Introduction to Digital Science

Second Semester

Python Fundamentals

Standards Mastery Framework/Flex Timing Prototype Semester

12 Weeks

Executive Summary

With the exception of the final Master Project, any signoff can be completed *at any time*. It is, however, strongly recommended that they be accomplished in order.

Students will create a ½" White binder with cover sleeve which will contain all of their papers and code printouts, organized in order. This will be submitted for an EXAM grade. *Details provided in class.*

All **S** (small) assignments are considered **normal** coursework
All **M** (medium) assignments are **Quiz** grades*
All **L** (large) assignments are **EXAM** grades*
Grades will be assigned for Quizzes and Exams Individually

All assignments will be given a rating of 1-4 for mastery as follows:

1: Student only understands most basic definitions/concepts. Cannot demonstrate mastery

2: Student can show limited understanding of the definitions/concepts with assistance

3: Student has mastered the definitions/concepts and can perform assigned tasks to standard(s)

4: Student has mastered and exceeded the standard by performing and demonstrating skills beyond what has been explicitly taught and/or required

Stars will be assigned on the status/signoff board as follows:

Blank: No signoff/Not Attempted

Red: 1

Blue: 2

Silver: 3

Gold: 4

For purposes of this initial program, half points will not be awarded

**Any irregularities on assessments are considered honor violations*

Running list of assignments

Size	#	Task
S	1	Paper: 1-page overview of Python Language
S	2	Code: Print Statements
S	3	Code: Comments – Single Line & Multiple Line
M	4	Code: Program that performs multiple forms of print statements, including variable use
S	5	Paper: 1-page overview of how to run Python on a Pi
S	6	Code: Write an algorithm and display the results of a calculation
S	7	Code: Accept input from a user, and then use that information in a print display
M	8	Code: Algorithm that uses multiple user inputs to provide a meaningful result
S	9	Code: Show knowledge of casting from various data types to others
S	10	Code: Show knowledge of using Tuples and Lists
S	11	Code: Show use of Sets and Dictionaries
M	12	Code: Write a program that uses user input to fill a List and Display the list
S	13	Code: Show use of both types of Iteration
S	14	Code: Show use of conditionals
S	15	Code: Write a function that returns a value and use it
L	16	Code: Program that contains two functions: One that calls the other for a value it needs
S	17	Paper: 1-page paper that explains Python modules
S	18	Code: Write a module in which you access a single named method in a separate program
S	19	Code: Add a second and third method to your module
M	20	Code: Write a module of 3 methods that are useful for a specific purpose & use them
S	21	Code: Demonstrate use of date-time object
S	22	Code: Demonstrate use of Try-Except statements in your code
S	23	Code: Create a text file, and read it into a program
M	24	Code: Create a program that reads a text file, changes the text, then writes the new text back to the text file
S	25	Code: Create a basic Lambda function
S	26	Code: Demonstrate knowledge of 4 new built-in Python functions
S	27	Code: Demonstrate knowledge of 4 additional built-in Python functions
M	28	Code: Write a meaningful program that uses at least 4 built in functions, user input, and 1 lambda function
S	29	Write a program that performs a simple function on a Raspberry Pi
S	30	Turn on an LED on a Raspberry Pi
S	31	Demonstrate use of an RGB LED on a Raspberry Pi
L	32	Write a Python program on a raspberry Pi that activates LED(s) based on a novel schema
MASTER PROJECT		
Phase A		Submit detailed project proposal
Phase B		Checkpoint 1
Phase C		Checkpoint 2
Completion		Submit final project
Presentation & Defense		Project Presentation, Demonstration, and Defense

Specifications Details for Signoffs and Checkpoints

1. One-page paper on the history, purpose, use case examples, and operations of programming in Python. Paper should include at a minimum:
    a. History of Python
    b. Most common uses
    c. Explanation of:
        i. Interactive mode
        ii. How files can be used to make full programs including naming of files
    d. Structure and syntax general discussion, particularly around structure of source code in a file
2. Program should demonstrate significant understanding of the various ways to print to the output screen. This includes printing variable values, literals, strings, tuples, lists, and other data structures common to Python.
3. Program should perform some function that is properly documented using:
    a. Header
    b. Single Line Comments
    c. Multiple-Line Comments
4. Program should provide significant output of some meaningful form, using at least three different types of print statements, with commenting that clearly and neatly describes those print statements. This will be the standard for all future code documentation. ***Assessment (Quiz)***
5. One-page paper on what a programmer needs to understand in order to run Python on a Raspberry Pi, to include storing files, finding them, interactive mode, and overall Pi usage of the language on the Raspbian Operating System.
6. Program should demonstrate a complex calculation, defined as some calculation beyond simple arithmetic. Examples: Convert a temperature or volume scale.
7. Program should accept user input and then display what has been entered as verification/feedback.
8. Program should accept multiple user inputs, then combine or reformat the user input and provide a meaningful output. ***Assessment (Quiz)***
9. Program should cast at least 3 different data types to another data type, then print the before and after values for each. If a numerical result, perhaps include it in a calculation to verify conversion.
10. Program should create at least one tuple and one list, modify values of each, and display each state and any value results of their usages.
11. Program should create at least one set and one dictionary, modify values of each, and display each state and any value results of their usages.

12. Program accepts user input into a list, then displays the list. User input should be validated to be correct for a specific set of input before it is entered into the list. ***Assessment (Quiz)***
13. Program properly uses both types of Python iteration.
14. Program properly uses multiple conditional statements to include both **if** conditions and **else** conditions.
15. Write a named function that returns a value and use it in your program.
16. Program uses a new named function, and that function calls a second named function as a helper function. ***Assessment (Exam)***
17. One-page paper explaining the purpose, layout, and procedures for Python modules. Explains why we have them, and where they are best used in professional code.
18. Program accesses a named function within a custom module and uses it.
19. Program demonstrates accessing a second and third named method in the module created in 18.
20. Program is a completely different one from 18-19, accessing and using a *new* module's three *new* methods. ***Assessment (Quiz)***
21. Program demonstrates the use, modification, and output capabilities of the built-in date-time object in Python
22. Program demonstrates use of Python Try-Except statement, with proper syntax and both working and error-throwing code to demonstrate correct exception handling
23. Program accesses a text file and internalizes its content, then prints it in a formatted manner
24. Program reads input from a text file, alters or updates the information, then returns it to the text file. Programmer can determine if it appends or replaces the data. Text file before and after is part of program signoff as well as source code. ***Assessment (Quiz)***
25. Program creates and *documents within the code*, a single non-trivial lambda function. Your program must access the function at least three times with different values as input to the function.
26. Program uses and documents the use and understanding of four (4) built-in native Python functions.
27. Program uses and documents the use and understanding of four (4) *other* (additional) built-in native Python functions.
28. Program is a meaningful program accepting user input, using a custom lambda function, and uses four (4) built in Python functions. ***Assessment (Quiz)***
29. Program performs a single function on a Raspberry Pi (No Electronics required).
30. Program turns on and off, a single LED on a Raspberry Pi.
31. Program activates all three colors on an RGB LED at programmer's discretion.
32. Program activates multiple LED lights on a Raspberry Pi in some manner determined by the programmer. ***Assessment (Exam)***

Spring Master Project Submission Checklist

Python Program

Must-includes (Graded Directly):

- □ Module (1) (In addition to main program)
- □ Methods (3) (May be functions and/or procedures)
- □ Lambda Function (1)
- □ User Input (multiple)
- □ Output (Results, prompts, etc.)
- □ Full documentation including header
- □ 3 Python built-in functions
- □ Must be meaningful code, that is on a subject unique to your personal interests. This is mandatory. Do not waste time writing code in which you are not interested.
- □ Must at a minimum, store information in an external file.

Project Enhancements:

- □ Program uses electronics/Raspberry Pi
- □ Program reads/writes from a text file
- □ Program reads/writes tuples, lists, or other ADS (abstract data structure) from a file
- □ Program uses a database on localhost
- □ Program uses TKinter()/GUI
- □ Other (Preapproved)____
- □ Other (Surprise)____

Submission Notes to teacher:

By signing below, I certify that the code I have written is my own. No one else has written this code but me, and any help I have received has been for clarification/understanding alone. No one has formed any algorithms in this work product but me.

___________________   _________________________   _______________

Name (Printed)   Signature   Date

Spring Master Project Proposal

Class: Introduction to Digital Technology

Period (Circle): 1-Schenk 1-Cowart 3-Schenk

Name (**FULL**):___________________________________

Executive Summary

Specifics to Assist in Project Approval

I understand that all code submitted under my Spring project must be of my own authoring. No one else may generate or otherwise write any algorithms for me. Any code submitted that is 3rd party will be presented to the teacher for prior approval if specific algorithms or libraries are required outside the libraries available in class as part of our normal lab work or language libraries provided.

____________________________________ ___________________

Signature Date

Project is: □ Approved □ Unapproved (See teacher) or
□ Approved as modified below

Spring Master Project Grading Checklist

Class: Introduction to Digital Technology

Period (Circle): 1-Schenk 1-Cowart 3-Schenk

Name (**FULL**):___________________________________

Python Program

Must-includes (Graded Directly):

- □ Module (1) (In addition to main program)
- □ Methods (3) (May be functions and/or procedures)
- □ Lambda Function (1)
- □ User Input (multiple)
- □ Output (Results, prompts, etc.)
- □ Full documentation including header
- □ 3 Python built-in functions
- □ Must be meaningful code, that is on a subject unique to your personal interests. This is mandatory. Do not waste time writing code in which you are not interested.
- □ Must at a minimum, store information in an external file.

Project Enhancements:

- □ Program uses electronics/Raspberry Pi
- □ Program reads/writes from a text file
- □ Program reads/writes tuples, lists, or other ADS (abstract data structure) from a file
- □ Program uses a database on localhost
- □ Program uses TKinter()/GUI
- □ Other (Preapproved)____
- □ Other (Surprise)____

Assigned Grade: ______________/______________ = _____________%

Teacher Notes: