Coding in Minecraft Advanced Coding with Python

Scope and Sequence

Lesson Title	Lesson Objectives	CSTA Standards	Average Completion Time (in hours)
Lesson 1: Introduction to MakeCode with Minecraft (Optional lesson)	• Learners will develop an understanding of how they can use code to build in Minecraft and develop a relationship with 'Agent.' • Learners code Agent to navigate through a maze. • Learners program Agent to place and break blocks • Learners will design and create a code (including a loop) to build a structure.	• **1B-AP-09** Create programs that use variables to store and modify data. • **1B-AP-10** Create programs that include sequences, events, loops, and conditionals. • **1B-AP-11** Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. • **1B-AP-13** Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.	2
Lesson 2: Writing Computer Algorithms	• Learners will understand what an algorithm is. • Learners will write algorithms. • Learners will understand the requirement for thorough instructions in computer programs.	• **2-AP-10** Use flowcharts and/or pseudocode to address complex problems as algorithms. • **2-AP-14** Create procedures with parameters to organize code and make it easier to reuse. • **2-AP-17** Systematically test and refine programs using a range of test cases	2

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CSTA Standards	Average Completion Time (in hours)
		• **2-AP-19** Document programs in order to make them easier to follow, test, and debug.	
Lesson 3: Moving to Python	• Learners will Understanding text-based programming language • Learners will be able to identify equivalent Python code for MakeCode blocks • Learners will be able to make modifications to Python code to affect the output	• **2-AP-12** Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. • **2-AP-14** Create procedures with parameters to organize code and make it easier to reuse. • **2-AP-17** Systematically test and refine programs using a range of test cases	2
Lesson 4: Python Loops	• Learners will identify and modify Repeat loops in Python • Learners will identify and modify While loops in Python • Learners will identify and modify For loops in Python	• **2-AP-12** Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. • **2-AP-14** Create procedures with parameters to organize code and make it easier to reuse. • **2-AP-17** Systematically test and refine programs using a range of test cases • **1B-AP-10** Create programs that include sequences, events, loops, and conditionals.	3

Lesson Title	Lesson Objectives	CSTA Standards	Average Completion Time (in hours)
Lesson 5: Python Logic	• Learners will identify and modify IF THEN and IF THEN ELSE statements in Python • Learners will use Comparison operators in Python • Learners will use Boolean operators in Python	• **2-AP-12** Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. • **2-AP-14** Create procedures with parameters to organize code and make it easier to reuse. • **2-AP-17** Systematically test and refine programs using a range of test cases • **1B-AP-10** Create programs that include sequences, events, loops, and conditionals.	3
Lesson 6: Errors	• Learners will define and Identify Semantic Errors • Learners will define and Identify Syntax Errors	• **2-AP-10** Use flowcharts and/or pseudocode to address complex problems as algorithms. • **2-AP-17** Systematically test and refine programs using a range of test cases • **2-AP-19** Document programs in order to make them easier to follow, test, and debug.	3
Lesson 7: Project	• Learners will Design and develop a programmatic solution for a problem using Python.	• **2-AP-13** Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. • **2-AP-14** Create procedures with parameters to organize code and make it easier to reuse.	5 - 10

Lesson Title	Lesson Objectives	CSTA Standards	Average Completion Time (in hours)
		• **2-AP-17** Systematically test and refine programs using a range of test cases • **1B-AP-10** Create programs that include sequences, events, loops, and conditionals.	
		Total Hours	20 - 30