CS 171 - Lab 2

Professor Mark W. Boady and Professor Adelaida Medlock

Content by Professor Lisa Olivieri of Chestnut Hill College

Detailed instructions to the lab assignment are found in the following pages.

- Complete all the exercises and type your answers in the space provided.
- For questions 12 and 21 of this lab you must submit screen shots of your source code (.py file)

What to submit:

- Lab sheet in PDF format to Gradescope

Submission must be done via Gradescope

Students' Names: Tony Kabilan Okeke

User IDs (abc123): tko35

Possible Points: 88

Your score out of 88:

Lab Grade on 100% scale:

Graded By (TA Signature):

Question 1: 7 points

Execute each of the below Python statements. Write the output for each on the space provided.

(a) (1 point) `print (16 + 3)`

19

(b) (1 point) `print (16 – 3)`

13

(c) (1 point) `print (16 * 3)`

48

(d) (1 point) `print (16 ** 3)`

4069

(e) (1 point) `print (16 / 3)`

5.333333333333333

(f) (1 point) `print (16 // 3)`

5

(g) (1 point) `print (16 % 3)`

1

Question 2: 7 points

State the arithmetic operation each symbol represents.

(a) (1 point) **+ addition**

(b) (1 point) **– subtraction**

(c) (1 point) *** multiplication**

(d) (1 point) **** exponentiation**

(e) (1 point) **/ division**

(f) (1 point) **// Floor division**

(g) (1 point) % **Modulus**

FYI: An **assignment statement** is a line of code that uses a "**=**" sign. The statement stores the result of an operation performed on the right-hand side of the sign into the variable memory location on the left-hand side.

Question 3: 2 points

Enter and execute the following two lines of Python code:

```
age = 15
print ("Your age is", age)
```

(a) (1 point) What does the assignment statement: **`age = 15`** do?
This stores the integer 15 into the variable, `age`

(b) (1 point) What happens if you replace the comma (**,**) in the **`print`** statement with a plus sign (**+**) and execute the code again?

The output would be "Your age is15" since `+` concatenates two strings with no spaces.

Question 4: 2 points

Execute the following code.

```
answer = 6 ** 2 + 3 * 4 // 2
final = answer % 4
```

(a) (1 point) What is the value of **`answer`** after running the code? **42**

(b) (1 point) What is the value of **`final`** after running the code? **2**

Question 5: 2 points

Execute the following code.

```
1  schoolName = "Drexel"
2  typeOfSchool = "University"
3  fullName = schoolName + typeOfSchool
4  print (fullName)
```

(a) (1 point) The third line of code contains an assignment statement. What is stored in the variable **`fullName`** when the line is executed?
The string literal "DrexelUniversity"

(b) (1 point) How can you fix the output so that the words are separated?
Replace the third line with `fullName = schoolName + ‘ ‘ + typeOfSchool`

Alternatively, delete the 3rd line and replace the 4th with:
`print (schoolName, typeOfSchool)`

FYI: The “**+**” concatenates the two strings stored in the variables into one string. “**+**” can only be used when both operators are strings.

Question 6: 2 points

Execute the following code.

```
addressNumber = 9701
streetName = "Germantown Ave"
streetAddress = addressNumber + streetName
print (streetAddress)
```

(a) (1 point) What happens when you execute the code?
A TypeError is produced because integers can’t be added to strings

(b) (1 point) How could you fix the code to work correctly?
Replace the first line with `addressNumber = “9701”`

Question 7: 2 points

Before entering the following code into the Python interpreter, try to figure out what you think the statement should print, then execute it.

```
myWord = "Hello!" * 10
print (myWord)
```

(a) (1 point) What you think it does?
Print a string with “Hello!” repeated 10 times (with no spaces)

(b) (1 point) What does it actually do?
Print a string with “Hello!” repeated 10 times (with no spaces)

Question 8: 6 points

Let’s take a look at a program that prompts the user for two numbers and subtracts them.

```
firstNumber = input ("Enter a number: ")
secondNumber = input (“Enter a number: ")
difference = firstNumber – secondNumber
print ("Difference = ", difference)
```

(a) (2 points) What happens when you execute the code?
A TypeError because subtraction is not valid for strings.

(b) (2 points) Fix the program so it works as intended. What did you do?
Replace the first two lines as follows:

```
firstNumber = int( input( “Enter a number: ” ) )
secondNumber` = int( input( “Enter a number: ” ) )
```

(c) (2 points) What does the `int` command do?
It converts its argument to an integer type

Question 9: 3 points

Write the line of Python code that calculates and prints the answer to the following arithmetic expressions:

(a) (1 point) $\mathbf{8}^{4}$
8 ** 4

(b) (2 points) The sum of 5 and 6 multiplied by the quotient of 34 and 7 using floating point arithmetic.
(5.0 + 6.0) * 34/7

Question 10: 1 point

Write an assignment statement that stores the remainder obtained from dividing **87** and **8** in the variable **leftover**
leftover = 87 % 8

Question 11: 2 points

Start with the following code.

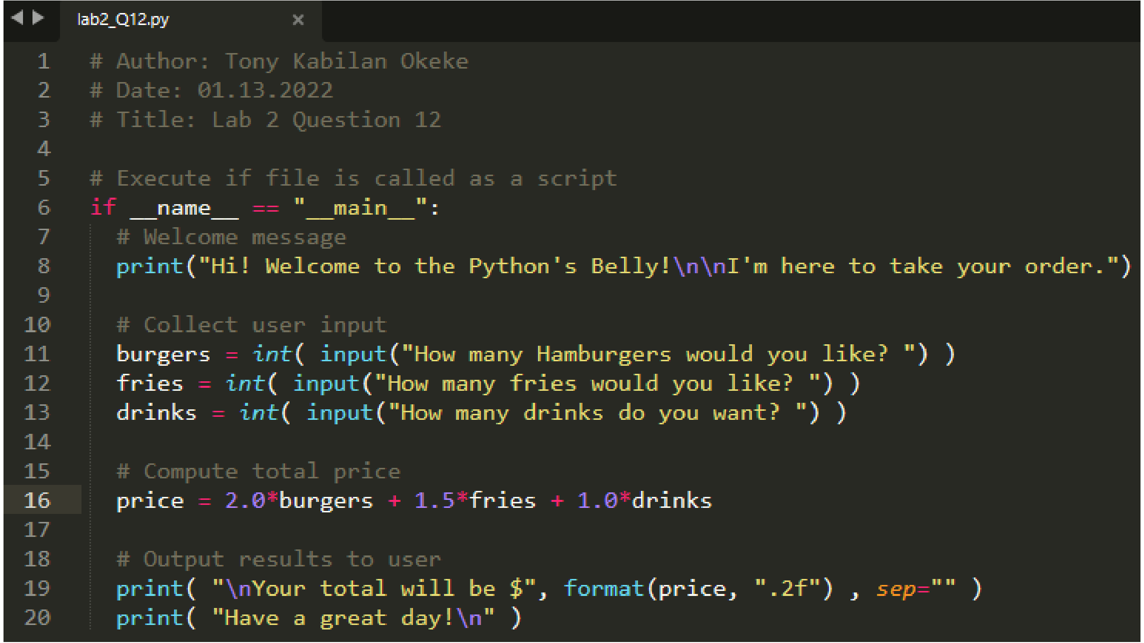
```
courseLabel  = "CMSC"
courseNumber = "190"
```

Write a line of Python code that concatenates the label with the number and stores the result in the variable **courseName**. Be sure that there is a space between the course label and the course number when they are concatenated.
courseName = courseLabel + “ ” + courseNumber

Question 12: 5 points

Create a program the outputs the total cost of a lunch order. Users should be prompted to input the number of hamburgers, fries, and drinks they want, and the program should print the total cost of the order. The hamburgers cost 2.00, fries cost 1.50, and drinks cost 1.00. Be creative and professional in prompting the user for the information and in displaying the output.

Take two screenshots. One of your source code file and one of the output of your code showing that it works. Paste both as answers to this question.



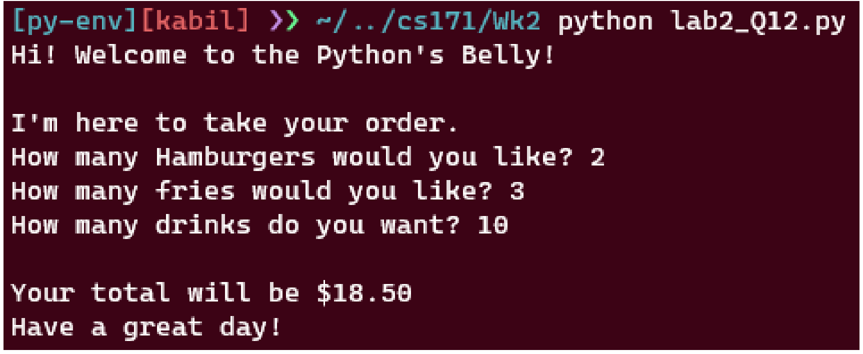
```python
# Author: Tony Kabilan Okeke
# Date: 01.13.2022
# Title: Lab 2 Question 12

# Execute if file is called as a script
if __name__ == "__main__":
    # Welcome message
    print("Hi! Welcome to the Python's Belly!\n\nI'm here to take your order.")

    # Collect user input
    burgers = int( input("How many Hamburgers would you like? ") )
    fries = int( input("How many fries would you like? ") )
    drinks = int( input("How many drinks do you want? ") )

    # Compute total price
    price = 2.0*burgers + 1.5*fries + 1.0*drinks

    # Output results to user
    print( "\nYour total will be $", format(price, ".2f") , sep="" )
    print( "Have a great day!\n" )
```



```
[py-env][kabil] >> ~/../cs171/Wk2 python lab2_Q12.py
Hi! Welcome to the Python's Belly!

I'm here to take your order.
How many Hamburgers would you like? 2
How many fries would you like? 3
How many drinks do you want? 10

Your total will be $18.50
Have a great day!
```

Question 13: 9 points

So far you have not been concerned about formatting output on the screen. Now you will discover how Python allows a programmer to precisely format output. Some of the differences are very subtle, so take notice of the differences in outputs.

Start by entering and executing the following code.

```
numLaptops = 7
laptopCost = 599.50
price = numLaptops * laptopCost
print ("Total cost of laptops:", price)
```

(a) (1 point) What is the problem with the manner in which the output is displayed?
There is no $ in the output. Also, the price only has 1 decimal place, not 2.

(b) (1 point) Replace the last line of code with the following:

```
print ("Total cost of laptops:", format (price, ".2f"))
```

How did the output change?
The price is printed with 2 decimal places

(c) (2 points) Replace the last line of code with the following:

```
print ("Total cost of laptops:  $", format (price, ".2f"))
```

How did the output change?
There is a $ printed in front of the price (with a space in between)

(d) (4 points) Experiment with the number `".2"` in the print statement by substituting the following numbers and state the results. What number does each display?

.4 **4196.5000**

.0 **4196**

.1 **4196.5**

.8 **4196.50000000**

(e) (1 point) Now replace `0.2f` with `12.5f`. What is the output?
"Total cost of laptops: $ 4196.50000" – The space between $ and the price has increased

Question 14: 4 points

In the command `format (variable, "n.nf")`, what does each of the following represent?

(a) (1 point) `variable` – **a variable stored in python**

(b) (1 point) First `n` – **defines the amount of space the formatted string will take up. If it is larger than the length of the string, adds padding to the string**

(c) (1 point) Second `n` – **the number of decimal places**

(d) (1 point) `f` – **indicates that string should be formatted as a float**

Question 15: 2 points

Revise the print statement by changing the “`f`” to “`d`” and `laptopCost = 600`. Execute the statements.

```
print ("Total cost of laptops:", format (price, ’2d’))
print ("Total cost of laptops:", format (price, ’10d’))
```

Explain the output format.
The first line just prints out: “Total cost of laptops: 359700”
The second line prints out: “Total cost of laptops: 359700”. In this case it adds 4 spaces in front of the string.
Also, changing “f” to “d” formats the variable as an integer instead of a float. Consequently, it returns an error unless price is converted to an integer first.

Question 16: 2 points

Explain how the function `format (var, ’10d’)` formats numeric data. `var` represents a whole number.
This function will allocate 10 characters to formatting the string. If the number stored in var is less than 10 digits long, it will be printed with an appropriate number of spaces in front of it so that the total length of the string is 10 characters. If the number is larger than 10 digits long, it will be printed in full with no additional spacing.

FYI: Computers perform four main operations on data:

- Input data into a computer
- Output data to a screen or file
- Process data using arithmetic, logical, searching or sorting operations
- Store data

Question 17: 10 points

Use the following program to answer the questions below. Run the code and experiment with it.

```
1   #Getting Information
2   itemName = input ("Enter the name of the item: ")
3   numItems = int (input ("Enter the number of items: "))
4   itemCost = float (input ("Enter the cost of one item: "))
5
6   #Calculating Price
7
8   totalCost = numItems * itemCost #Procesing, Storage
9
10  #Print results
11  print ("Item name:", itemName)
12  print ("Cost of one item:", itemCost)
13  print ("Number of items purchased:", numItems)
14  print ("Total Cost: $", format (totalCost, "0.2f"), sep = " ")
```

(a) (8 points) In the program listed above, label each line of code in which the following operations are executed: Input, Output, Processing, Storage. You only need to label lines that execute statements.
Input & Storage: Lines 2, 3, and 4
Processing & Storage: Line 8
Output: Lines 11, 12, 13, and 14

(b) (2 points) Examine the last line of code and its corresponding output. Explain what **`sep = " "`** does.
The `sep` argument specifies what character is used to join the arguments of the print function.

Question 18: 4 points

(a) (2 points) Enter and execute the following code: **`print ("Hello" * 10)`**. What does it do?
It prints the string “Hello” 10 times with no spaces

(b) (2 points) Write one line of Python code that will print the word **“Happy!”** one hundred times.
`print( “Happy!” * 100)`

Question 19: 4 points

Start with the following code.

```
itemCost = input ("Enter cost of item: ")
```

(a) (2 points) Add one line of code that calculates the cost of 15 items and stores the result in the variable **`totalCost`**.
`totalCost = 15 * itemCost`

(2 points) Add one line of code that prints the total cost with a label, a dollar sign, and exactly two decimal places. Sample output: **Total Cost: $22.50**
`print( “Total Cost: $”, format(totalCost, ‘.2f’), sep=‘’)`

Question 20: 2 points

Start with the following code.

```
height1 = 67850
height2 = 456
```

Add two new lines of code to make the program give the following output. (The columns are 7 characters wide.)

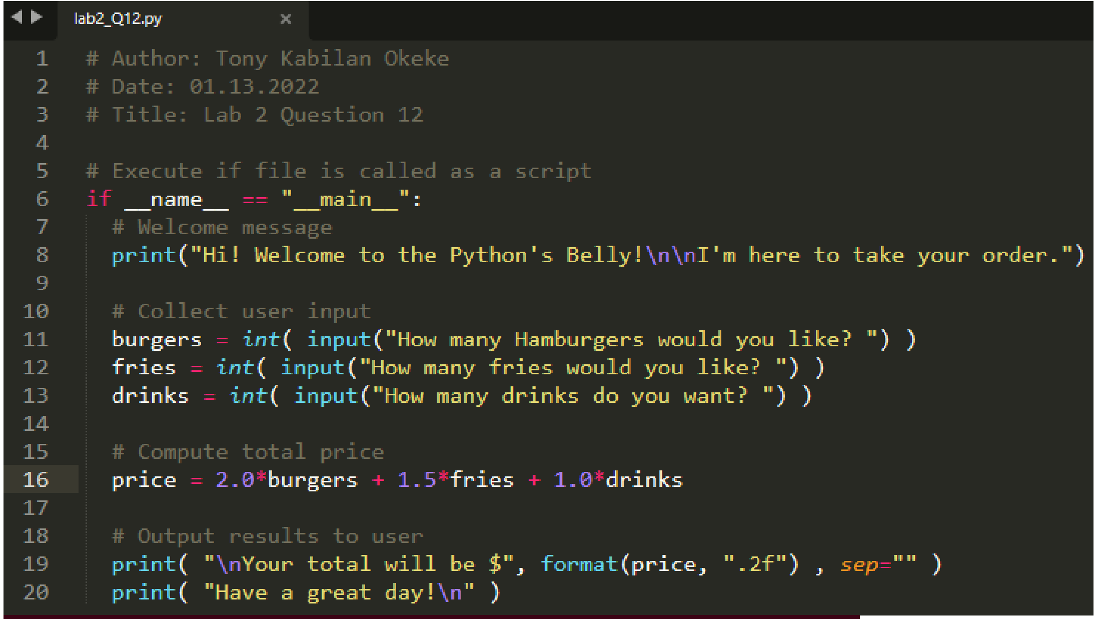
```
Height 1:   67850
Height 2:     456
```

```
print (“Height 1:”, format(height1, “7d”))
print (“Height 2:”, format(height2, “7d”))
```

Question 21: 10 points

Modify your answer to Question 12 (Lunch Order). Use formatting commands to make the output easier to read. Be creative and professional in prompting the user for the information and in displaying the output.

Take two screenshots. One of your source code file and one of the output of your code showing that it works. Paste both as answers to this question.



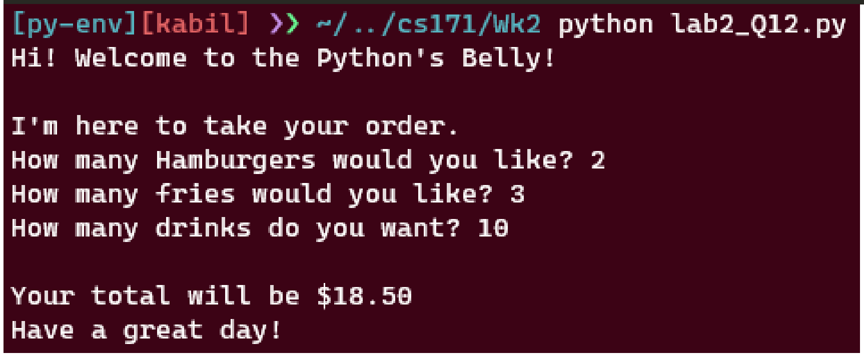
```python
# Author: Tony Kabilan Okeke
# Date: 01.13.2022
# Title: Lab 2 Question 12

# Execute if file is called as a script
if __name__ == "__main__":
  # Welcome message
  print("Hi! Welcome to the Python's Belly!\n\nI'm here to take your order.")

  # Collect user input
  burgers = int( input("How many Hamburgers would you like? ") )
  fries = int( input("How many fries would you like? ") )
  drinks = int( input("How many drinks do you want? ") )

  # Compute total price
  price = 2.0*burgers + 1.5*fries + 1.0*drinks

  # Output results to user
  print( "\nYour total will be $", format(price, ".2f") , sep="" )
  print( "Have a great day!\n" )
```



```
[py-env][kabil] ❯❯ ~/../cs171/Wk2 python lab2_Q12.py
Hi! Welcome to the Python's Belly!

I'm here to take your order.
How many Hamburgers would you like? 2
How many fries would you like? 3
How many drinks do you want? 10

Your total will be $18.50
Have a great day!
```