ENGR 14100
Python 1 ACT: Introduction to Python

Recall the guidelines for activities:

1. You should work as a team; **all** team members will be held responsible for all material.
2. You should work on this Task using one computer for the entire team unless otherwise directed.
3. The time estimate given is approximately three times the amount required for an experienced user to complete the task. If you are not making progress, take action to get unstuck.
4. Do not write on the activity sheets and be sure to return them at the end of class.

Task 1 (of 5)

Learning Objectives: Create and execute simple scripts comprised of basic Python concepts; Describe the characteristic and purpose of modules in 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; Apply course code standard in development of Python scripts.
Initial Computer Operator: Person with smallest hands

Background: Modules are a simple way to structure a program. Generally speaking, there are standard modules in the standard library and there can be other Python files, or directories containing Python files, in the current directory (each of which constitutes a module). Modules in Python must be imported before use. For example,

```
import math
```

imports the math standard module. After importing this module, you can use the functions inside math module with the following syntax:

```
math.sqrt(10)
```

Write a Python program that computes the area of a triangle, which has two sides of 12 cm and 4 cm and an angle of 30 degree in between. You can use the following formula to calculate the area of a triangle:

$$\square\square\square\square = 1/2 * \square * \square * \square\square\square(\square)$$

The program should then output the result to the screen in the following format (2 lines):

```
The area of a triangle is __[units] for a given length of ___[units]
and ___[units] with the angle of ____ [units] in between.
```

Note: The sine function in Python returns the sine of x assuming x is in radians.

Example:
Please note that the actual computation results below may not be correct. This example is intended to elucidate matters of format only.

```
The area of a triangle is 12[cm^2] for a given length of 12[cm] and
4[cm] with the angle of 30 [degrees] in between.
```

Task 1 Files:

1) Py1_ACT_Task1_*login*.py

Task 2 (of 5)

Learning Objectives: 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.
Initial Computer Operator: Person with most letters in their first name

Background: In engineering context, you need to deal with various data types to solve your problem. Some of the important categories of data types in Python include Integer types, Floating Point types, and String types.

The following is a section of code that was entered into Python to calculate the volume of a sphere:

```
>>>import math
>>>radius = 6
>>>volume = 4 / 3 * math.pi * radius ** 3
>>>print(‘The volume of a sphere is ’, volume, ‘[cm^3] for a given’)
>>>print(‘diameter of ’, 2*radius, ‘[cm].’)
```

Include the answers to the following questions in a PowerPoint slide.

What changes do you need to make for the code to get the following output?

```
The volume of a sphere is 904 [cm^3] for a given
diameter of  12 [cm].
```

And what changes do you need to make for the code to get the following output?

```
The volume of a sphere is 905 [cm^3] for a given
diameter of  12 [cm].
```

Task 2 Files:

1) Py1_ACT_*login*.pdf

Task 3 (of 5)

Learning Objectives: Perform arithmetic operations in Python (i.e. addition, subtraction, multiplication, division, and exponentiation), while keeping in mind order of operations; Employ the Notepad++ text editor to write, edit, and save Python code; Output Python data from script to screen.
Initial Computer Operator: Person with largest shoe size (remember to rotate)

Background: Python 3 is a powerful programming language for performing computations and database management. As an engineering tool it offers standard mathematical operations as part of its basic environment. Therefore, you can design standard computational models, process data, and generate reports. The language must work within the limits of the machine it is running on. As a result, there may be differences between what you compute by hand and what Python computes.

Part A

1) In PowerPoint create a table with three columns: problem number, hand calculation, and Python calculation.
2) Assuming that equations 1-7 shown below are computed sequentially in the order shown, calculate by hand the result of each expression.
3) Perform these calculations in the Python terminal.
4) Fill in the table on the PowerPoint slide.

Hand Calculation represents the results of the arithmetic as would typically be displayed on a calculator.

Python Calculation represents the results computed and displayed from Python3

Hint: Remember the order of operations. You may need to import the math module to do some of the calculations in Python. Also, for hand calculations, interpret double division symbol as a regular division, but type exactly as shown in Python.

1. A = 10
2. B = A^2
3. C = B – A * 2 / 5
4. D = 13 // 3 + 4 /3
5. E = 4 ^(3 *2)
6. F = 4 ^ (1 / 2)
7. G = 133 % 20

Part B

Answer the following questions:

1. What differences did you notice between the hand calculations and the Python calculations?
2. What Python function can you use to output these variables to the screen?
3. What syntax differences exist between Python and your calculator? Specifically, what are the differences when doing division, modulus operations (remainder), and exponentiation?

Include the answers from **Part A** and **Part B** in a PDF.

Task 3 Files:

1) `Py1_ACT_`*`login`*`.pdf`

Task 4 (of 5)

Learning Objectives: 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.
Initial Computer Operator: Person with fewest letters in their last name

Background: Variables (or indicators) are unique labels for values in memory. These variable names should be descriptive so it is easier to understand how they relate to your algorithm. The form of these must be consistent. Also, Python needs to define its own set of unique indicators that are called keywords. This activity has you debug a series of statements and correct them so that the program will run.

The following is a section of code that was entered into Python and contains numerous errors involving variables.

1) For each line of code that needs correction you should record all of the corrections necessary to make the code valid.
2) Explain why the changes need to be made.

NOTE: Python has reserved variable words as well as restrictions on what characters may be used in a variable name.

Save your answers in:

`Py1_ACT_`*`login`*`.pdf`

```
1.  >>>a=6
2.  >>>B-2=4.99999
3.  >>>CounT= 12.5
4.  >>>
5.  >>>class = a + B_2 – CounT
6.  >>>
7.  >>>name@name = Count * a
8.  >>>
9.  >>>ln = a^2
10. >>>
11. >>>Sum = Name – Ln-a*2
12. >>>
13. >>>def = a^CounT
14. >>>
15. >>>3vari = B_2 – 7
```

Task 5 (of 5)

Learning Objectives: Employ the Notepad++ text editor to write, edit, and save Python code; 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.
Initial Computer Operator: Person with the most books with them today

Background: Modules are handy when they can be used by others. With proper information in the header on how to use them and appropriate comments and description text throughout the algorithm, other programmers will know how to use the new function appropriately. Therefore, comments are needed to quickly and easily support future users of a module. This also applies to essentially all other programming languages. Commenting can also make it easier to debug your own modules by reminding you why you did certain things in your code.

Complete the following activity:

Create a Python program that calculates and displays the area of a circle. Include comments explaining:

i. Acceptable inputs.
ii. The steps in your script.

Task 4 Files:

1) `Py1_ACT_Task5_`*`login`*`.py`

Bonus Activity Submissions

Instructions: Complete and submit **ALL** Task materials associated with this Activity (see 'Submit Files' below). You are allowed to combine the work you and your team completed during the Activity with materials you individually (or as a team) complete outside of class. The Bonus Activity Submission will not be graded and returned to you like a typical assignment. Instead, it will be reviewed, and the bonus point awarded, for its completeness, i.e., for completing ALL the Tasks associated with the Activity. Submitting an incomplete Bonus Activity (something less than all of the Tasks) will be considered an act of **Academic Dishonesty** for which the penalty will be forfeiture of the opportunity to earn future Bonus Activity Submission points.

There are two options for completing the materials for the Bonus Activity Submission:

As an Individual: Combine the work you and your team completed in class with materials you have individually completed outside of class. When submitting an individual Bonus Activity Submission you will append your electric signature (i.e., your typed name) at the top of the file that represents your individual work. Your electronic signature indicates that this is your individual work and you have not collaborated with other individuals (other than the teaching team) to obtain the final materials being submitted – working with other individuals/groups (e.g., discussing ideas and concepts, helping find errors, talking about potential solutions to errors) is permissible up to the point where the work represents a coloration (i.e., working with another person or group to achieve an answer). Any work previously completed by your team should include each team member's electronic signature. The significance of an electronic signature by an individual for team work is stated below.

As a Team or Ad Hoc Group: Combine the work you and your team completed in class with materials your team (or ad hoc group) completed outside of class **(For the Bonus Activity Submission ONLY**: you are allowed to work with any other members of the class to complete the assignment). However, you should exercise care when appending your electronic signature to ensure you are in full compliance). When submitting a Bonus Activity that has been worked on as a team (or ad hoc group) each person will append his/her electronic signature (i.e., his/her typed name) at the top of the file that represents the collaborative work. The electronic signature of each individual implies he/she: was an active participant in the preparation of the materials; and has a general understanding of **ALL** the materials being submitted. Even for work submitted as a team, each individual who wishes to receive credit must submit the team's file (with all appropriate signatures) to their own individual assignment drop box.

Submit Files: Submit *all* files electronically via Blackboard to the appropriate box on time.

1) `Py1_ACT_`*`login`*`.pdf`
2) `Py1_ACT_Task1_`*`login`*`.py`
3) `Py1_ACT_Task5_`*`login`*`.py`