Lab 1 – PART 1

Topics Covered:

- Introduction to Pycharm
- Printing and simple arithmetic
- Assigning Values to Variables, Arithmetic Expressions, Operator Precedence
- Boolean Expressions
- Strings & Concatenation
- Printing, Formatting Strings, Escape Sequences
- Conversion Types, Casting

<u>Pre-Lab Questions</u>

Answer the following questions before the Lab.

Question 1: What function is used to print/output a message to the screen?

a) output()
b) sysprint()
[✓ c) print()]
d) systemprint()

Question 2: Which of the following will correctly print "Hello World!"?

[✓ a) print("Hello World!")]
b) print(Hello World!)
c) sysprint("Hello World!")
d) sysprint(Hello World!)

Question 3: Take a look at the following line of code. Here we are creating a variable *answer* that will store the value of *4+8.* What is the correct way to print this variable?

```
answer = 4+8
```

a) print("answer")
[✓ b) print(answer)]

Question 4: Take a look at the following line of code. Here we are creating a variable *x* that will store the value of *6.* Which one of the following print statements will add 4 to *x*, and multiply the result by 8?

```
x = 6
```

a) print(x+4*8)
b) print(x*4+8)
[✓ c) print((x+4)*8)]

Question 5: For the following print statement, determine whether the output will be True, False, or an error.

```
print(3>6)
```

a) True
[✓ b) False]
c) Error

Question 6: For the following code segment, determine whether the output will be True, False, or an error.

```
x = 3
print(x=3)
```

a) True
b) False
[✓ c) Error]

Question 7: For the following print statement, determine whether the output will be True, False, or an error.

```
print(7==8)
```

a) True
[✓ b) False]
c) Error

Question 8: For the following code segment, determine whether the output will be True, False, or an error.

```
x = 5
print(x+3*0>2)
```

[✓ a) True]
b) False
c) Error

Question 9: Which of the following lines of code will produce this output?

```
The car went,

"vroom!"
```

a) print("The car went, \n"vroom!")
b) print("The car went, \n\vroom!\")
c) print("The car went, \n\"vroom!\")
[✓ d) print("The car went, \n\"vroom!\"")]

Question 10: Look at the following string variables. We want to format these strings for output. Which of the following correctly formats the string using the **interpolation method** so that the output is *Becky 32*?

```
name = "Becky"
age = 32
```

a) print("%d %s" % (name, age))
[✓ b) print("%s %d" % (name, age))]
c) print("{} {}".format(age,name))
d) print("{} {}".format(name,age))

Question 11: Look at the following string variables. We want to format these strings for output. Which of the following correctly formats the string using the **format function method** so that the output is *Becky 32*?

```
name = Becky
age = 32
```

a) print("%d %s" % (name, age))
b) print("%s %d" % (name, age))
c) print("{} {}".format(age,name))
[✓ d) print("{} {}".format(name,age))]

Question 12: What is the output of the following print statement concatenation?

```
print("6" + "8")
```

[✓ a) 68]
b) 86
c) 14
d) 48

Lab 1 – PART 2

The following exercises require some code to be entered and executed. If you have questions, ask your TA.

1. **Write the following code and see what is printed:**

```
answer1 = 5+3
answer2 = 7*4
print("answer1")
print(answer1)
```

2. **Run the following code to see how formatting works.**

```
x=7
print(x)
y=x+5
# The following prints using the format function
print("x is {} and y is {}".format(x,y))
# The following prints using the interpolation method
print ("x is %d and y is %d" % (x,y))
```

3. **Follow the instructions below to complete and run a program.**

Let's say we had a shop, *MyShoppe*, and wanted to prompt for the shop name. We would use:

```
shopName = input("Please enter the shop name: ")
```

This will store the user-entered shop name into the variable shopName. Let's prompt for the quantity of 2 items in the shop, a ring and glasses.

```
ringQTY = input("Please enter ring QTY: ")
glassesQTY = input("Please enter glasses QTY: ")
```

Run the program as it is; it prompts for user input, but doesn't output anything. Let's add print statements for our current variables:

```
print("Shop name is {}".format(shopName))
print("Ring QTY is {}".format(ringQTY))
print("Glasses QTY is {}".format(glassesQTY))
```

Test the program. We can see that the program works and prints the correct quantities that we entered.

Let’s say we wanted to add up all of our inventory in our shop and print the total. Try:

```
print("Inventory Total: {}".format(ringQTY+glassesQTY))
```

What do you think the inventory total will be if we have 4 rings and 5 glasses? Test it (run the program).

Notice how the output is 45 and not 9. Why is that? Well, when you take a value from input it is a string value by default. So when we ask for the shop name, it is a string value. When we ask for the QTY, it is also a string value by default. What is happening when we add these 2 quantity string values is they are being concatenated. This means we need to convert the user entered values for QTY into integers, and we do this using *casting*.

Modify the prompts to include casting:

```
ringQTY = int(input("Please enter ring QTY: "))
glassesQTY = int(input("Please enter glasses QTY: "))
```

Now run the program -- we get the value 9!