, acquire, and present information using acceptable and appropriate internet etiquette standards when using a variety of websites, databases, Google apps and extension for various writing, research, and multimedia presentations.

Career Ready Practices

Communicate effectively and effectively and with reason

Apply appropriate academic and technical skills

Demonstrate creativity and innovation

Use technology to enhance productivity

Work productively in teams while using cultural global competence

Employ valid and reliable research strategies

Applicable career skills will be the focus within each unit and students will apply these skills in their class discussions, readings, research and presentations.

Career options are discussed as they arise throughout the course

Unit of Study: Coding Module 1		
Grade Level: Middle School Mixed Grade Levels	Suggested Timeframe: 1-2 weeks	
Primary Performance Expectations/Standards: <u>Algorithms and Programming</u> 8.1.2.AP.1: Model daily processes by creating and following algorithms to complete tasks.		
8.1.2.AP.5: Create programs with sequences and simple loops to accomption tasks. 8.1.2.AP.4: Break down a task into a sequence of steps. 8.1.2.AP.5: Describe a program's sequence of events, goals, and expected outcomes.		
8.1.2.AP.6: Debug errors in an algorithm or program that includes sequences and simple loops.		

8.2.2.ITH.3: Identify how technology impacts or improves life.8.2.2.ITH.4: Identify how various tools reduce work and improve daily tasks.	
 Essential Questions: What is/are: coding? directions? algorithms? computer program? sequence? an event? What does it mean to debug? 	 Enduring Understandings: Individuals develop and follow directions as part of daily life. A sequence of steps can be expressed as an algorithm that a computer can process. Computers follow precise sequences of steps that automate tasks. Complex tasks can be broken down into simpler instructions, some of which can be broken down even further. People work together to develop programs for a purpose, such as expressing ideas or addressing problems. The development of a program involves identifying a sequence of events, goals, and expected outcomes, and addressing errors (when necessary). Technology has changed the way people live and work. Various tools can improve daily tasks and quality of life.

Learning Targets: Students will be able to:

- Define and utilize coding vocabulary
- Form an algorithm
- Create an event
- Debug an issue found to make the code run

Instructional Best Practices/Activities:

To assist in meeting the goals of this unit of study, throughout the course of the school year, students may:

- Utilize beginner coding apps, such as Code Karts and Code Safari to introduce coding practices online.
- Access Code.org to complete the Hour of Code and beyond.
- Create an event and algorithm using various games on http://botlogic.us/
- Order movement commands (cardinal directions) as sequential steps in a program (CODE p. 37)
- Represent an algorithm as a computer program (CODE p. 37)
- Count the number of times an action should be executed and represent it as instruction in a program (CODE p. 37)
- Utilize Google's Santa Tracker Code Boogie to code an elf to dance using block coding.

Assessment:

- Assessment checklists
- Teacher observations
- Completion of activity/tasks/projects
- Student reflection

- Santa Tracker <u>https://santatracker.google.com/village.html</u> Code Boogie
- Code.org
- Tynker
- Code Safari
- Code Karts
- Google
- Safari
- <u>http://botlogic.us/</u>

Unit of Study: Coding Module 2	
Grade Level: Middle School Mixed Grade Levels	Suggested Timeframe: 1-2 weeks
Primary Performance Expectations/Standards: Interaction of Technology and Humans 8.2.2.ITH.3: Identify how technology impacts or improves life. Data Analysis 8.1.2.DA.2: Store, copy, search, retrieve, modify, and delete data using a computing device. Algorithms and Programming 8.1.2.AP.3: Create programs with sequences and simple loops to accomplish tasks. 8.1.2.AP.4: Break down a task into a sequence of steps. 8.1.2.AP.5: Describe a program's sequence of events, goals, and expected outcomes. 8.1.2.AP.6: Debug errors in an algorithm or program that includes sequences and simple loops.	
 Essential Questions: What is/are: coding? directions? algorithms? computer program? sequence? an event? a loop What does it mean to debug? 	 Enduring Understandings: Technology has changed the way people live and work. Computers store data that can be retrieved later. Data can be copied, stored in multiple locations, and retrieved. Computers follow precise sequences of steps that automate tasks. People work together to develop programs for a purpose, such as expressing ideas or addressing problems. The development of a program

involves identifying a sequence of events, goals, and expected outcomes, and addressing errors (when necessary).

Learning Targets:

Students will be able to:

- Define and utilize computer science vocabulary (algorithm, event, debug, loops)
- Form an algorithm
- Create an event
- Debug an issue found to make the code run

Instructional Best Practices/Activities:

To assist in meeting the goals of this unit of study, students may:

- Access Kodable to complete lessons:
 - Unplugged Unit Opening Activities
 - Unplugged Activities
- Game Courses:
 - Kodable Basics (Beginner) 3 Lessons
 - Intro to Sequence: 1, 2, 3... Roll! 3 Levels
 - Intro to Conditions: If Fuzz, Then Roll 4 Levels
 - Intro to Debugging: Buggy Basics 3 Levels
- Introductory Courses (Beginner) 4 Lessons
 - What is Sequence? 3 min video
 - Alien Algorithms (Sequence Practice) 3 Levels
 - What are Conditions? 3 min video
 - Fuzzy Fiesta (Conditions Practice) 3 Levels
 - Bugs Away (Debugging Practice) 3 Levels
 - Crossy Fuzz (Conditions Practice) 3 Levels
- Intermediate Course 3 Lessons
 - \circ What is Loops? 4 min video
 - Loopy Lessons 3 Levels
 - Loop, Roll, Repeat 3 Levels
 - Iffy Loops 3 Levels
- Sequence Extra Courses 3 Lessons
 - Love Landing 3 Levels (Beginner)
 - Sweet Street 3 Levels
 - Rainbow Road 3 Levels (Debugging)

Assessment:

To show evidence of meeting the goals unit of study, students may be assessed as follows:

- Successfully writing an algorithm
- State how computer science affects the world that we live in
- Correctly use computer science terminology in conversations
- Completing an event
- Teacher observation
- Completed projects

- Code.org
- Kodable

• Classroom Materials https://dashboard.kodable.com/v1/#/curriculum/printables/classroom materials • Vocabulary Cards https://dashboard.kodable.com/v1/#/curriculum/printables/vocab_cards • Seesaw Activities https://www.kodable.com/learn/seesaw-kodable-activities/ Tynker Jr • Madewithcode.com • Google • Safari • Scratch Jr • Botlogic.us • Literacy Connection • • How to Code a Sandcastle Online Book

https://youtu.be/0bXwTjgl6GE

Unit of Study: Coding Module 3		
Grade Level: Middle School Mixed Grade Levels	Suggested Timeframe: 2-3 weeks	
Primary Performance Expectations/Standards:		
Algorithms and Programming8.1.2.AP.3: Create programs with sequences and simple loops to accomplish tasks.8.1.2.AP.4: Break down a task into a sequence of steps.8.1.2.AP.5: Describe a program's sequence of events, goals, and expected outcomes.8.1.2.AP.6: Debug errors in an algorithm or program that includes sequences and simple loops.Interaction of Technology and Humans8.2.2.ITH.3: Identify how technology impacts or improves life.Engineering Design8.2.2.ED.2: Collaborate to solve a simple problem, or to illustrate how to build a product using		
the design process. 8.2.2.ED.3: Select and use appropriate tools and materials to build a product using the design process		
 Essential Questions: What is a computer? What are examples of computers? How do these computers know what to do? What are errors that need to be solved in our classroom? What are ways we can fix these errors? Why is learning computer programming Important? What are tasks or problems in our lives that we can decompose? (ex. 	 Enduring Understandings: Technology has changed the way people live and work. Computers follow precise sequences of steps that automate tasks. People work together to develop programs for a purpose, such as expressing ideas or addressing problems. The development of a program involves identifying a sequence of 	

Playing a sport, getting ready for school)

- What happens when I ask a question? When I ask the question (event), you raise your hand to share an answer. You can say your answer when I call on you (event).
- What keyword starts a conditional?
- What are examples of stacks in the real world?
- What are examples of queues in the real world?
- How can we work together in collaborative teams when programming?
- What kind of game do you want to make?

events, goals, and expected outcomes and addressing errors (when necessary).

• Engineering design is a creative process for meeting human needs or wants that can result in multiple solutions.

Learning Targets:

Students will be able to:

- define and utilize computer science vocabulary <u>https://dashboard.codespark.com/dashboard/lessons/plans/glossary</u>
- create an algorithm
- debug find and fix errors in a program, use trial & error to solve a problem
- use a loop when a sequence of commands repeat & recognize loops in the world around me
- decompose a problem by breaking it down into smaller problems & use decomposition to solve a problem
- sequence commands to solve a problem and understand the concept of efficiency and that efficient programs use the fewest commands
- understand events cause things to happen
- understand a programmer can use IF statements to make actions happen under certain conditions & use IF statements to make programs more dynamic and efficient
- understand that data is stored and organized, stacks have elements that can be added and removed from the top and a queue has a front and a back
- work with my peers to solve problems and program on CodeSpark
- use game design strategies to create my own game <u>https://youtu.be/RM04n0-QtNo</u>

Instructional Best Practices/Activities:

To assist in meeting the goals of this unit of study, students may:

- Access Code.org to complete the Hour of Code and beyond courses Utilize CodeSpark to:
- Engage, Explore & Enrich with Algorithms & Sequencing (Lessons 1.1, 1.2 & 1.3)
- Engage, Explore & Enrich with Debugging (Lessons 2.1, 2.2 & 2.3)
- Engage, Explore & Enrich with Loops (Lessons 3.1, 3.2 & 3.3)
- Engage & Explore with Decomposition (Lessons 4.1, 4.2)
- Engage, Explore & Enrich with Advanced Sequencing (Lessons 5.1, 5.2 & 5.3)
- Engage & Explore with Events (Lessons 6.1, & 6.2)
- Engage, Explore & Enrich with Conditionals (Lessons 7.1, 7.2 & 7.3)
- Engage & Explore with Stacks & Queues (Lessons 8.1, & 8.2)
- Engage & Explore with Pair Programming (Lessons 9.1, & 9.2)

• Engage & Explore with Game Maker (Lessons 10.1, & 10.2)

Assessment:

To show evidence of meeting the goals unit of study, students may be assessed as follows:

- successfully defining and utilizing computer science vocabulary
- understanding how to create an algorithm, debug, use a loop, decompose problems, sequence commands, understand events, IF statements, stacks, queues, solve problems, and design a game by successfully completing all activities on CodeSPark. Student progress is monitored by a lesson on the CodeSpark teacher dashboard.

Resources:

• CodeSpark

Other Resources:

- Code.org
- Botlogic
- Kodable
- Tynker
- Scratch

Unit of Study: Coding Module 4	
Grade Level: Middle School Mixed Grade Levels	Suggested Timeframe: 2-3 weeks
 Primary Performance Expectations/Standards 8.1.5.AP.1: Compare and refine multiple algorithms for the same task and determine which is the most appropriate. 8.1.5.AP.2: Create programs that use clearly named variables to store and modify data. 8.1.5.AP.3: Create programs that include sequences, events, loops, and conditionals. 8.1.5.AP.4: Break down problems into smaller, manageable sub-problems to facilitate program development. 8.1.5.AP.5: Modify, remix, or incorporate pieces of existing programs into one's own work to add additional features or create a new program. 8.1.5.AP.6: Develop programs using an iterative process, implement the program design, and test the program to ensure it works as intended. 8.1.5.IC.1: Identify computing technologies that have impacted how individuals live and work and describe the factors that influenced the changes. 	
 Essential Questions: How can different algorithms yield the same result? How can we determine which algorithm is the best option? What are loops, and how do they apply to programming? How can we look at a program and determine its different elements? How do we break a program into smaller parts? 	 Enduring Understandings: Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific use than others. A variety of control structures are used to change the flow of program execution (e.g., sequences, events, loops, conditionals). Programs can be broken down into

Learning Targets:

- Students will understand the meaning of algorithms.
- Students will be able to create simple algorithms to solve puzzles and complete tasks within a designated coding application.
- Students will be able to create loops by revising their algorithms in order to make them more efficient.
- Students will analyze algorithms and determine which is the most efficient and effective to solve a particular task.
- Students will understand the meaning of loops and their application in coding.
- Students will be able to create for loops within the coding application chosen by their teacher
- Students will discuss how evolving technologies have benefitted the world due to the impact of coding.
- Students will explain how communities use data and technology to develop measures in response to effects of climate change.

Instructional Best Practices/Activities:

- Begin the unit with a brief programming pre-Assessment to gauge students' current understanding of coding, their comfort level with different applications, and any preliminary questions they may have. Use information from pre Assessment to differentiate lessons.
- Short, relevant "unplugged" activities are recommended to introduce a concept. These provide real-world connections to things students already understand and help struggling learners to conceptualize coding vocabulary.
- After introducing related vocabulary and concepts through unplugged activities or real world examples, teacher will guide students through the in-class application of the concept using the coding platform of their choice.
- Struggling students may be partnered with others in order to work collaboratively for greater success. Additionally, teacher can pull small groups of students having difficulty to work more closely and explain in greater detail.

Assessment:

- Student work on various coding platforms can be evaluated through the teacher dashboard/data provided through that application.
- Teacher surveys, short quizzes and responses
- Teacher observation, questioning, anecdotal notes

- CodeSpark- iPad App •

- Code.org
 Possible Unplugged Activities

 Unplugged Coding Activities from Kodable
 Teaching Algorithms with Origami
 Loops Unplugged (Jumping Jack Activity)
- District Robots

Unit of Study: Coding Module 5		
Grade Level: Middle School Mixed Grade Levels	Suggested Timeframe: 2-3 weeks	
Primary Performance Expectations/Standards:		
 8.1.5.AP.1: Compare and refine multiple algorithms for the same task and determine which is the most appropriate. 8.1.5.AP.2: Create programs that use clearly named variables to store and modify data. 8.1.5.AP.3: Create programs that include sequences, events, loops, and conditionals. 8.1.5.AP.4: Break down problems into smaller, manageable sub-problems to facilitate program development. 8.1.5.AP.5: Modify, remix, or incorporate pieces of existing programs into one's own work to add additional features or create a new program. 8.1.5.AP.6: Develop programs using an iterative process, implement the program design, and test the program to ensure it works as intended. 8.1.5.IC.1: Identify computing technologies that have impacted how individuals live and work and describe the factors that influenced the changes. 		
 Essential Questions: In programming, where there are many possible solutions, how can we choose the algorithm that solves a specific problem more efficiently? What are conditional statements, and how do they apply to programming? How is a programming language like spoken/written language? How is it different? How can we look at a program and determine its components? How do we break a program into smaller parts? Why would this be advantageous? Why do individuals develop programs? What steps in the process are necessary to create a functional program? 	 Enduring Understandings: Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific use than others. A variety of control structures are used to change the flow of program execution (e.g., sequences, events, loops, conditionals). Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that already exist. Individuals develop programs using an iterative process involving design, implementation, testing, and review. The development and modification of computing technology is driven by an individual's needs and wants and can 	

affect individuals differently.

Learning Targets:

- Students will recall information introduced in prior modules and understand the meaning of algorithms and loops in the context of programming.
- Students will understand the meaning and context of conditional (if-then) statements.
- Students will be able to revise the conditional statements they create in order to make their algorithms more efficient.
- Students will analyze algorithms and determine which is the most efficient and effective to solve a particular task.
- Students will make meaningful connections between coding conditional statements and other academic areas as well as their everyday lives.
- Students will be able to create conditional statements within the coding application chosen by their teacher
- Students will discuss how evolving technologies have benefitted the world due to the impact of coding.
- Students will explain how communities use data and technology to develop measures in response to effects of climate change.

Instructional Best Practices/Activities:

- Begin the unit with a brief programming pre-Assessment to gauge students' current understanding of coding, their comfort level with different applications, and any preliminary questions they may have. Use information from pre Assessment to differentiate lessons.
- Review coding fundamentals in introductory lessons, including algorithms and for loops.
- Provide students with coding vocabulary Resources and allow them time to explore the different terms while creating a Slides presentation putting vocabulary into their own words or finding an example of the term from their experience.
- Use real world examples to explain conditionals. Many video games relate to this topic, quick examples/demonstrations can be done for engagement and explanation.
- Utilize turn and talk or small group discussions to define conditional statements in real life using cause/effect as an example (connect to prior knowledge from ELA)
- Short, relevant "unplugged" activities can be an effective way to introduce or further explain conditionals.
- Once the concept has been presented, allow students time to work independently or cooperatively based on their mastery levels. Work with small groups as appropriate in order to clarify concepts or deepen understanding.

Assessment:

- Student work on various coding platforms can be evaluated through the teacher dashboard/data provided through that application.
- Teacher surveys, short quizzes and responses
- Teacher observation, questioning, anecdotal notes

- CodeSpark- iPad app
- Code.org- website
- Swift Playgrounds (Learn to Code 1)- iPad app
- Learn To Code 1 & 2 Swift Vocabulary Cards
- Swift Vocabulary- Intro Assignment

- •
- Unplugged Activities Conditionals with Cards
 - Simon Says If Then Else
 - Unplugged: Red Light, Green Light
- District Robots •

Last updated 8/3/23

<u>Appendix</u>

Standards in Action

Montague Township School District believes in offering an interdisciplinary approach to teaching and learning because students are able to make connections and relationships by bringing together separate content disciplines, skills and knowledge around common themes, issues, or problems. The NJ Department of Education mandates the following be identified as areas of study beneficial to integration into all grade levels and content areas.

Please click the hyperlink for further information on each area:

Career Readiness, Life Literacies, and Key Skills Climate Change Education Contributions of Disabled and LGBT Individuals Holocaust Education Amistad Commission Social and Emotional Learning Diversity, Equity and Inclusion Asian American Pacific Islander

Types of Assessments

Students will be assessed across the units and year in a variety of ways. The link below indicates resources for developing assessments and general examples of assessments that teachers may utilize across all of the content areas.

Formative, Summative, Alternative, and Benchmark Assessments

Accommodations & Modifications for Special Education, ELL, G&T, 504 Plans and At Risk:

Modifications and Accommodations