# Coding in Minecraft Expert Coding in Minecraft with Python

## Scope and Sequence

| Lesson Title | Lesson Objectives | CSTA Standards | AP CSP Learning Objective | Average Completion Time (in hours) |
|---|---|---|---|---|
| Lesson 1: Variables | Learners will understand the definition of variables.<br><br>Learners will understand the different types of variables and when they should be used in code.<br><br>Learners will be able to create variables and assign them values.<br><br>Learners will understand the definition of computing and computing networks. | 3B-AP-12: Compare and contrast fundamental data structures and their uses.<br><br>3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.<br><br>3A-AP-21: Evaluate and refine computational artifacts to make them more usable and accessible.<br><br>3B-AP-10: Use and adapt classic algorithms to solve computational problems. | CSN-1.A: Explain how computing devices work together in a network<br><br>AAP-1.A: Represent a value with a variable<br><br>AAP-1.B: Determine the value of a variable because of an assignment.<br><br>CSN-1.A: Explain how computing devices work together in a network | 5 hrs |


© Prodigy Learning | prodigylearning.com | codingcredentials.com

| Lesson Title | Lesson Objectives | CSTA Standards | AP CSP Learning Objective | Average Completion Time (in hours) |
|---|---|---|---|---|
| | | 3B-CS-02: Illustrate ways computing systems implement logic, input, and output through hardware components. | | |
| Lesson 2: Operators | Learners will understand operators and how they are used in code.<br><br>Learners will be able to identify and use operators in Python.<br><br>Learners will be able to evaluate expressions that use arithmetic operators.<br><br>Learners will be able to write expressions using relational operators. | 3B-AP-12: Compare and contrast fundamental data structures and their uses. | AAP-2.E: For relationships between two variables, expressions, or values:<br><br>a. Write expressions using relational operators. b. Evaluate expressions that use relational operators. | 5 hrs |

| Lesson Title | Lesson Objectives | CSTA Standards | AP CSP Learning Objective | Average Completion Time (in hours) |
|---|---|---|---|---|
| | Learners will be able to evaluate expressions that use relational operators. | | | |
| Lesson 3: | Learners will understand what a conditional is and why it is used in code.<br><br>Learners will learn what an If statement is and how to use it in their code.<br><br>Learners will learn what an If / Else statement is and how to use it in their code. | 3A-NI-04: Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing.<br><br>3A-AP-13: Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.<br><br>3A-AP-15: Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program | AAP-2.G: Express an algorithm that uses selection without using a programming language.<br><br>AAP-2.H: For selection: a. Write conditional statements b. Determine the result of conditional statements. | 8 hrs |

| Lesson Title | Lesson Objectives | CSTA Standards | AP CSP Learning Objective | Average Completion Time (in hours) |
|---|---|---|---|---|
| | | performance, and explain the benefits and drawbacks of choices made.<br><br>3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.<br><br>3A-AP-21: Evaluate and refine computational artifacts to make them more usable and accessible.<br><br>3B-AP-10: Use and adapt classic algorithms to solve computational problems. | | |
| Lesson 4: Advanced Logic | Learners will learn what an If / Else If / Else statement is and why it is used in code.<br><br>Learners will learn how to use an If / Else If / Else statement in their code. | 3A-CS-02: Compare levels of abstraction and interactions between application software, system software, and hardware layers | AAP-2.G: Express an algorithm that uses selection without using a programming language. | 8 hrs |

| Lesson Title | Lesson Objectives | CSTA Standards | AP CSP Learning Objective | Average Completion Time (in hours) |
|---|---|---|---|---|
| | Learners will learn what nested conditionals are and why they are used in code.<br><br>Learners will learn how to use nest conditionals in their code.<br><br>Learners will learn what the different types of computing are including their strengths, weaknesses, and when to use them. | 3A-AP-13: Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.<br><br>3A-AP-15: Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.<br><br>3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.<br><br>3A-AP-21: Evaluate and refine computational artifacts to make them more usable and accessible. | AAP-2.H: For selection: a. Write conditional statements b. Determine the result of conditional statements.<br><br>AAP-2.I: For nested selection: a. Write nested conditional statements. b. Determine the result of nested conditional statements. | |

| Lesson Title | Lesson Objectives | CSTA Standards | AP CSP Learning Objective | Average Completion Time (in hours) |
|---|---|---|---|---|
| | | 3B-AP-10: Use and adapt classic algorithms to solve computational problems. | | |
| Lesson 5: Loops | Learners will be able to identify and build code which uses while loop statements.<br><br>Learners will be able to identify and build code which uses for loop statements.<br><br>Learners will be able to identify and build code which uses nested loops. | 3A-AP-15 Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made. | AAP-2.K For iteration: a. Write iteration statements. b. Determine the result or side effect of iteration statements. | 8 hrs |
| Lesson 6: Coding Practice | Learners will be able to identify and build code that uses variables.<br><br>Learners will be able to identify and use operators in code. | 3B-AP-10: Use and adapt classic algorithms to solve computational problems.<br><br>3B-AP-12: Identify and Use Operators in Python | AAP-1.A: Represent a value with a variable<br><br>AAP-2.E: For relationships between two variables, expressions, or values: | 8 hrs |

| Lesson Title | Lesson Objectives | CSTA Standards | AP CSP Learning Objective | Average Completion Time (in hours) |
|---|---|---|---|---|
| | Learners will be able to identify and build code using complex selections and conditionals.<br><br>Learners will be able to identify and build loops and nested loops that use complex compound conditionals. | 3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.<br><br>3A-AP-13: Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.<br><br>3A-AP-15 Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made. | a. Write expressions using relational operators. b. Evaluate expressions that use relational operators.<br><br>AAP-2.H: For selection: a. Write conditional statements b. Determine the result of conditional statements<br><br>AAP-2.I: For nested selection: a. Write nested conditional statements. b. Determine the result of nested conditional statements.<br><br>AAP-2.K For iteration: a. Write iteration statements. b. Determine the result or side effect of iteration statements. | |

| Lesson Title | Lesson Objectives | CSTA Standards | AP CSP Learning Objective | Average Completion Time (in hours) |
|---|---|---|---|---|
| Lesson 7: | Learners will build a game using all the programming knowledge previously learned such as variables, operators, logic, and loops. | 3B-AP-12: Identify and Use Operators in Python<br><br>3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.<br><br>3A-AP-13: Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.<br><br>3A-AP-15 Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made. | AAP-2.E: For relationships between two variables, expressions, or values:<br><br>a. Write expressions using relational operators. b. Evaluate expressions that use relational operators.<br><br>AAP-2.H: For selection: a. Write conditional statements b. Determine the result of conditional statements<br><br>AAP-2.I: For nested selection: a. Write nested conditional statements. b. Determine the result of nested conditional statements.<br><br>AAP-2.K For iteration: a. Write iteration statements. b. Determine the result or side effect of iteration statements. | 10 hrs |

| Lesson Title | Lesson Objectives | CSTA Standards | AP CSP Learning Objective | Average Completion Time (in hours) |
|---|---|---|---|---|
| Lesson 8: Functions | Learners will be able to understand the concept of modularity and identify when modularity can or should be used in code<br><br>Learners will be able to identify, understand, call, and build functions<br><br>Learners will be able to identify, understand, call and build functions that take parameters or return data.<br><br>Learners will be able to identify and understand algorithmic efficiency. | 3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.<br><br>3B-AP-15: Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.<br><br>3A-AP-16: Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.<br><br>3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs. | AAP-3.B: Explain how the use of procedural abstraction manages complexity in a program.<br><br>CRD-2.B: Explain how a program or code segment functions.<br><br>AAP-3.A: For procedure calls: a. Write statements to call procedures. 3.B b. Determine the result or effect of a procedure call<br><br>AAP-3.C: Develop procedural abstractions to manage complexity in a program by writing procedures. | 7 hrs |

| Lesson Title | Lesson Objectives | CSTA Standards | AP CSP Learning Objective | Average Completion Time (in hours) |
|---|---|---|---|---|
| | | 3B-AP-14: Construct solutions to problems using student-created components, such as procedures, modules and/or objects. | | |
| Lesson 9: Algorithms | Learners will compare multiple algorithms to determine if they yield the same side effect or result.<br><br>Learners will create algorithms.<br><br>Learners will combine and modify existing algorithms. | 3B-AP-11: Evaluate algorithms in terms of their efficiency, correctness, and clarity.<br><br>3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects<br><br>3B-AP-10: Use and adapt classic algorithms to solve computational problems.<br><br>3B-AP-22: Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality). | AAP-2.L: Compare multiple algorithms to determine if they yield the same side effect or result<br><br>AAP-2.M: For algorithms: a. Create algorithms. B. Combine and modify exisiting algorithms. | 7 hrs |

| Lesson Title | Lesson Objectives | CSTA Standards | AP CSP Learning Objective | Average Completion Time (in hours) |
|---|---|---|---|---|
| Lesson 10: Collections | Learners will be able to identify, understand when to use and build code which uses lists, tuples, and dictionaries (creation, indexing, modifying, traversing, common methods) | 3A-AP-14: Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.<br><br>3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.<br><br>3A-AP-21: Evaluate and refine computational artifacts to make them more usable and accessible. | AAP-1.C: Represent a list or string using a variable. 3.A<br><br>AAP-1.D For data abstraction: a. Develop data abstraction using lists to store multiple elements 3.B<br><br>b. Explain how the use of data abstraction manages complexity in program code. 3.C<br><br>AAP-2.N For list operations: a. Write expressions that use list indexing and<br><br>list procedures. 2.B b. Evaluate expressions that use list indexing and list procedures. 4.B | 7 hrs |
| Lesson 11: Structure and Error Handling | Learners will be able to document code using comments and documentation stings. | 3A-AP-21: Evaluate and refine computational artifacts to make them more usable and accessible. | CRD-2.G: Describe the purpose of a code segment or program by writing documentation. | 7 hrs |

| Lesson Title | Lesson Objectives | CSTA Standards | AP CSP Learning Objective | Average Completion Time (in hours) |
|---|---|---|---|---|
| | Learners will be able to correctly use indentation to structure code appropriately.<br><br>Learners will describe the purpose of a code segment or program by writing documentation. | 3A-AP-23: Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.<br><br>3B-AP-20: Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (code documentation) in a group software project. | | |
| Lesson 12: Programming Practice | Learners will be able to identify and build code that uses variables.<br><br>Learners will be able to identify and use operators in code.<br><br>Learners will be able to identify and build code using complex selections and conditionals. | 3B-AP-10: Use and adapt classic algorithms to solve computational problems.<br><br>3B-AP-12: Identify and Use Operators in Python<br><br>3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. | AAP-1.A: Represent a value with a variable<br><br>AAP-2.E: For relationships between two variables, expressions, or values:<br><br>a. Write expressions using relational operators. b. Evaluate expressions that use relational operators. | |

| Lesson Title | Lesson Objectives | CSTA Standards | AP CSP Learning Objective | Average Completion Time (in hours) |
|---|---|---|---|---|
| | Learners will be able to identify and build loops and nested loops that use complex compound conditionals.<br><br>Learners will be able to identify, understand, call, and build functions<br><br>Learners will create algorithms.<br><br>Learners will combine and modify existing algorithms.<br><br>Learners will be able to identify, understand when to use and build code which uses lists | 3A-AP-13: Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.<br><br>3A-AP-15 Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.<br><br>3B-AP-15: Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.<br><br>3A-AP-16: Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions. | AAP-2.H: For selection: a. Write conditional statements b. Determine the result of conditional statements<br><br>AAP-2.I: For nested selection: a. Write nested conditional statements. b. Determine the result of nested conditional statements.<br><br>AAP-2.K For iteration: a. Write iteration statements. b. Determine the result or side effect of iteration statements.<br><br>AAP-3.B: Explain how the use of procedural abstraction manages complexity in a program.<br><br>CRD-2.B: Explain how a program or code segment functions. | |

| Lesson Title | Lesson Objectives | CSTA Standards | AP CSP Learning Objective | Average Completion Time (in hours) |
|---|---|---|---|---|
| | | 3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.<br><br>3B-AP-14: Construct solutions to problems using student-created components, such as procedures, modules and/or objects.<br><br>3A-AP-21: Evaluate and refine computational artifacts to make them more usable and accessible. | AAP-3.A: For procedure calls: a. Write statements to call procedures. 3.B b. Determine the result or effect of a procedure call<br><br>AAP-3.C: Develop procedural abstractions to manage complexity in a program by writing procedures.<br><br>AAP-2.L: Compare multiple algorithms to determine if they yield the same side effect or result<br><br>AAP-2.M: For algorithms: a. Create algorithms. B. Combine and modify existing algorithms.<br><br>AAP-1.C: Represent a list or string using a variable. 3.A | |

| Lesson Title | Lesson Objectives | CSTA Standards | AP CSP Learning Objective | Average Completion Time (in hours) |
|---|---|---|---|---|
| | | | AAP-1.D For data abstraction: a. Develop data abstraction using lists to store multiple elements 3.B<br><br>b. Explain how the use of data abstraction manages complexity in program code. 3.C<br><br>AAP-2.N For list operations: a. Write expressions that use list indexing and<br><br>list procedures. 2.B b. Evaluate expressions that use list indexing and list procedures. 4.B | |
| Lesson 13: Planning a Product | Learners will design a program and its user interface.<br><br>Learners will develop a program using a development process. | 3B-AP-17: Plan and develop programs for broad audiences using a software life cycle process.<br><br>3B-AP-21: Develop and use a series of test cases to verify that a program performs according to its design specifications. | CRD-2.E: Develop a program using a development process<br><br>CRD-2.F: Design a program and its user interface. | 8 hrs |

| Lesson Title | Lesson Objectives | CSTA Standards | AP CSP Learning Objective | Average Completion Time (in hours) |
|---|---|---|---|---|
| | | 3B-AP-23: Evaluate key qualities of a program through a process such as a code review.<br><br>3A-AP-19: Systematically design and develop programs for broad audiences by incorporating feedback from users. | | |
| Lesson 14: Building the Product | Learners will build a game using all the programming knowledge previously learned such as variables, operators, logic, loops, collections, algorithms, and functions. | 3B-AP-10: Use and adapt classic algorithms to solve computational problems.<br><br>3B-AP-12: Identify and Use Operators in Python<br><br>3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.<br><br>3A-AP-13: Create prototypes that use algorithms to solve computational | AAP-1.A: Represent a value with a variable<br><br>AAP-2.E: For relationships between two variables, expressions, or values:<br><br>a. Write expressions using relational operators. b. Evaluate expressions that use relational operators.<br><br>AAP-2.H: For selection: a. Write conditional statements b. Determine the result of conditional statements | 15 hrs |

| Lesson Title | Lesson Objectives | CSTA Standards | AP CSP Learning Objective | Average Completion Time (in hours) |
|---|---|---|---|---|
| | | problems by leveraging prior student knowledge and personal interests.<br><br>3A-AP-15 Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.<br><br>3B-AP-15: Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.<br><br>3A-AP-16: Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.<br><br>3A-AP-18: Create artifacts by using procedures within a program, combinations | AAP-2.I: For nested selection: a. Write nested conditional statements. b. Determine the result of nested conditional statements.<br><br>AAP-2.K For iteration: a. Write iteration statements. b. Determine the result or side effect of iteration statements.<br><br>AAP-3.B: Explain how the use of procedural abstraction manages complexity in a program.<br><br>CRD-2.B: Explain how a program or code segment functions.<br><br>AAP-3.A: For procedure calls: a. Write statements to call procedures. 3.B b. Determine the result or effect of a procedure call | |

| Lesson Title | Lesson Objectives | CSTA Standards | AP CSP Learning Objective | Average Completion Time (in hours) |
|---|---|---|---|---|
| | | of data and procedures, or independent but interrelated programs.<br><br>3B-AP-14: Construct solutions to problems using student-created components, such as procedures, modules and/or objects.<br><br>3A-AP-21: Evaluate and refine computational artifacts to make them more usable and accessible.<br><br>3B-AP-17: Plan and develop programs for broad audiences using a software life cycle process.<br><br>3B-AP-21: Develop and use a series of test cases to verify that a program performs according to its design specifications. | AAP-3.C: Develop procedural abstractions to manage complexity in a program by writing procedures.<br><br>AAP-2.L: Compare multiple algorithms to determine if they yield the same side effect or result<br><br>AAP-2.M: For algorithms: a. Create algorithms. B. Combine and modify existing algorithms.<br><br>AAP-1.C: Represent a list or string using a variable. 3.A<br><br>AAP-1.D For data abstraction: a. Develop data abstraction using lists to store multiple elements 3.B<br><br>b. Explain how the use of data abstraction manages complexity in program code. 3.C | |

| Lesson Title | Lesson Objectives | CSTA Standards | AP CSP Learning Objective | Average Completion Time (in hours) |
|---|---|---|---|---|
| | | 3B-AP-23: Evaluate key qualities of a program through a process such as a code review. | AAP-2.N For list operations: a. Write expressions that use list indexing and list procedures. 2.B b. Evaluate expressions that use list indexing and list procedures. 4.B<br><br>CRD-2.E: Develop a program using a development process<br><br>CRD-2.F: Design a program and its user interface. | |