ECS 32A: Programming Assignment #1

Instructor: Aaron Kaloti

Summer Session #1 2020

Contents

1 Changelog

You should always refer to the latest version of this document.

- v.1: Initial version.
- v.2: Fixed output of second example of part #1.

2 Due Date

This assignment is due the night of Monday, June 29. Gradescope will say 12:30 AM on Tuesday, June 30, due to the "grace period" (as described in the syllabus). *Do not rely on the grace period for extra time; this is risky.*

Some students tend to email me very close to the deadline. This is also a bad idea. There is no guarantee that I will check my email right before the deadline.

3 General Submission Requirements

Use whichever environment/editor that you prefer, so long as it produces a Python file. Make sure that you submit the Python files with the correct names to Gradescope; the Python file names are given in each part below. These names must match *exactly*; if you name your file for part #1 `sumProd.py` instead of `sum_prod.py`, that won't work. You may submit infinitely many times to Gradescope, before the deadline.

Once the deadline occurs, whatever submission is your **active submission** will be the one that dictates your final score. By default, your active submission will be your latest submission. However, you can change which submission is your active submission, which can be useful in a few rare cases; I'll talk about this either on Wednesday (6/24) or on Thursday (6/25).

As I mentioned during the first lecture, the output of your programs needs to match the expected output ***exactly***.

4 Reading User Input

Programs often respond to some sort of input. At this level, that input will always be provided by the user via their keyboard. This need not always be the case. (Consider a self-driving car that gets input from sensors.) For the two parts of this programming assignment, you will need to get input from the user. We will talk about this extensively during the Wednesday (6/24) lecture, but just in case you want to get an early start or want a more text-based explanation, here you go.

If you want to have the user enter an integer, use the below line of code. The variable `val` can have any name that you wish. Moreover, the message that is within the double quotes can be whatever you want it to be. If you want to allow for the user to enter a `float` (i.e. a number that may have something past the decimal point), then use `float` instead of `int` in the line below. Be careful about copy/pasting the below line; when you copy/paste a line from a PDF, there is a good chance that certain characters, e.g. the quotation mark, will be distorted; it is safer to retype the line by hand.

```
val = int(input("Enter an integer: "))
```

When the above line is reached in your Python program, the program will print `"Enter an integer: "` and stall until the user types an integer and presses the Enter key. After that, the program will resume, and whatever integer (which can be multiple digits, e.g. `53`, `-8792`) that the user entered will be placed into the variable `val`.

If you want to have the user enter a string, use the below line of code. The only significant difference compared to the above line is that the `int(...)` is removed.

```
first_name = input("Enter your first name: ")
```

4.1 Example: Add Two to User Input

Below is a program that prompts the user for an integer and then prints the result of adding two to that integer.

```
val = int(input("Enter a number: "))
new_val = val + 2
print("{} plus 2 is {}.".format(val, new_val))
```

Here are two examples of the above program being executed.

```
Enter a number: 7
7 plus 2 is 9.
```

```
Enter a number: -1
-1 plus 2 is 1.
```

5 Problems

5.1 Part #1

In a file called `sum_prod.py`, write a Python program that prompts the user for (i.e. asks the user to enter) two numbers and prints the sum and product of those two numbers. Here are some examples of how your program should work:

```
Enter first integer: 5
Enter second integer: 2
The sum is 7
The product is 10
```

```
Enter first integer: -2
Enter second integer: 3
The sum is 1
The product is -6
```

```
Enter first integer: 5
Enter second integer: 13
The sum is 18
The product is 65
```

Notice that there is a space between the colon of the "input prompt" and the actual value entered by the user. That space must be generated by your program. Below is a line that you could use to get the first input from the user; notice the space after the colon.

```
a = int(input("Enter first integer: "))
```

You may assume that the user will only enter integers. Do not use `float(...)`; use `int(...)` instead, as shown above.

As stated above, the output of your program must match the expected output ***exactly***. If, for the last example input, your program says `The sum seems to be 18` instead of `The sum is 18`, then the Gradescope autograder will give you a zero. (As stated above, you can submit infinitely many times before the deadline.) As another example, if your program prints `The sum is 18.` (notice the period at the end) instead of `The sum is 18` for the last input, then the autograder will give you a zero. As yet another example, if your program prints five blank lines after printing everything it was supposed to print, then – again – the autograder will give you a zero.

5.2 Part #2

In a file called `animal.py`, write a Python program that prompts the user for their favorite animal and one word that they would use to describe that animal. Your program should then tell the user what they entered in the way demonstrated in the examples below.

In this first example, for the first input, the user enters "dragon", and for the second input, the user enters "flying".

```
What is your favorite animal? dragon
In one word, how would you describe this animal? flying
Your favorite animal is the flying dragon.
```

In this example, for the first input, the user enters "cat", and for the second input, the user enters "meowing".

```
What is your favorite animal? cat
In one word, how would you describe this animal? meowing
Your favorite animal is the meowing cat.
```

In this example, for the first input, the user enters "snake", and for the second input, the user enters "slithering".

```
What is your favorite animal? snake
In one word, how would you describe this animal? slithering
Your favorite animal is the slithering snake.
```

You may assume the user won't enter any erroneous inputs, e.g. more than one word for the first or second input.

UCDAVIS
COMPUTER SCIENCE