Coding in Minecraft Intermediate Coding Using MakeCode

Scope and Sequence

Lesson Title	Lesson Objectives	CSTA Standards	ISTE 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 programme Agent to place and break blocks • Learners 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.	**1.5.c Decompose Problems** Students break problems into component parts, extract key information and develop descriptive models to understand complex systems or facilitate problem-solving. **1.5.d Algorithmic Thinking** Students understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions.	2
Lesson 2: Using Variables in MakeCode	• Learners will learn the types of variables and what they are used for. • Learners will learn how to create a variable in MakeCode	• **2-AP-11** Create clearly named variables that represent different data types and perform operations on their values. • **2-AP-12** Design and iteratively develop programs that combine control	**1.5.c Decompose Problems** Students break problems into component parts, extract key information and develop	2

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CSTA Standards	ISTE Standards	Average Completion Time (in hours)
	• Learners will use how to use a variable with an argument in MakeCode	structures, including nested loops and compound conditionals. • **2-AP-13** Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	descriptive models to understand complex systems or facilitate problem-solving. **1.5.d Algorithmic Thinking** Students understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions.	
Lesson 3: Logic in MakeCode	• Learners will learn what conditionals are in MakeCode. • Learners will learn what an If Statement is in MakeCode. • Learners will learn how to use If Statements in MakeCode. • Learners will learn what an If / Else Statement is in MakeCode. • Learners will learn how to use If / Else Statements in MakeCode.	• **2-AP-11** Create clearly named variables that represent different data types and perform operations on their values. • **2-AP-12** Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. • **2-AP-13** Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	**1.5.a Problem Definitions** Students formulate problem definitions suited for technology-assisted methods such as data analysis, abstract models and algorithmic thinking in exploring and finding solutions. **1.5.c Decompose Problems** Students break problems into component parts, extract key information and develop	2

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CSTA Standards	ISTE Standards	Average Completion Time (in hours)
			descriptive models to understand complex systems or facilitate problem-solving. **1.5.d Algorithmic Thinking** Students understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions.	
Lesson 4: Advanced Logic in MakeCode	• Learners will learn what and If / Else If/ Else statement is in MakeCode. • Learners will learn how to use an If / Else If/ Else statement in their code.	• **2-AP-11** Create clearly named variables that represent different data types and perform operations on their values. • **2-AP-12** Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. • **2-AP-13** Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	**1.5.a Problem Definitions** Students formulate problem definitions suited for technology-assisted methods such as data analysis, abstract models and algorithmic thinking in exploring and finding solutions. **1.5.c Decompose Problems** Students break problems into component parts, extract key information and develop	3

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CSTA Standards	ISTE Standards	Average Completion Time (in hours)
			descriptive models to understand complex systems or facilitate problem-solving. **1.5.d Algorithmic Thinking** Students understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions.	
Lesson 5: Loops in MakeCode	• Learners will learn what loops are in MakeCode. • Learners will identify and use Repeat Loops in MakeCode. • Learners will identify and use While loops in MakeCode. • Learners will identify and use For loops in MakeCode.	• **2-AP-11** Create clearly named variables that represent different data types and perform operations on their values. • **2-AP-12** Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. • **2-AP-13** Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs	**1.5.a Problem Definitions** Students formulate problem definitions suited for technology-assisted methods such as data analysis, abstract models and algorithmic thinking in exploring and finding solutions. **1.5.c Decompose Problems** Students break problems into component parts, extract key information and develop	3

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CSTA Standards	ISTE Standards	Average Completion Time (in hours)
		• **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.	descriptive models to understand complex systems or facilitate problem-solving. **1.5.d Algorithmic Thinking** Students understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions.	
Lesson 6: Advanced Loops in MakeCode	• Learners will identify and use Forever loops in MakeCode. • Learners will learn how to use the on pause command in MakeCode. • Learners will identify and use the on start loop in MakeCode. • Learners will identify and use functions in MakeCode.	• **2-AP-11** Create clearly named variables that represent different data types and perform operations on their values. • **2-AP-12** Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. • **2-AP-13** Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs	**1.5.a Problem Definitions** Students formulate problem definitions suited for technology-assisted methods such as data analysis, abstract models and algorithmic thinking in exploring and finding solutions. **1.5.c Decompose Problems** Students break problems into component parts, extract key information and develop	3

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CSTA Standards	ISTE Standards	Average Completion Time (in hours)
		• **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.	descriptive models to understand complex systems or facilitate problem-solving. **1.5.d Algorithmic Thinking** Students understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions.	
Lesson 7: Project	• Learners will be able to effectively use standard programming blocks in MakeCode such as loops, variables and logic • Learners will be able to design and develop a programmatic solution for a given problem	• **2-AP-11** Create clearly named variables that represent different data types and perform operations on their values. • **2-AP-12** Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. • **2-AP-13** Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs	**1.5.a Problem Definitions** Students formulate problem definitions suited for technology-assisted methods such as data analysis, abstract models and algorithmic thinking in exploring and finding solutions. **1.5.b Data Sets** Students collect data or identify relevant data sets, use digital tools to analyze them and represent data in various ways to	5 - 10

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	CSTA Standards	ISTE Standards	Average Completion Time (in hours)
			facilitate problem-solving and decision-making. **1.5.c Decompose Problems** Students break problems into component parts, extract key information and develop descriptive models to understand complex systems or facilitate problem-solving. **1.5.d Algorithmic Thinking** Students understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions.	
		Total Hours		20 - 30

© Prodigy Learning | prodigylearning.com | codingcredentials.com