**CSC 148 Lab 3: Functions and Conditionals**

**Your name:** Jeffrey Williams

**AU ID:** 5030143

**Due: Before class, September 24h, 2021**

**Covered topics: Variables, Assignments, Functions (with return), Conditionals and Simple looping**

For this lab, please first complete sections 1 to 3, filling the blanks on this docx file. Hence, put your solution of the exercises in section 4 in **separate .py files**, and name the files using the following convention **QuestionNumber\_YourName.py**.

Put all the .py files in a folder, zip it, and upload the zipped folder to Canvas.

You will exercise function definition, calling return and conditionals (if, else, elif and nested if statement) in this lab.

For more tutorials on if, elif, else, please refer to your textbook.

<http://www.greenteapress.com/thinkpython/html/thinkpython006.html>

and here:

<http://anh.cs.luc.edu/python/hands-on/3.1/handsonHtml/ifstatements.html>

1. (6pts) Repeated string concatenation. You know that \* is the multiplication operator for integers and floats. It turns out that \* can also be used to “repeat” a string, in a sense that the following

examples will clarify. Enter (or copy-paste) each of the following expressions into the Python console and write down the evaluation results (if any).

|  |  |
| --- | --- |
| **Expression** | **Evaluation** |
| s = ‘abc.’ | None, value stored |
| s | ‘abc’ |
| s\*2 | ‘abcabc’ |
| s\*4 | ‘abcabcabcabc’ |
| n = 3 | None, value stored |
| s\*n | ‘abcabcabc’ |

2. (4pts) Random number generation. The built-in module random contains functions for generating random numbers. One such function is randint(a,b), which, if a and b are integers, randint(a,b) return a random integer *N* such that a <= N <= b

Let’s try it out. First, in Python, make the name of the random module accessible to Python by entering **import random** at the interactive prompt.

Next, circle the function call that is correct given the **import random** command you just entered.

random.randint(1,3) randint(1,3)

Why would the other option give you an error in this case?

“randint” needs to be called from the library it came from, random. Otherwise, it would not know where to append “randint” from.

Run the command above again, you will see that a different integer is generated. How do you generate the same random integer?

To get the same random integer in this case, you would change the maximum value from 3 to 1. There would be no other integer to print in that case.

Write down the code for the following:

Generate a random integer between 1 and 100,

If it is greater than 50, print out “you won!”

Verity your answer with PyCharm.

import random

x = random.randint(1,100)

if x > 50:

print(“You won!”)

else:

print(“You lost…”)

3. Call frames. Write down a call box (how a variable stores its values).

Write the call frames (box notation) of every variable in the following code:

|  |  |
| --- | --- |
|  | **Value of variable** |
| total = 2000 | 2000 |
| r = 0.1 | 0.1 |
| x = total - total \*r | 1000.0 |
| total = x | 1000.0 |
| r = r\*2 | 0.2 |
| x = total-total\*r | 1440.0 |
| print(x) | 1440.0 |
| total | 1000.0 |

4. Defining and calling functions:

1. (3pts) Write a function, **greetings\_multi\_hi(name, value)** that takes two arguments, a string and an integer and prints “Hi, xxx” (xxx is the string of **name**) of the **value** number of times.

For example:

greetings\_multi\_hi(“john”, 2) ->

“Hi, John”

“Hi, John”

Do not forget to call your functions.

1. (3pts) Complete function **rand\_hi(name)**. Let’s now make rand\_hi() in *rand\_greetings.py* do something interesting: generate a random number of “hi”s, using the function randint() we saw earlier.

For example, rand\_hi(“John”) will print out “hi, John” a random number of times. The random integer is generated by the computer.

Then in the script rand\_greetingspy call the function rand\_hi(name) to output the results.

1. (2pts) A local discount store has a policy of putting labels with dates on all of its new merchandise. If an item has not been sold within two weeks, the store discounts the item by 25% for the third week, 50% for the fourth week, and 75% for the fifth week. After that no additional discounts are given. Develop the function **new\_price(initial\_price, num\_week)**, which takes the initial price of an item and the number of weeks since the item was dated and produces the selling price of the item.

*def new\_price(initial\_price, num\_week):*

*# your code goes here*