ENGR 14100
Python 1 PA

Individual Assignment: See the course syllabus for a definition of what this constitutes.

Task 1 (of 2)

Learning Objectives: Perform arithmetic operations in Python (i.e., addition, subtraction, multiplication, division, and exponentiation), while keeping in mind order of operations; Create valid identifiers, accounting for relevant Python rules (i.e. keywords) and code standard; Implement the use of expressions in Python to assign values to variables; Recognize and explain the differences between various data types in Python; Convert between data types (e.g. strings to integers, or floats to integers) in Python based on program needs; Identify which resources are available to you to aid in learning Python; Import modules in Python (i.e. math module); Employ the Notepad++ text editor to write, edit, and save Python code; Output data from a script to the screen in Python.

Task:

As you further explore programming, you will find that there are a number of concepts and features hidden within each language that lend greatly to the usability of the language in a niche field or when looking to solve a specific problem in a specific way.

Python contains a number of hidden elements within its built-in modules that apply to unique situations or tasks. In this activity, you will be asked to delve deeper into the world of Python modules. In order to learn about and apply these modules, you will need to do some research on your own. Being able to use your resources to help you find answers to difficult problems will be an important skill for you to master, not only for this class, but also going forward as an engineer.

Suggested modules to research: random; fractions

Suggested built-in functions to research: round()

Investigate the difference between the following and how it affects the calling of functions:

from (Module) import (function)
import module

You are spending your summer consulting at a start-up company that creates on-line educational tools to assist elementary school students in practicing math. You are working on a tool that demonstrates to students the equivalence between doing calculations with decimal numbers and doing the same calculations with fractions. The tool should randomly generate two numbers between zero and ten, round the numbers to the nearest tenth, add the rounded numbers together, and then convert these rounded numbers to fractions and repeat the addition using the fractions.

Write a Python program that will follow the procedure outlined above. Your program output should be in the form shown below with each fraction in its simplest form (i.e. lowest common denominator.) Utilizing random.seed() is required in your program for it to be graded properly.

First Random Number: 0.59875
Second Random Number: 1.5238

$$.6 + 1.5 = 2.1$$

$$\frac{3}{5} + \frac{3}{2} = \frac{21}{10}$$

Before you start programming in Python, create a flow chart representing the algorithm and save it in a pdf file.

Note that your program will be graded based upon the output and the process that you used to reach this output. Your program should not produce any errors when running or points will be deducted.

Task 1 Files:

1. `Py1_PA_Task1_`*`login`*`.py`
2. `Py1_PA_`*`login`*`.pdf`

Task 2 (of 2)

Learning Objectives: Create and execute simple scripts comprised of basic Python concepts; Output data from a script to the screen in Python; Apply course code standard in development of Python scripts; Modularize and comment code in Python for readability and reusability.

Task:

You have just started your first co-op assignment at Mho-ROM Electronic Components, Inc., a large manufacturer of semiconductors and integrated circuits. Over the years, Mho-ROM engineers have written a number of computer programs in languages that are no longer officially supported by the company. With many of those engineers retired or soon to retire, the company is concerned that it will no longer have the in-house expertise it needs to maintain these programs, leaving the company vulnerable should problems arise. Your supervisor has given you a first assignment to update several of these programs to use a modern programming language such as Python.

To get started, you decide to tackle a widely used algorithm which helps to model resistance in a circuit. Since you have already completed ECE 201, you know that resistance can come from anywhere in a circuit: wires, board traces, components within the circuit, or resistors themselves. You are aware that you will need to accommodate two types of resistance networks: series and parallel. Additionally, you know that they are computed differently. Looking back at your ECE 201 notes, you find the following equations for parallel and series resistance:

$$\frac{1}{R_{equiv}} = \frac{1}{R_1} + \frac{1}{R_2} + \ldots + \frac{1}{R_n} \qquad \text{(Parallel Resistance)}$$

$$R_{equiv} = R_1 + R_2 + \ldots + R_n \qquad \text{(Series Resistance)}$$

Additionally, you know that any network of resistances can be calculated by inputting two resistances at a time (the superposition principle) until the entire network has been accounted for. Thus, one only has to deal with two resistances at any time.

For this assignment, you will need to write a Python program to calculate the equivalent resistance of both the series and parallel configurations of two resistors. The program should output the results for both series AND parallel calculations. The output should be formatted in a table modeled similar to the one seen below, with all values rounded to the nearest tenth. You have been given a test case by your supervisor to illustrate the computation which you can use to test your program outputs correct values.

Type of Resistance	First Resistance	Second Resistance	Equivalent Resistance
Parallel	10.0 ohms	50.0 ohms	8.3 ohms
Series	10.0 ohms	50.0 ohms	60.0 ohms

Next, you must demonstrate the correctness of your program by applying it to the test case where the first resistance is an input from the user and the second resistance is $\sqrt{5.6\pi}$ ohms. (Hint: When finding the value of $\sqrt{5.6\pi}$, be as exact as possible.) To better display your logic, create a flowchart and save it in a pdf file.

Task 2 Files:

1) `Py1_PA_Task2_`*`login`*`.py`
2) `Py1_PA_`*`login`*`.pdf`

Make sure you include your INDIVIDUAL HEADER in your file(s) and submit those files to the appropriate turn-in box on Blackboard.