CS 101
HOMEWORK 2

For the first exercises, indicate the value returned by the Python interpreter without actually running the code. Write on a sheet of paper the answers and hand it in next week.

For the programming assignments, create a new Python script file that you will name *yourname_homework2.py* Save this file in a folder on your computer where you will keep all your homework assignments. Write some Python code inside this file that I will be able to run on my computer without errors.

When finished, upload your file in Google Classroom using the Homework 2 assignment link. Don't forget to add comments at the beginning of each problem and throughout your code for clarity.

Tasks

1. What is the value displayed in the Python shell? Try to answer without actually running the code. Think about how integers and float types are handled by the Python interpreter. You may run the code to see if you answered correctly.

(a) $5 + 2 - 3$
(b) $3 * 2.0$
(c) $- - 5$
(d) $15/2$
(e) $15.0/2.0$
(f) $(3 * *2 + 1) * *2$

2. What is the value of the last expression? Try to answer without running the code. After you answered, you may check by running the code. Keep in mind that some code might generate an error, so you can answer: error and a short explanation.

(a)

```
>>> x = 4
>>> x + 1.0
```

(b)

```
>>> x = 4
>>> x + 1.0
>>> x
```

(c)

```
>>> y = 2
>>> y = 5.0
>>> y
```

(d)

```
>>> a = 3
>>> a = a + 5.0
>>> a
```

(e)

```
>>> a = 3
>>> a = a + 5.0
>>> b
```

3. Remember that // is integer division and / is floating point division. The exercises also help you understand what the assignment operator does (=) and the equal to (==) operator does (it checks whether the value of two expressions are equal or not and returns a boolean with values of True or False). You may later check your answer by running the code.

(a)

```
>>> 15//2
```

(b)

```
>>> a = 10
>>> b = 3
>>> a = a//b
>>> a
```

(c)

```
>>> x = 20
>>> x == 2.0
>>> x
```

(d)

```
>>> a = 20
>>> a = a == 2.0
>>> a
```

(e)

```
>>> y = 20
>>> y = y == 20
>>> y
```

4. Write a Python program in which given two integer numbers it prints their product, only if the product is equal to or lower than 1000, else it prints their sum. For example given variables num1 and num2 with values of 2 and 5:

```
num1 = 2
num2 = 5
```

your program should display: The product is 10. If variables num1 and num2 have high values like 100 and 800, then their product would be over 1000 and you have to add the two numbers.

```
num1 = 100
num2 = 800
```

your program should display: The sum is 900.
Use the conditional statement and the print statement. For convenience I included the two slides about these statements next.

- The conditional allows a program to test a condition and if the condition is True a set of statements are executed and if the condition is False a different set of statements are executed
- In Python the conditional looks like this:

```
if boolean_expression:
    block_of_code_if_boolean_expression_is_True
else:
    block_of_code_if_boolean_expression_is_False
```

- The block of code executed if the expression if False can be missing entirely:

```
if boolean_expression:
    block_of_code_if_boolean_expression_is_True
```

The print statement tells the Python interpreter to call the function print(‘some string’) and prints out the string inside the parenthesis. See the example:

```
>>>print(‘print a string’)
```

You could use print to display a string followed by the value of a variable like in this example:

```
>>>name = ‘John’
>>>print (‘My name is‘, name)
```

If you run this code, notice how a space character will be inserted between the string ‘My name is’ and name. The variable can also contain a numeric value like in this example:

```
>>>number = 10
>>>print(‘Number :’, number)
```

In fact, you can use print to display several variables and strings at a time. Run the following code:

```
>>>x = 10
>>>y = 20
>>>print(‘x:’, x, ‘y:’, y, ‘x+y=’, x+y)
```

Notice how the arguments of print inside the parenthesis are separated by comma.

I am including some extra programming problems for those of you who enjoy more practice. These problems are not part of the regular homework but we will discuss them in class.

1. Write a program that counts the number of words in a string that is entered by the user at the beginning of the program. You may use **input()** to prompt the user for a string. For example:
   str = input(”Enter a string: ”)
   print(str)
   You might also use **number = str.count(substring)**. For example:
   ’Excellent’.count(’e’) will return 2, the number of times letter ’e’ is found in string ’Excellent’.
2. Write a program to prompt the user to enter a positive number then return the square root of the number.