2nd Semester - Introduction to Computer Science Curriculum Map

Unit Order Considerations

As you implement this curriculum into your program, there are some things to consider when deciding what units best fits program goals.

- **Preparing students to take Advanced Placement Computer Science A** - It is recommended completing the units in the following order 1, 2, 3, 4, 7, 6, 8, 5(Optional)
- **Preparing students to gain a credential such as the Microsoft Technology Associate (MTA) certification** - It is recommended completing the course units in the following order 1, 2, 3, 4, Additional Topics. 5(Optional)
- **Offering this course as a stand alone computer science course** - It is recommended completing the course units in the following order - 1, 2, 3, 4, 5, 6, 7, 8

Unit 1 - Introduction to Python

Lesson	Objectives	Lab
1.01: Set Up	Define and identify: IDE, Python. Identify the key concepts that will be covered in the course. Set up and log into an account for the course's online IDE. Save and turn in a file via the online IDE.	N/A
1.02: Interactive Mode	Define and identify: interpreter, string, integer, float, value, errors, console, expression. Use the Python interpreter to evaluate simple math expressions. Distinguish between an integer, float, and string.	Using the Interpreter
1.03: Script Mode and Variables	Define and identify: script, print, run, output, variable. Write a simple script and run it in the IDE. Print values out to the console (both composed values and from variables). Compare script mode vs interactive mode. Know how to store a value into a variable.	Printing & Variables
1.04: Variables Input	Define and identify: comments, storing, mutability, variable assignment, input. Assign and swap variables. Store user input into a variable.	Magic Genie
1.05: Quiz & Debugging	Demonstrate their understanding of key concepts covered up to this point. Define and identify: debugging, syntax errors. Analyze and respond to error messages.	N/A
1.06: MadLibs	Apply basic Python knowledge about inputs/outputs and variables to create a game of Madlibs. Practice good debugging skills.	Mad Libs

Unit 2 - Data Types and Conditionals

Lesson	Objectives	Lab
2.01: Data Types & Casting	Define and identify: type, string, casting, floating point number (float), integer. Describe different representations of data in Python. Convert from one data type to another data type.	Casting
2.02: Booleans & Expressions	Define and identify: Boolean, expression, composition, True, False. Evaluate a Boolean expression. Compose Boolean expressions using and, or, not, <, >, and ==.	Can I or Can't I?

Lesson	Objectives	Lab
2.03: Conditionals	Define and identify: if, else, `elif`, conditionals, flow of control. Create chaining if statements. Understand how conditional statements alter the flow of control of a program.	Game Show
2.04: Lists	Define and identify: list, item, index, integer. Be able to access items from a list using the index. Create lists of different types. Use the length function.	College Chooser
2.05: Lists 2	Define and identify: slice, append, pop, remove. Slice a list. Add and remove elements from a list	Tic-Tac-Toe
2.06: Game Loop	Define and identify: while loop. Use a while loop to simulate game play.	Tic-Tac-Toe Revisited
2.07: Text Game	Use knowledge of lists, Booleans, conditionals, and while loops to create a text-based adventure game.	Text Monster Game

Unit 3 - Functions

Lesson	Objectives	Lab
3.01: Built In Functions	Define and identify: function, arguments, calling, importing, returning. Call the built-in randint function, using arguments. Utilize code other people have written in the Python documentation. Understand the difference between printing and returning.	Magic 8-Ball
3.02: User-Defined Functions	Define and identify: abstraction, def. Create functions.	Birthday Song & Random Cards
3.03: Return vs Print	Define and identify: return, none, void. Explain and demonstrate the difference between printing and returning.	War (Card Game)
3.04: Debugging and Scope	Define and identify: scope, aliasing, stack trace. Understand that changing a list in a function updates the list outside of the function. Understand that updating variables in a function does not affect the variable outside of the function. Understand global variables. Draw a simple stack trace.	Aliasing & Scope
3.05: Oregon Trail	Use project planning skills to complete a longer-term project. Create functions to organize a project. Apply skills learned in units 1-3 to create a functioning program.	Oregon Trail

Unit 4 - Nested Loops and Lists

Lesson	Objectives	Lab
4.01: Looping Basics	Define and identify: for loop, item, iteration, scope. Recall looping in Snap! and reapply the concept in Python. Loop through (traverse) the items in a list. Be aware of the scope of variables during iteration.	de_vowel
4.02: For Loops	Define and identify: range. Use the range and `len()` function to update lists via for loops.	Getting Loopy

Lesson	Objectives	Lab
4.03: Nested For Loops	Define and identify: nested for loops, stack trace. Use nested for loops via a function and a for loop. Use nested for loops via two loops nested. Use a stack trace to understand and demonstrate the flow of nested for loops.	Nested For Loops
4.04: Nested Lists & Looping	Define and identify: nested list. Use nested for loops to traverse through nested lists.	Shopping List
4.05: Debugging and Quiz	Read and understand longer programs involving loops. Demonstrate knowledge of looping, lists, and nested loops/lists. Debug programs involving for loops and lists.	Debugging Practice
4.06: Tic-Tac-Toe	Use project planning skills to complete a larger project. Utilize loops, lists, and nested loops/lists to create a Tic-Tac-Toe game.	Tic-Tac-Toe

Unit 5 (Optional) - Music Programming

Lesson	Objectives	Lab
5.01: Earsketch Intro	Define and identify: Digital Audio Workstation (DAW), sound tab, fitMedia(), setTempo(). Play beats using the above functions. Loop through items in a list. Be aware of the scope of variables during iteration.	Intro to EarSketch
5.02: EarSketch Music	Define and identify: rhythm, beat, tempo, measures, setEffect(), makeBeat(). Play beats using the functions. Loop through items in a list. Be aware of the scope of variables during iteration.	EarSketch Music
5.03: Earsketch Control Flow	Define and identify: modulo. Review looping and control structures. Use looping concepts in music making via EarSketch. Use control structures to create music.	Earsketch Control Flow
5.04: EarSketch User-Defined Functions	Define and identify: abstraction, section, A-B-A form. Create and apply user-defined functions to create songs with complicated form.	User-Defined Functions
5.05: EarSketch Project	Create a complete song in EarSketch with multiple parts. Utilize EarSketch's features and functions.	EarSketch Song

Unit 6 - Dictionaries

Lesson	Objectives	Lab
6.01: Introduction to Dictionaries	Define and identify: dictionary, key, value. Create dictionaries of key-value pairs. Access and update items from dictionaries.	Dictionaries & Memes
6.02: Dictionaries Methods	Define and identify: pop, default value. Update values in a dictionary. Add values to a dictionary. Remove values from a dictionary.	Word Counter
6.03: Dictionaries of Lists	Create dictionaries with keys and values of different types. Update, append, or remove list values in a dictionary.	Dictionaries Storing Lists
6.04: Dictionaries Looping	Use loops to traverse through key/value pairs in a dictionary	Dictionaries Looping

Lesson	Objectives	Lab
6.05: Guess Who	Use dictionaries to create the game Guess Who	Buying an Umbrella

Unit 7 - Introduction to Object Oriented Programming

Lesson	Objectives	Lab
7.01: User-Defined Types	Define and identify: class, instance, object, attributes. Create a class and instantiate. attributes to an instance. Manipulate instances and attributes through a function.	Create a Color Class
7.02: User-Defined Types, Part 2	Define and identify: self, `__init__`. Create a class with an `__init__` method. Understand and use the self argument. Instantiate a class with arguments.	Pet Class
7.03: Methods	Define and identify: method, `__str__`, `__add__`, operator overloading. Create a class with an `__init__` method. Understand and use the self argument. Instantiate a class with an argument.	Kangaroo Class
7.04: Inheritance	Define and identify: inheritance, parent class, child class. Create a class that inherits from anther class. Overwrite methods of parent class in a child class.	Pokemon Child Classes
7.05: Pokemon	Engage in class design before beginning coding. Apply what was learned with respect to classes, methods, and inheritance to create an implementation of Pokemon.	Pokemon

Unit 8: Final Project

Lesson	Objectives	Lab
8.01: Final Project Brainstorming and Evaluating	Recall project planning basics from last semester. Identify factors to use when choosing between project ideas. Rank a group of proposed project ideas using the identified factors.	N/A
8.02: Defining Requirements	Define key scenarios for a project and the features required to implement each scenario. Explain the importance of wireframing when designing an application.	N/A
8.03: Building a Plan	Identify the main components of a functional project specification and explain the purpose of each section. Develop a project idea into a full, detailed specification.	N/A
8.04: Project Implementation	Use the skills developed throughout the course to implement a medium- to large-scale software project. Realistically evaluate progress during software development and identify when cuts are necessary. Prioritize features and scenarios and choose which should be eliminated or modified if/when resources and/or time become limited.	N/A

Supplemental Culture Day Lessons

Lesson	Objectives	Lab
01: Binary Day	Define and identify: binary. Describe different representations of data. Represent decimal numbers in binary.	N/A