

CSE101 – Spring 2020

Programming Assignment #1

Due March 22, 2020 by 11:59pm, KST. The assignment is worth 12 points.

Instructions

For each of the following problems, create an error-free Python program.

- Each program should be submitted in a separate Python file that follows a particular naming convention: Submit the answer for problem 1 as “Assign1Answer1.py” and for problem 2 as “Assign1Answer2.py” and so on. We have provided initial code to get you started on Problems 1, 2, and 3 (linked to in each problem).
- These programs should execute properly in PyCharm using the setup we created in lab.
- At the top of every file add your name and Stony Brook email address in a comment.

Programming conventions to use

Developers use different standards for programming conventions. Whitespace and variable names vary based on the language and their personal taste or organization. Consistency is important. As you program more you'll develop your own conventions, but for this class use the conventions below:

Naming:

- Choose concise but informative variable names that would be clear to someone reading your code. Avoid uncommon abbreviations. For example, `hourly_rate` is better than `hr`. An alternative style is to write `hourlyRate`. You can use whichever option you prefer, just be consistent.
- Variables and functions should always start with a lower-case letter. For example, `hourlyRate` and not `HourlyRate`.

Whitespace:

- Use blank lines (whitespace) in your program consistently to make your program more readable.

The descriptions of conventions will not be repeated in future assignments, but be sure to continue following them for the rest of the course.

Problem 1 (1 point)

Go through the following steps:

1. Download the provided [Assign1Answer1.py](#) file and bring it into PyCharm.
2. Fill in your info at the top of the file.
3. Run it in PyCharm to make sure it works. Try to understand how all the code works.
4. Add a new function below `printSum`. This new function should print the information described below while satisfying the following requirements:
 - a. Print your name with quotation marks around it (e.g. "Alice Park")
 - b. On the next line, indent the line with a tab character and then print your major and what year you started at SUNY Korea.

So when printed it will look like:

```
"Alice Park"  
    Computer Science 2020
```

5. Add code that calls your new function and test to make sure it works without any errors.

Problem 2 (1 point)

Download the provided [Assign1Answer2.py](#) file and bring it into PyCharm.

Write the implementation in the provided `celsiusToFahrenheit` function that will take the `celsius` input parameter and convert the value into Fahrenheit. Use the following formula to make the conversion:

$$F = (C * \frac{9}{5}) + 32$$

Problem 3: (2 points)

Download the provided [Assign1Answer3.py](#) file and bring it into PyCharm.

The provided program has a function named `shapeName` that is supposed to print out the name of a shape that has 3 to 6 sides. However, the provided program does not work because it has 3 different bugs. Find and fix these three bugs.

Additionally, in the code, write a comment (starting with `#`) describing each bug and what you needed to do to fix it.

Problem 4 (2 points)

The pH scale is used in chemistry to determine how acidic or alkaline something is. It typically ranges from 0 – 14 with lower numbers being acidic, higher numbers being alkaline, and 7 being considered "neutral".

Write a function that takes user input and correctly describes a given pH value as either "Alkaline", "Acidic", "Neutral", or "Invalid input" if the pH number is not between 0 and 14 (inclusive).

Note: below is what a sample input line to your program should look like (on lines starting with >>>) and the resulting output. If the problem says to take user input, any user input lines will always start with >>>. This convention will be used for all of the assignments.

```
>>>pH: 3.4
```

```
Acidic
```

```
>>>pH: 7
```

```
Neutral
```

```
>>>pH: 9
```

```
Alkaline
```

```
>>>pH: 50
```

```
Invalid input
```

```
>>>pH: -7
```

```
Invalid input
```

Problem 5 (2 points)

You can calculate the length of sides of a right triangle using the Pythagorean Theorem:

$$a^2 + b^2 = c^2, \text{ where } a, b, \text{ and } c \text{ are the lengths of the right triangle's 3 sides.}$$

Write a function named `isRightTriangle` that takes three integers as parameters for the length of sides, and returns as a string whether or not they form a right triangle.

Provide 3 test cases with this program to test the code. A test case is code that calls the `isRightTriangle` function and provides specific values to test the function and then prints the output. The provided code for problem 1, 2, and 3 provides examples of what test cases look like. Below are examples of two test cases and expected output.

Note: the lines with >>> in this code example are input to your Python program that you would write in the Python file. They are not user input (with the `input` function), as the problem did not say to take user input.

```
>>> print(isRightTriangle(3, 4, 5))
```

```
It forms a right triangle
```

```
>>> print(isRightTriangle(3, 1, 6))
```

```
It does not form a right triangle
```

Problem 6 (2 points)

The distance between Seoul and Los Angeles is 5950 *miles*. Write a Python program that first takes in as user input a flying speed in *kilometers per hour*, then calls a function that does the following:

Given a parameter of a flying speed in *kilometers per hour*, return from the function the number of *seconds* required to fly between Seoul and Los Angeles and then print out this value (display the value rounded to the nearest second with no decimal point).

Note: 1 kilometer = 0.621 miles

```
>>>Enter the flying speed in kilometers per hour: 800
```

```
Time required to travel from Seoul to Los Angeles: 43116 seconds
```

Problem 7: (2 points)

Below are 6 expressions that each have a missing operator for the blank space `__`.

For each expression, find the correct operator (either `*`, `+`, `**` or `%`) using Python code to evaluate all four possible operators. This may be helpful to do in the Python Interactive Console.

Then, in the `Assign1Answer7.py` file, for each of the 6 expressions, write *only* the Python expression that results in a correct answer.

For example:

Given the expression: `19 __ 5 = 4`

The missing operator is `%`

So for that example, you would write this correct Python expression in your file:

```
>>> print(19 % 5)    # Will print 4
```

Add all correct expressions in a single Python program and submit on blackboard.

1. `21 __ 7` = 0
2. `5 * 3 __ 3` = 135
3. `((21 * 7) __ 16) % 31` = 8
4. `((20 __ 80) + 337) % 1000` = 937
5. `((100 % 19) + 20) __ 7` = 4
6. `17 % 2 + 31 __ 2` = 2