

Coding

Statement of Organization:

This course will provide students with the core knowledge needed to program in any language. They will begin by exploring programming including variables, input and output, conditions, loops, and expressions and quickly move on to using block coding to code sprites as well as a Raspberry PI and Breadboard. With this practical introduction to computer science concepts, students will be prepared to take their knowledge of programming to the next level.

Expected Levels of Achievement:

Students are expected to achieve mastery of concepts and skills presented by obtaining an eight percent (80%) or better on all graded materials and projects.

Methods of Evaluation:

Student competence will be assessed through the use of one or more of the following methods:

- Project
- Demonstration
- Rubric
- Checklist
- Worksheet

Estimated Instructional Time:

Coding is a one-credit elective which meets for a minimum of one hundred twenty (120) clock hours and is available for students in grades 9-12.

Technology Integration:

Students will use technology as a core requirement of the course.

Standards:

NGSS

- HS-ETS1-1
- HS-ETS1-2
- HS-ETS1-3
- HS-ETS1-4
- HS-ETS2-2
- HS-ETS4-3

ISTE

Innovative Designer 4a, 4b, 4c, 4d

Computational Thinker 5c, 5d

Creative Communicator 6a, 6b, 6c, 6d

Big Idea/Reporting Category	Concepts	Competencies/Eligible Content	Vocabulary
INTRODUCTION TO PROGRAMMING	Introduce students to programming and how computers “think” through demonstrations and activities to prepare them to write their own set of instructions for a computer (a.k.a. a computer program).	Investigate how computers “think.”	Program Format Hyperlink Animation Effect
		Explore the consequences of instructions that are not accurate, correct, and/or detailed.	
		Confirm that there is no “interpretation” when it comes to developing instructions for computers.	
		Program a game using slide presentation software • Test the game. • Add obstacles. • Add spin action. • Add timing effect • Add an action • Add a motion path • Add levels	
BLOCK CODING	Students will solve basic problems and receive an introduction to block coding in a non-threatening, graphical, and game-like environment.	Perform introductory block coding.	
		Encounter conditional and looping statements.	
		Use some advanced coding techniques such as functions.	
EVENT DRIVEN PROGRAMMING	Students will continue working with block coding and event driven programming to create an interactive tutorial.	Use block coding to code various sprites.	Motion Looks Sound Events Control – conditional and looping statements Sensing Operators – used with condition and variables.
		Use events.	
		Code a short, interactive game	
SEQUENTIAL CONTROL STRUCTURES	Students will demonstrate their understanding of sequential control structure by programming two sprites to interact. This module is essential as subsequent modules will build upon what is presented here.	Use sequential control structure in programming.	
		Properly sequence instructions using the wait block.	
		Program sprites to tell a knock-knock joke.	

Big Idea/Reporting Category	Concepts	Competencies/Eligible Content	Vocabulary
LOOPING, ANIMATIONS, AND SOUNDS	Students will use looping to animate sprites. Students will import and play music with their Animations.	Students will use loops to investigate how they work in programming languages.	Forever loop Condition
		Students will import sprites with multiple costumes and switch between those costumes using looping structures.	
		Students will bring sounds into their animation and play music.	
		Students will program and produce a music video.	
	Broadcasting is the fourth type of event that students will learn to control their code. Students will send a broadcast message to sprites and the stage, which receive the message and perform code associated with the receipt of that message.	Create events.	
		Change backgrounds for the stage using events.	
		Properly use the Broadcast event to create a story in Scratch.	
VARIABLES AND CONDITIONS	Students will be introduced to variables and the decision-making control structure (a.k.a. if statements). They will learn how to create a timer with variables and loops. They will complete a guided practice for creating a game. At the end of this module, students will be ready to code their own game.	Correctly employ variables in their programs.	
		Practice using if and if/else statements.	
		Code a simple timer.	
		Program and produce a simple, interactive game.	
DEVELOPING YOUR OWN GAME	Students will code a game.	Code a playable game.	
PREPARE BREADBOARD TO RECEIVE CODE	Students will set up a Raspberry Pi and prepare it to receive and run code to control breadboard components. Students will focus on getting their breadboards set up and putting electrical components on the breadboards.	Set up the breadboard for a Raspberry Pi correctly.	Breadboard
		Safely connect electrical components on a breadboard.	
		Handle a breadboard safely and keep it in working order	
		Add the appropriate blocks to code a Raspberry Pi and breadboard components.	