CodeWizardsHQ
Essentials

Python Scope and Sequence / Quarter 1

Chapter 2

LESSON	LESSON TYPE	BIG IDEAS / TOPICS / CONCEPTS	STANDARDS (CSTA)	STANDARDS (CSTA)	LEARNING OBJECTIVES
Lesson 1	Teaching	Data Types Mathematical Operations	2-AP-13	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	TLW: Differentiate between different types of data in their programs.
			2-AP-17	Systematically test and refine programs using a range of test cases.	TLW: Understand how to perform mathematical operations in their code and display the results.
Lesson 2	Teaching	Variables F-Strings	2-AP-11	Create clearly named variables that represent different data types and perform operations on their values.	TLW: Understand variables as containers that store changing values.
			1B-AP-09	Create programs that use variables to store and modify data.	TLW: Be familiar with creating, updating, and using variables in programs.
			1B-AP-15	Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.	
Lesson 3	Teaching	Raw Strings ASCII Art Unicode Multi-line Strings	2-AP-13	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	TLW: Be familiar with using differing types of strings (raw, multi-line, and f).
			2-AP-17	Systematically test and refine programs using a range of test cases.	TLW: Revise their "About Me" paragraph using an f-string.
			2-AP-11	Create clearly named variables that represent different data types and perform operations on their values.	TLW: Print ASCII art and unicode symbols inside of multi-line strings.

Chapter 2

LESSON	LESSON TYPE	BIG IDEAS / TOPICS / CONCEPTS	STANDARDS (CSTA)	STANDARDS (CSTA)	LEARNING OBJECTIVES
Lesson 4	Project	Receipt Printer	2-AP-13	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	TLW: Review their understanding of data types, variables and strings, and mathematical operations as they complete the culminating project.
			2-AP-17	Systematically test and refine programs using a range of test cases.	TLW: Create variables to represent each item and its price and calculate the total cost of a shopping trip.
			2-AP-11	Create clearly named variables that represent different data types and perform operations on their values.	TLW: Focus on format as they display these values in the final receipt.
			1B-AP-12	Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.	

Chapter 3

Lesson 1	Teaching	Function Calls Functions With Parameters Using the wizardlib Library	2-AP-13	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	TLW: To display sentences about their favorite things, make a score, and create username and password prompt.
Lesson 2	Teaching	Mathematical Operators (Power, Modulus, Floor)	1B-AP-09	Create programs that use variables to store and modify data.	TLW: Set up an "About Me" page with a focus on saving what the user types into text boxes.
			2-AP-16	Incorporate existing code, media, and libraries into original programs, and give attribution.	TLW: Create working buttons that react to mouse click.
Lesson 3	Project	Python Calculator	3A-AP-13	Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.	TLW: Build a functioning calculator in Python capable of adding, subtracting, multiplying, diving, power, and floor.

	LESSON	LESSON TYPE	BIG IDEAS / TOPICS / CONCEPTS	STANDARDS (CSTA)	STANDARDS (CSTA)	LEARNING OBJECTIVES
Chapter 4	Lesson 1	Teaching	If / Else Mathematical Comparisons Importing Random Library	1B-AP-10	Create programs that include sequences, events, loops, and conditionals.	TLW: Understand conditional statements in Python: if, elif and else.
				2-AP-12	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	TLW: Understand the definition of conditional statements, why they are so important, and how to use them.
	Lesson 2	Teaching	Elif	1B-AP-10	Create programs that include sequences, events, loops, and conditionals.	TLW: Understand the "greater than or equal to" and "less than or equal to" operators.
				2-AP-12	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	TLW: Understand the "elif" statement.
	Lesson 3	Project	Racing Game	1B-AP-10	Create programs that include sequences, events, loops, and conditionals.	TLW: Apply an understanding of conditional statements to create a racing game project.
				2-AP-12	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	

Chapter 5	Lesson 1	Teaching	Nested If Statements AND / OR	1B-AP-10	Create programs that include sequences, events, loops, and conditionals.	TLW: Understand how to build complex conditional statements that accurately simulate real-world decisions.
				2-AP-12	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	TLW: Learn how to build on conditional statements with the keywords "and" and "or".

	LESSON	LESSON TYPE	BIG IDEAS / TOPICS / CONCEPTS	STANDARDS (CSTA)	STANDARDS (CSTA)	LEARNING OBJECTIVES
Chapter 5	Lesson 2	Teaching	NOT / !=	3A-AP-17	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.	TLW: Understand how to build on conditional statements with the keyword "not" and the "not equal to" operator represented as "!=".
	Lesson 3	Project	Rise of the Robots	2-DA-09	Refine computational models based on the data they have generated.	TLW: Understand how to elaborate on conditional statements with an, or, not, and !=.
				2-AP-11	Create clearly named variables that represent different data types and perform operations on their values.	TLW: Be comfortable with if, elif, and else statements as a whole.
Chapter 6	Lesson 1	Teaching	Intro to Loops While Loops Break / Continue	1B-AP-10	Create programs that include sequences, events, loops, and conditionals.	TLW: Understand what a loop is as it relates to programming and the problems it can solve.
				2-AP-12	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	TLW: Understand that loops are used to repeat segments of code.
	Lesson 2	Teaching	For Loops Range( )	1B-AP-10	Create programs that include sequences, events, loops, and conditionals.	TLW: Understand how to use "for" loops as well as know when to use a "for" loop.
				2-AP-12	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	TLW: Know how to make a counter, break apart strings, and use the "break" and "continue" keywords with a "for" loop.
	Lesson 3	Project	Fizz Buzz	1B-AP-10	Create programs that include sequences, events, loops, and conditionals.	TLW: Understand how to use loops to generate a counter as well as how to change various iterations of the loop cycle with conditionals.
				2-AP-12	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	TLW: Make use of the break and continue keywords.
				3A-AP-13	Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.	TLW: Understand how to keep track of and display the number of times a certain word appears.

Chapter 7

LESSON	LESSON TYPE	BIG IDEAS / TOPICS / CONCEPTS	STANDARDS (CSTA)	STANDARDS (CSTA)	LEARNING OBJECTIVES
Lesson 1	Teaching	User-Defined Functions Return Statement	2-AP-14	Create procedures with parameters to organize code and make it easier to reuse.	TLW: Become acquainted with making functions instead of using pre-existing wizardlib and Python functions.
			2-AP-19	Document programs in order to make them easier to follow, test, and debug.	TLW: Practice making functions with exercises that walk through the creation of a large variety of functions.
Lesson 2	Teaching	Variable Scope Global Versus Local	2-AP-14	Create procedures with parameters to organize code and make it easier to reuse.	TLW: Understand variable scope and the differences in what is stored in a variable depending on whether it's in a function or not.
			2-AP-19	Document programs in order to make them easier to follow, test, and debug.	TLW: Be comfortable using the global keyword.
Lesson 3	Teaching	User-Defined Functions with Parameters	2-AP-14	Create procedures with parameters to organize code and make it easier to reuse.	TLW: Learn to incorporate parameters, learn how to work with them when defining a function, and learn about different types of arguments.
Lesson 4	Project	Rock, Paper, Scissors	2-AP-12	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	TLW: Understand how to use a combination of student-created functions, existing Python, and wizardlib functions together to create something fun or practical.
			2-AP-14	Create procedures with parameters to organize code and make it easier to reuse.	TLW: Make a rock, paper, scissors game that applies chapter learnings.
			3A-AP-13	Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.	

Chapter 8

LESSON	LESSON TYPE	BIG IDEAS / TOPICS / CONCEPTS	STANDARDS (CSTA)	STANDARDS (CSTA)	LEARNING OBJECTIVES
Lesson 1	Project	Escape the Dungeon Capstone: Part 1	3A-AP-13	Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.	TLW: Work on creating a dungeon map and programming the code for each room.
Lesson 2	Project	Escape the Dungeon Capstone: Part 2	3A-AP-13	Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.	TLW: How to add locked doors with keys to dungeon map.
			3A-AP-17	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.	TLW: Fix issues with re-entering rooms for dungeon map.
Lesson 3	Project	Escape the Dungeon Capstone: Part 3	3A-AP-13	Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.	TLW: Finish dungeon map game.
			3A-AP-17	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.	