Coding in Minecraft - Python Expert Coding – Exam Objective Domain

ID	Exam & Learning Objectives	Evidence	NC CS Standards Alignment	AP CSP Learning Objectives Alignment
1	**Variables & Data Types**			
1.1	Identify and use variables in Python and assign appropriate data types	Be able to define a variable using Python Be able to choose an appropriate data type of str, int, float or bool and explain why	HS-AP-04 Select an appropriate data structure for information of a given problem.	AAP-1.A Represent a value with a variable. 3.A
1.2	Convert between data types	Be able to convert from one data type to another	HS-AP-04 Select an appropriate data structure for information of a given problem.	AAP-1.B Determine the value of a variable because of an assignment. 4.B
1.3	Identify and use Operators	Be able to identify and use Operators in Python - assignment, comparison, logical, arithmetic, is and in Be able to determine operator precedence and sequence of execution	HS-AP-04 Select an appropriate data structure for information of a given problem.	AAP-2.C Evaluate expressions that use arithmetic operators. 4.B AAP-2.E For relationships between two variables, expressions, or values: a. Write expressions using relational operators. 2.B b. Evaluate expressions that use relational operators. 4.B

© Prodigy Learning | prodigylearning.com | codingcredentials.com

ID	Exam & Learning Objectives	Evidence	NC CS Standards Alignment	AP CSP Learning Objectives Alignment
2	**Selection/Branching - If Statements**			
2.1	Identify and build If statements	Be able to identify and build code using if statements	ICS-AP-04 Justify the selection of specific control structures, considering implementation, readability, and program performance.	AAP-2.H For selection: a. Write conditional statements. 2.B b. Determine the result of conditional statements. 4.B
2.2	Identify and build complex if statements	Be able to identify and build code using elif and else statements to create complex selections	ICS-AP-04 Justify the selection of specific control structures, considering implementation, readability, and program performance.	AAP-2.I For nested selection: a. Write nested conditional statements 2.B b. Determine the result of nested conditional statements. 4.B
2.3	Identify and use If statements that contain compound conditionals	Be able to identify and build code using complex compound conditionals (Or, And) and Not operator	ICS-AP-04 Justify the selection of specific control structures, considering implementation, readability, and program performance.	AAP-2.F For relationships betweenBoolean values: a. Write expressions using logical operators. 2.Bb. Evaluate expressions that use logic operators. 4.B

© Prodigy Learning | prodigylearning.com | codingcredentials.com

ID	Exam & Learning Objectives	Evidence	NC CS Standards Alignment	AP CSP Learning Objectives Alignment
3	**Iteration - Loops**			
3.1	Identify and build while loops	Be able to identify and build code which uses while loop statements	ICS-AP-05 Iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.	AAP-2.K For iteration: a. Write iteration statements. 2.B b. Determine the result or side effect of iteration statements. 4.B
3.2	Identify and build for loops	Be able to identify and build code which uses for loop statements	ICS-AP-05 Iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.	AAP-2.K For iteration: a. Write iteration statements. 2.B b. Determine the result or side effect of iteration statements. 4.B
3.3	Identify and build nested loops	Be able to identify and build code which uses nested loops	ICS-AP-05 Iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.	AAP-2.K For iteration: a. Write iteration statements. 2.B b. Determine the result or side effect of iteration statements. 4.B
3.4	Identify and build loops that use complex compound conditionals	Be able to identify and build loops that use complex compound conditionals (Or, And, Not)	ICS-AP-05 Iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.	AAP-2.F For relationships between Boolean values: a. Write expressions using logical operators. 2.B b. Evaluate expressions that use logic operators. 4.B
3.5	Identify and use loop controls	Be able to identify, understand and implement continue, break, pass statements in loops	HS-AP-05 Illustrate the flow of execution of a recursive algorithm	AAP-2.K For iteration: a. Write iteration statements. 2.B b. Determine the result or side effect of iteration statements. 4.B

© Prodigy Learning | prodigylearning.com | codingcredentials.com

ID	Exam & Learning Objectives	Evidence	NC CS Standards Alignment	AP CSP Learning Objectives Alignment
4	**Collections**			
4.1	Identify and use lists	Be able to identify, understand when to use and build code which uses lists (creation, indexing, modifying, traversing, common methods)	ICS-AP-03 Utilize lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables. ICS-AP-06 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. ICS-AP-07 Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.	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
4.2	Identify and use tuples	Be able to identify, understand when to use and build code which uses tuples (creation, unpacking, iterating, common methods)	ICS-AP-03 Utilize lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables. ICS-AP-06 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. ICS-AP-07 Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.	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

© Prodigy Learning | prodigylearning.com | codingcredentials.com

				b. Evaluate expressions that use list indexing and list procedures. 4.B
4.3	Identify and use dictionaries	Be able to identify, understand when to use and build code which uses dictionaries (creation, modifying, membership, iteration, common methods)	HS-AP-08 Create computational artifacts with pre-existing procedures, external components, libraries and APIs.	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

© Prodigy Learning | prodigylearning.com | codingcredentials.com

ID	Exam & Learning Objectives	Evidence	NC CS Standards Alignment	AP CSP Learning Objectives Alignment
5	**Structure and document code appropriately**			
5.1	Use whitespace and indentation	Be able to correctly use whitespace and indentation to structure code appropriately	HS-AP-12 Compose standard documentation for computational artifacts to make it easier to follow, test, and debug.	CRD-2.G Describe the purpose of a code segment or program by writing documentation. 4.A
5.2	Document code	Be able to document code using comments and documentation stings and generate documentation using pydoc	HS-AP-12 Compose standard documentation for computational artifacts to make it easier to follow, test, and debug.	CRD-2.G Describe the purpose of a code segment or program by writing documentation. 4.A CRD-2.H Acknowledge code segments used from other sources. 1.C
5.3	Code review	Be able to perform a code review of another person's code	HS-AP-12 Compose standard documentation for computational artifacts to make it easier to follow, test, and debug.	

© Prodigy Learning | prodigylearning.com | codingcredentials.com

ID	Exam & Learning Objectives	Evidence	NC CS Standards Alignment	AP CSP Learning Objectives Alignment
6	**Error handling**			
6.1	Understand the different types of errors and causes	Be able to identify, understand, differentiate, and perform basic troubleshooting procedures and fix syntax errors, semantic/logic error, and runtime errors	HS-AP-10 Justify that a computational artifact meets design specifications with systematic testing and debugging methods	CRD-2.I For errors in an algorithm or program: a. Identify the error. 4.C b. Correct the error. 4.C
6.2	Understand code testing and create tests	Be able to understand the need for and importance of testing and be able to create tests to validate code	HS-AP-10 Justify that a computational artifact meets design specifications with systematic testing and debugging methods	CRD-2.J Identify inputs and corresponding expected outputs or behaviours that can be used to check the correctness of an algorithm or program. 4.C
6.3	Identify and build code that handles errors and exceptions	Be able to identify, understand when to use and build code which uses exception handling - try, else, finally, raise, except	HS-AP-10 Justify that a computational artifact meets design specifications with systematic testing and debugging methods	

© Prodigy Learning | prodigylearning.com | codingcredentials.com

ID	Exam & Learning Objectives	Evidence	NC CS Standards Alignment	AP CSP Learning Objectives Alignment
7	**Modularity**			
7.1	Understand modularity and identify when to use	Be able to understand the concept of modularity and identify when modularity can or should be used in code	ICS-AP-06 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. HS-AP-07 Analyze general patterns applicable to a solution.	AAP-3.B Explain how the use of procedural abstraction manages complexity in a program. 3.C CRD-2.B Explain how a program or code segment functions. 4.A
7.2	Identify and build functions	Be able to identify, understand, use (call) and build functions - def statement	ICS-AP-06 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. HS-AP-07 Analyze general patterns applicable to a solution. HS-AP-08 Create computational artifacts with pre-existing procedures, external components, libraries and APIs.	AAP-3.A For procedure calls: a. Write statements to call procedures. 3.B b. Determine the result or effect of a procedure call. 4.B AAP-3.C Develop procedural abstractions to manage complexity in a program by writing procedures. 3.B
7.3	Identify and build functions that take or return data	Be able to identify, understand, use (call) and build functions that take parameters or return data - passing parameters/arguments (inc optional and keyword arguments) and return statement	ICS-AP-06 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. HS-AP-07 Analyze general patterns applicable to a solution.	AAP-3.A For procedure calls: a. Write statements to call procedures. 3.B b. Determine the result or effect of a procedure call. 4.B AAP-3.C Develop procedural abstractions to manage complexity in a program by writing procedures. 3.B

© Prodigy Learning | prodigylearning.com | codingcredentials.com

			HS-AP-08 Create computational artifacts with pre-existing procedures, external components, libraries and APIs.	

© Prodigy Learning | prodigylearning.com | codingcredentials.com

ID	Exam & Learning Objectives	Evidence	NC CS Standards Alignment	AP CSP Learning Objectives Alignment
8	**Input and Output**			
8.1	Identify and build code that performs console input operations	Be able to identify and build code which reads input from the console and stores or processes appropriately	HS-CS-03 Illustrate the ways computing systems implement logic, input, and output through hardware components.	CRD-2.J Identify inputs and corresponding expected outputs or behaviors that can be used to check the correctness of an algorithm or program. CRD-2.B Explain how a program or code segment functions.
8.2	Identify and build code that performs console output operations	Be able to identify and build code that outputs to the console and format output appropriately	HS-CS-03 Illustrate the ways computing systems implement logic, input, and output through hardware components.	CRD-2.J Identify inputs and corresponding expected outputs or behaviors that can be used to check the correctness of an algorithm or program. CRD-2.B Explain how a program or code segment functions.
8.3	Identify and build code that performs file input operations	Be able to identify and build code that opens, reads, and closes text files including using the with statement	HS-CS-03 Illustrate the ways computing systems implement logic, input, and output through hardware components.	CRD-2.J Identify inputs and corresponding expected outputs or behaviors that can be used to check the correctness of an algorithm or program. CRD-2.B Explain how a program or code segment functions.

© Prodigy Learning | prodigylearning.com | codingcredentials.com