Coding in Minecraft Expert Coding with JavaScript

Scope and Sequence – Georgia

Lesson Title	Lesson Objectives	AP Computer Science Principles	CSTA Standards
Lesson 1: Variables Average Completion time: Approximately 5 hours	Learners will understand the definition of variables. Learners will understand the different types of variables and when they should be used in code. Learners will be able to create variables and assign them values. Learners will understand the definition of computing and computing networks.	**CSN-1.A:** Explain how computing devices work together in a network **AAP-1.A:** Represent a value with a variable **AAP-1.B:** Determine the value of a variable because of an assignment. **CSN-1.A**: Explain how computing devices work together in a network	**3B-AP-12**: Compare and contrast fundamental data structures and their uses. **3A-AP-17:** Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. **3A-AP-21:** Evaluate and refine computational artifacts to make them more usable and accessible. **3B-AP-10**: Use and adapt classic algorithms to solve computational problems. **3B-CS-02:** Illustrate ways computing systems implement logic, input, and output through hardware components.

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	AP Computer Science Principles	CSTA Standards
Lesson 2: Operators Average Completion time: Approximately 5 hours	Learners will understand operators and how they are used in code. Learners will be able to identify and use operators in JavaScript. Learners will be able to evaluate expressions that use arithmetic operators. Learners will be able to write expressions using relational operators. Learners will be able to evaluate expressions that use relational operators.	**AAP-2.E:** For relationships between two variables, expressions, or values: a. Write expressions using relational operators. b. Evaluate expressions that use relational operators.	**3B-AP-12**: Compare and contrast fundamental data structures and their uses. **3B-CS-02:** Illustrate ways computing systems implement logic, input, and output through hardware components.
Lesson 3: Logic Average Completion time: Approximately 8 hours	Learners will understand what a conditional is and why it is used in code. Learners will learn what an If statement is and how to use it in their code. Learners will learn what an If / Else statement is and how to use it in their code.	**AAP-2.G:** Express an algorithm that uses selection without using a programming language. **AAP-2.H:** For selection: a. Write conditional statements b. Determine the result of conditional statements.	**3A-NI-04:** Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing. **3B-AP-10:** Use and adapt classic algorithms to solve computational problems.

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	AP Computer Science Principles	CSTA Standards
			3A-AP-13: Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests. **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. **3A-AP-17**: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. **3A-AP-21**: Evaluate and refine computational artifacts to make them more usable and accessible.

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	AP Computer Science Principles	CSTA Standards
Lesson 4: Advanced Logic Average Completion time: Approximately 8 hours	Learners will learn what an If / Else If / Else statement is and why it is used in code. Learners will learn how to use an If / Else If / Else statement in their code. Learners will learn what nested conditionals are and why they are used in code. Learners will learn how to use nest conditionals in their code. Learners will learn what the different types of computing are including their strengths, weaknesses, and when to use them.	**AAP-2.G:** Express an algorithm that uses selection without using a programming language. **AAP-2.H:** For selection: a. Write conditional statements b. Determine the result of conditional statements. **AAP-2.I:** For nested selection: a. Write nested conditional statements. b. Determine the result of nested conditional statements.	**3A-CS-02**: Compare levels of abstraction and interactions between application software, system software, and hardware layers **3A-AP-13**: Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests. **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. **3A-AP-17:** Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	AP Computer Science Principles	CSTA Standards
			3A-AP-21: Evaluate and refine computational artifacts to make them more usable and accessible. **3B-AP-10**: Use and adapt classic algorithms to solve computational problems.
Lesson 5: Loops Average Completion time: Approximately 8 hours	Learners will be able to identify and build code which uses while loop statements. Learners will be able to identify and build code which uses for loop statements. Learners will be able to identify and build code which uses nested loops.	**AAP-2.K** For iteration: a. Write iteration statements. b. Determine the result or side effect of iteration statements.	**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. **3A-AP-21:** Evaluate and refine computational artifacts to make them more usable and accessible. **3B-AP-10**: Use and adapt classic algorithms to solve computational problems.

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	AP Computer Science Principles	CSTA Standards
Lesson 6: Coding Practice Average Completion time: Approximately 8 hours	Learners will be able to identify and build code that uses variables. Learners will be able to identify and use operators in code. Learners will be able to identify and build code using complex selections and conditionals. Learners will be able to identify and build loops and nested loops that use complex compound conditionals.	**AAP-1.A:** Represent a value with a variable **AAP-2.E:** For relationships between two variables, expressions, or values: a. Write expressions using relational operators. b. Evaluate expressions that use relational operators. **AAP-2.H:** For selection: a. Write conditional statements b. Determine the result of conditional statements **AAP-2.I:** For nested selection: a. Write nested conditional statements. b. Determine the result of nested conditional statements. **AAP-2.K** For iteration: a. Write iteration statements. b. Determine the result or side effect of iteration statements.	**3B-AP-10:** Use and adapt classic algorithms to solve computational problems. **3B-AP-12**: Identify and Use Operators in JavaScript **3A-AP-17**: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. **3A-AP-13**: Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests. **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.

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	AP Computer Science Principles	CSTA Standards
Lesson 7: Project Average Completion time: Approximately 10 hours	Learners will build a game using all the programming knowledge previously learned such as variables, operators, logic, and loops.	**AAP-2.E:** For relationships between two variables, expressions, or values: a. Write expressions using relational operators. b. Evaluate expressions that use relational operators. **AAP-2.H:** For selection: a. Write conditional statements b. Determine the result of conditional statements **AAP-2.I:** For nested selection: a. Write nested conditional statements. b. Determine the result of nested conditional statements. **AAP-2.K** For iteration: a. Write iteration statements. b. Determine the result or side effect of iteration statements.	3B-AP-12: Identify and Use Operators in JavaScript 3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. 3A-AP-13: Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests. 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.

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	AP Computer Science Principles	CSTA Standards
Lesson 8: Functions Average Completion time: Approximately 7 hours	Learners will be able to understand the concept of modularity and identify when modularity can or should be used in code Learners will be able to identify, understand, call, and build functions Learners will be able to identify, understand, call and build functions that take parameters or return data. Learners will be able to identify and understand algorithmic efficiency.	**AAP-3.B:** Explain how the use of procedural abstraction manages complexity in a program. **CRD-2.B:** Explain how a program or code segment functions. **AAP-3.A:** For procedure calls: a. Write statements to call procedures. 3.B b. Determine the result or effect of a procedure call **AAP-3.C:** Develop procedural abstractions to manage complexity in a program by writing procedures.	**3A-AP-17**: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. **3B-AP-15:** Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution. **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. **3A-AP-18:** Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs. **3B-AP-14:** Construct solutions to problems using student-created

© Prodigy Learning | prodigylearning.com | codingcredentials.com

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

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	AP Computer Science Principles	CSTA Standards
Lesson 10: Collections Average Completion time: Approximately 7 hours	Learners will be able to identify, understand when to use and build code which uses lists (creation, indexing, modifying, traversing, common methods)	**AAP-1.C:** Represent a list or string using a variable. 3.A **AAP-1.D** For data abstraction: a. Develop data abstraction using lists to store multiple elements 3.B b. Explain how the use of data abstraction manages complexity in program code. 3.C **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	**3A-AP-14:** Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables. **3A-AP-17:** Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. **3A-AP-21**: Evaluate and refine computational artifacts to make them more usable and accessible.
Lesson 11: Structure and Error Handling Average Completion time: Approximately 7 hours	Learners will be able to document code using comments and documentation stings. Learners will be able to correctly use indentation to structure code appropriately.	**CRD-2.G:** Describe the purpose of a code segment or program by writing documentation.	**3A-AP-21**: Evaluate and refine computational artifacts to make them more usable and accessible. **3A-AP-23**: Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	AP Computer Science Principles	CSTA Standards
	Learners will describe the purpose of a code segment or program by writing documentation.		**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 Average Completion time: Approximately 5 hours	Learners will be able to identify and build code that uses variables. Learners will be able to identify and use operators in code. Learners will be able to identify and build code using complex selections and conditionals. Learners will be able to identify and build loops and nested loops that use complex compound conditionals. Learners will be able to identify, understand, call, and build functions Learners will create algorithms.	**AAP-1.A:** Represent a value with a variable **AAP-2.E:** For relationships between two variables, expressions, or values: a. Write expressions using relational operators. b. Evaluate expressions that use relational operators. **AAP-2.H:** For selection: a. Write conditional statements b. Determine the result of conditional statements **AAP-2.I**: For nested selection: a. Write nested conditional statements. b. Determine the result of nested conditional statements.	**3B-AP-10:** Use and adapt classic algorithms to solve computational problems. **3B-AP-12**: Identify and Use Operators in JavaScript **3A-AP-17**: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. **3A-AP-13**: Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests. **3A-AP-15** Justify the selection of specific control structures when tradeoffs involve implementation,

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	AP Computer Science Principles	CSTA Standards
	Learners will combine and modify existing algorithms. Learners will be able to identify, understand when to use and build code which uses lists	**AAP-2.K** For iteration: a. Write iteration statements. b. Determine the result or side effect of iteration statements. **AAP-3.B:** Explain how the use of procedural abstraction manages complexity in a program. **CRD-2.B:** Explain how a program or code segment functions. **AAP-3.A:** For procedure calls: a. Write statements to call procedures. 3.B b. Determine the result or effect of a procedure call **AAP-3.C:** Develop procedural abstractions to manage complexity in a program by writing procedures. **AAP-2.L:** Compare multiple algorithms to determine if they yield the same side effect or result	readability, and program performance, and explain the benefits and drawbacks of choices made. **3B-AP-15**: Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution. **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. **3A-AP-18**: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs. **3B-AP-14**: Construct solutions to problems using student-created components, such as procedures, modules and/or objects.

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	AP Computer Science Principles	CSTA Standards
		AAP-2.M: For algorithms: a. Create algorithms. B. Combine and modify existing algorithms. **AAP-1.C:** Represent a list or string using a variable. 3.A **AAP-1.D** For data abstraction: a. Develop data abstraction using lists to store multiple elements 3.B b. Explain how the use of data abstraction manages complexity in program code. 3.C **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	**3A-AP-21:** Evaluate and refine computational artifacts to make them more usable and accessible.
Lesson 13: Planning a Product	Learners will design a program and its user interface.	**CRD-2.E:** Develop a program using a development process	**3B-AP-17:** Plan and develop programs for broad audiences using a software life cycle process.

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	AP Computer Science Principles	CSTA Standards
Average Completion time: Approximately 8 hours	Learners will develop a program using a development process.	**CRD-2.F:** Design a program and its user interface.	**3B-AP-21**: Develop and use a series of test cases to verify that a program performs according to its design specifications. **3B-AP-23:** Evaluate key qualities of a program through a process such as a code review. **3A-AP-19:** Systematically design and develop programs for broad audiences by incorporating feedback from users.
Lesson 14: Building the Product Average Completion time: Approximately 15 hours	Learners will build a game using all the programming knowledge previously learned such as variables, operators, logic, loops, collections, algorithms, and functions.	**AAP-1.A**: Represent a value with a variable **AAP-2.E:** For relationships between two variables, expressions, or values: a. Write expressions using relational operators. b. Evaluate expressions that use relational operators. **AAP-2.H:** For selection: a. Write conditional statements b. Determine the result of conditional statements	3B-AP-10: Use and adapt classic algorithms to solve computational problems. 3B-AP-12: Identify and Use Operators in JavaScript 3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	AP Computer Science Principles	CSTA Standards
		AAP-2.I: For nested selection: a. Write nested conditional statements. b. Determine the result of nested conditional statements. **AAP-2.K** For iteration: a. Write iteration statements. b. Determine the result or side effect of iteration statements. **AAP-3.B:** Explain how the use of procedural abstraction manages complexity in a program. **CRD-2.B:** Explain how a program or code segment functions. **AAP-3.A:** For procedure calls: a. Write statements to call procedures. 3.B b. Determine the result or effect of a procedure call **AAP-3.C:** Develop procedural abstractions to manage complexity in a program by writing procedures.	3A-AP-13: Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests. 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. 3B-AP-15: Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution. 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. **3A-AP-18:** Create artifacts by using procedures within a program, combinations of data and procedures,

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	AP Computer Science Principles	CSTA Standards
		AAP-2.L: Compare multiple algorithms to determine if they yield the same side effect or result **AAP-2.M:** For algorithms: a. Create algorithms. B. Combine and modify existing algorithms. **AAP-1.C:** Represent a list or string using a variable. 3.A **AAP-1.D** For data abstraction: a. Develop data abstraction using lists to store multiple elements 3.B b. Explain how the use of data abstraction manages complexity in program code. 3.C **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	or independent but interrelated programs. **3B-AP-14:** Construct solutions to problems using student-created components, such as procedures, modules and/or objects. **3A-AP-21:** Evaluate and refine computational artifacts to make them more usable and accessible. **3B-AP-17:** Plan and develop programs for broad audiences using a software life cycle process. **3B-AP-21:** Develop and use a series of test cases to verify that a program performs according to its design specifications. **3B-AP-23**: Evaluate key qualities of a program through a process such as a code review.

© Prodigy Learning | prodigylearning.com | codingcredentials.com

Lesson Title	Lesson Objectives	AP Computer Science Principles	CSTA Standards
		CRD-2.E: Develop a program using a development process **CRD-2.F:** Design a program and its user interface.	

© Prodigy Learning | prodigylearning.com | codingcredentials.com