Southern Cayuga Central School District – Curriculum Map

Subject: **Computer Coding** School Year: 2021-2022

Title or Topics w/ NYS Standards	Essential Questions & Vocabulary	Content Skills (Activities to cover Essential Questions)	Major Assessments (Tests, Project, etc.)	Time Frame
Introduction to Computational Thinking • **CCSS.MATH.PRACTICE.MP1** Make sense of problems and persevere in solving them. • **CCSS.MATH.PRACTICE.MP5** Use appropriate tools strategically.	• How can I begin to think like a programmer? • What sorts of problem solving skills to programmers use? • Pseudo Code • Algorithm • Top-Down Programming	• What makes a good programmer? Class Discussion • Blind Draw – Group Activity	• In-class Assignments	September (First Week)
Strings • **CCSS.MATH.PRACTICE.MP1** Make sense of problems and persevere in solving them. • **CCSS.MATH.PRACTICE.MP5** Use appropriate tools strategically. • **CCSS.MATH.PRACTICE.MP6** Attend to precision.	• How can I manipulate text in my program? • How can I communicate with a user within my program? • What are some basic coding conventions that I should be using when I program? • String • Concatenate • Variable • Input • Output • Comments • str() • input() • print() • .py	• Examine Code – Group Discussion • Hello World – Write first program • Project – Replicate Text Art • Run “conversation.py” – Practice input and string manipulation • Project – Mad Libs • Peer Code Critique	• Warm Ups • Exit Tickets	September
Boolean Statements • **CCSS.MATH.PRACTICE.MP1** Make sense of problems and persevere in solving them. • **CCSS.MATH.PRACTICE.MP5** Use appropriate tools strategically.	• How can I use numeric values in my code? • How can I break my code down to run differently in different situations? • Integer • Floating Point Number • Import • Random • random.randint()	• Logic Statements – Unplugged Activity • Following a chain of logic commands • Integers vs. Floating Point Numbers & Generating Random Numbers • Writing if statements	• Warm Ups • Exit Tickets • Project – Simple Guessing Game • Project – Choose Your Own Adventure	September - October

• **CCSS.MATH.PRACTICE.MP6** Attend to precision.
• **CCSS.MATH.PRACTICE.MP7** Look for and make use of structure.
	• Boolean Argument
	• Equal
	• Inequalities
	  ○ Greater than
	  ○ Less than
	• If
	• Else If
	• Else
	• int()
	• float()
	• if():
	• elif():
	• else:
		• Writing if else statements
		• Project – Simple Guessing Game		

Loops • **CCSS.MATH.PRACTICE.MP1** Make sense of problems and persevere in solving them. • **CCSS.MATH.PRACTICE.MP5** Use appropriate tools strategically**.** • **CCSS.MATH.PRACTICE.MP6** Attend to precision. • **CCSS.MATH.PRACTICE.MP7** Look for and make use of structure.	• How can I repeat code? • For loop • While Loop • Range • for i in range(): • while(): • range():	• What would make the "Basic Guessing Game" better? – Group Brainstorm • How computers count – an exploration in for loops • Update "Basic Guessing Game" • For loops vs. While loops • Update "Basic Guessing Game" again • Peer Code Critique	• Warm Ups • Exit Tickets • Project – Improved Guessing Game	October
Math with Floats and Ints • **CCSS.MATH.PRACTICE.MP1** Make sense of problems and persevere in solving them. • **CCSS.MATH.PRACTICE.MP4** Model with mathematics. • **CCSS.MATH.PRACTICE.MP5** Use appropriate tools strategically.	• How can I manipulate numeric variables to update my code? • Are there ways to get additional built in methods? • Import • Addition (+) • Multiplication (*) • Subtraction (-) • Division (/) • Exponents (**) • Square Root (math.sqrt()) • Trig Functions (math.sin())	• Evaluating Expressions in Python • Syntax of mathematical functions • Project – Total Bill • Combining what we know – Using mathematical values to make decisions – What type of roots? Group exercise • Project – What type of triangle?	• Warm Ups • Exit Tickets • Project – Total Bill • Project – What type of triangle is it?	October – November
Physical Computing with Raspberry Pi	• What sorts of physical technology can be controlled with code?	• Set Up Raspberry Pi Computers • Creating Circuits	• Warm Up • Exit Tickets	November - December

• **CCSS.MATH.PRACTICE.MP1** Make sense of problems and persevere in solving them. • **CCSS.MATH.PRACTICE.MP5** Use appropriate tools strategically. • **CCSS.MATH.PRACTICE.MP6** Attend to precision.	• Raspberry Pi • Circuit • Button • Switch • Timer • Breadboard • LED • Sensor	• Timers • Controlling Lights • Project – Morse Code • The structure of buttons within a circuit • Using buttons • Project – Reflex Game	• Project – Morse Code • Project – Reflex Game	
Functions / Methods • **CCSS.MATH.PRACTICE.MP1** Make sense of problems and persevere in solving them. • **CCSS.MATH.PRACTICE.MP5** Use appropriate tools strategically. • **CCSS.MATH.PRACTICE.MP6** Attend to precision.	• How can I create a new method to use throughout my code? • Function • Method • Arguments • def • return	• Evaluate code – What do you want to be able to change? – Class discussion • Writing a function • Practice – write a function that prints out a small piece of text art • Using arguments in a function • Practice – write a function that multiplies a number by 3 and prints it out • Using return in a function • Practice – write a function that returns True if the argument is an even number and False if it is odd • Project – Finding the Maximum	• Warm Ups • Exit Tickets • Project – Finding the Maximum Function	January
Midterm Project • **CCSS.MATH.PRACTICE.MP1** Make sense of problems and persevere in solving them. • **sCCSS.MATH.PRACTICE.MP5** Use appropriate tools strategically.	•	• Come up with an idea for a program with a purpose that would make use of the structures you have learned this year. (Elevator Pitch) • Write pseudo code describing what your code will actually do in different situations (Project Proposal) • Write the program (Midterm Project) • Peer Code Critique	• Elevator Pitch • Project Proposal • Midterm Project	January
Lists & More with Strings	• How do I manipulate or check many values at	• Creating and using lists	• Warm Ups	February

• **CCSS.MATH.PRACTICE.MP1** Make sense of problems and persevere in solving them.
• **CCSS.MATH.PRACTICE.MP5** Use appropriate tools strategically.
	once?
	• List
	• Index
	• Splicing
	  ○ [:]
	  ○ [#:]
	  ○ [:#]
	  ○ [#:#]
	• Append
	• Remove
	• In
		• Checking to see if something is in the list
		• Looping through lists
		• Adding and removing items from a list
		• Indexing a list
		• Project – Palindrome Check
		• Using splicing with strings
		• Changing one (or more) letters within a string
		• Project - Hangman
			• Exit Tickets
			• Project - Palindrome Check
			• Project - Hangman	

Graphing in Python • **CCSS.MATH.PRACTICE.MP1** Make sense of problems and persevere in solving them. • **CCSS.MATH.PRACTICE.MP4** Model with mathematics. • **CCSS.MATH.PRACTICE.MP5** Use appropriate tools strategically.	• How can I visualize information using Python? • Equation • Mathplotlib • Graph • Plt.plot() • Linspace()	• Creating a plot from two lists • Creating a plot from an equation and linspace() • Adding labels and changing colors on the graph • Project – Physics Data • Creating 3D images • Determining equations for 3D images & adjusting coefficients • Project – 3D Image • Project – Pixar Code	• Warm Ups • Exit Tickets • Project – Physics Data • Project – 3D Image • Project – Pixar Code	February - March
Generalizing Code • **CCSS.MATH.PRACTICE.MP1** Make sense of problems and persevere in solving them. • **CCSS.MATH.PRACTICE.MP2** Reason abstractly and quantitatively. • **CCSS.MATH.PRACTICE.MP5** Use appropriate tools strategically. • **CCSS.MATH.PRACTICE.MP7** Look for and make use of structure.	• How can I write code that will work in any possible situation?	• Group Activity – Edit pre-written code so that it is more general	• In-class Assignments	March
Code Quest Practice • **CCSS.MATH.PRACTICE.MP1** Make sense of	• How can I break down a task into smaller and more manageable parts? • How can I solve a problem with many different	• Top-down design practice • Group Activities – Work on one Code Quest challenge at a time	• Warm Ups • Exit Tickets • Project –	March - April

problems and persevere in solving them. • **CCSS.MATH.PRACTICE.MP5** Use appropriate tools strategically.	possible starting conditions? • Top-down design	(Switch groups every day) • Group Activities – How many challenges can you complete? (Switch groups every other day) • Group Activity – One Week Challenge (Which team can earn the most points by the end of the week – students make decisions about how to divide up work)	Previous Code Quest Challenge • Optional: Code Quest Competition	
AppInventor • **CCSS.MATH.PRACTICE.MP1** Make sense of problems and persevere in solving them. • **CCSS.MATH.PRACTICE.MP5** Use appropriate tools strategically.	• How can I create an app that works on my phone?	• Introduction to App Inventor & block coding • Complete two basic level tutorials • Complete two intermediate level tutorials • Project – Create your own app • Peer Critique	• Warm Ups • Exit Tickets • Project – Create your own app	May
Final Project • **CCSS.MATH.PRACTICE.MP1** Make sense of problems and persevere in solving them. • **CCSS.MATH.PRACTICE.MP5** Use appropriate tools strategically.		•	•	June