Massachusetts Standards Alignment with Python with Robots Curriculum

Computational Thinking	Unit 1	Unit 2	Unit 3	Unit 4
6-8.CT.a Abstraction				
6-8.CT.a.1 Describe how data is abstracted by listing attributes of everyday items to represent, order and compare those items (e.g., street address as an abstraction for locations; car make, model, and license plate number as an abstraction for cars).				
6-8.CT.a.2 Define a simple function that represents a more complex task/problem and can be reused to solve similar tasks/problems.				
6-8.CT.a.3 Use decomposition to define and apply a hierarchical classification scheme to a complex system, such as the human body, animal classification, or in computing.				
6-8.CT.b Algorithms				
6-8.CT.b.1 Design solutions that use repetition and conditionals.				
6-8.CT.b.2 Use logical reasoning to predict outputs given varying inputs.				
6-8.CT.b.3 Individually and collaboratively decompose a problem and create a sub-solution for each of its parts (e.g., video game, robot obstacle course, making dinner).				
6-8.CT.b.4 Recognize that more than one algorithm can solve a given problem.				
6-8.CT.b.5 Recognize that boundaries need to be taken into account for an algorithm to produce correct results.				
6-8.CT.c Data				
6-8.CT.c.1 Demonstrate that numbers can be represented in different base systems (e.g., binary, octal, and hexadecimal) and text can be represented in different ways (e.g., American Standard Code for Information Interchange [ASCII]).				
6-8.CT.c.2 Describe how computers store, manipulate, and transfer data types and files (e.g., integers, real numbers, Boolean Operators) in a binary system.				
6-8.CT.c.3 Create, modify, and use a database (e.g., define field formats, add new records, manipulate data), individually and collaboratively, to analyze data and propose solutions for a task/problem.				
6-8.CT.c.4 Perform a variety of operations such as sorting, filtering, and searching in a database to organize and display information in a variety of ways such as number formats (scientific notation and percentages), charts, tables, and graphs.				
6-8.CT.c.5 Select and use data-collection technology (e.g., probes, handheld devices, geographic mapping systems) to individually and collaboratively gather, view, organize, analyze, and report results for content-related problems.				
6-8.CT.d Programming and Development				
6-8.CT.d.1 Individually and collaboratively compare algorithms to solve a problem, based on a given criteria (e.g., time, resource, accessibility).				
6-8.CT.d.2 Use functions to hide the detail in a program.				
6-8.CT.d.3 Create a program, individually and collaboratively, that implements an algorithm to achieve a given goal.				
6-8.CT.d.4 Implement problem solutions using a programming language, including all of the following: looping behavior, conditional statements, expressions, variables, and functions.				
6-8.CT.d.5 Trace programs step-by-step in order to predict their behavior.				
6-8.CT.d.6 Use an iterative approach to development and debugging to understand the dimensions of a problem clearly.				
6-8.CT.e Modeling and Simulation				
6-8.CT.e.1 Create a model of a real-world system and explain why some details, features and behaviors were required in the model and why some could be ignored.				
6-8.CT.e.2 Use and modify simulations to analyze and illustrate a concept in depth (e.g., light rays/mechanical waves interaction with materials, genetic variation).				
6-8.CT.e.3 Select and use computer simulations, individually and collaboratively, to gather, view, analyze, and report results for content-related problems (e.g., migration, trade, cellular function).				