Lab 3 – PART 1

Topics Covered:

- While Loops
- For Loops

Pre-Lab Questions

Question 1: Take a look at the following code segment. What is the output?

```
count = 1
while count <= 10:
    count = count + 1
    print(count)
```

a) This loop will print out 0-10

b) This loop will print out 1-10

c) This loop will print out 2-10

[✓ d) This loop will print out 2-11]

e) This loop will print the number 1 infinitely

Question 2: Take a look at the following code segment. What is the output?

```
count = 1
while count <= 10:
      print(count)
      count = count + 1
```

a) This loop will print out 0-10

[✓ b) This loop will print out 1-10]

c) This loop will print out 2-10

d) This loop will print out 2-11

e) This loop will print the number 1 infinitely

Question 3: Take a look at the following code segment. What is the output?

```
count = 1
while count <= 10:
      print(count)
```

a) This loop will print out 0-10

b) This loop will print out 1-10

c) This loop will print out 2-10

d) This loop will print out 2-11

[✓ e) This loop will print the number 1 infinitely]

Question 4: Take a look at the following code segment. What is the output?

```
for x in range(10):
    print(x)
```

a) This loop will print out 1-10

b) This loop will print out 0-10

c) This loop will print out 1-9

[✓ d) This loop will print out 0-9]

Question 5: Take a look at the following code segment. What is the output?

```
for x in range(1,10,2):
    print(x)
```

a) This loop will print out 1-10

b) This loop will print out 2-9

c) This loop will print out every even number from 0-10

[✓ d) This loop will print out every odd number from 0-10]

e) This loop will print the number 1 infinitely

Question 6: Take a look at the following code segment. What is the output?

```
for x in range(1,10):
    print(x)
```

a) This loop will print out 0-9

b) This loop will print out 2-9

[✓ c) This loop will print out 1-9]

d) This loop will print out every odd number from 0-10

e) This loop will print the number 1 infinitely

Question 7: Take a look at the following code segment. What is the output?

```
for y in range(0,5):
    for x in range(1,11):
        print(x)
```

a) This loop will print out 0-5, 11 times

b) This loop will print out 0-5, 10 times

[✓ c) This loop will print out 1-10, 5 times]

d) This loop will print out 1-10, 6 times

Question 8: Answer the following modulo operations. The modulo operator yields the remainder when the first operand is divided by the second.

5%3 = 2
4%2 = 0
7%15 = 7
29%10 = 9
4%16 = 4
12%6 = 0

Why Modulo? This operator is helpful in loops because sometimes we want to do a specific event every n times. For example, if we wanted to do something every 3rd iteration of the loop we would check if the current count%3 == 0.

Question 9: Take a look at the following code segment. What is the output?

```
for x in range(20):
    if x % 3 == 0:
       print(x)
```

[✓ a) This loop will print out 0, 3, 6, …, 18]
b) This loop will print out 1, 3, 6, …, 18
c) This loop will print out all the odd numbers from 0-19
d) This loop will print out all the odd numbers from 1-19

Question 10: Take a look at the following code segment. What is the output?

```
for x in range(20):
    if x % 2 == 0:
       print(x)
```

a) This loop will print out 0, 2, 4, …, 20
b) This loop will print out 1, 2, 4, …, 20
[✓ c) This loop will print out all the even numbers from 0-19]
d) This loop will print out all the even numbers from 1-19

Question 11: Take a look at the following code segment. What is the output?

```
for x in range(7, -3, -1):
    print(x)
```

[✓ a) This loop will print out 7, 6, 5, …, -2]
b) This loop will print out 6, 5, 4 …, -3
c) This loop will print out -2, -1, 0, …, 7
d) This loop will print out -3, -2, -1, …, 6

Lab 3 – PART 2

1. **Follow the instructions below to construct a complete program.**

   Write a program that reads a set of integer values. First, we ask the user for how many numbers they plan to enter, and then ask the user to enter the values, then print:

   a. The average of the values.
   b. The smallest of the values.
   c. The largest of the values.
   d. The range, that is difference between the smallest and largest values.

   Here is some sample output:

```
How many numbers do you want to use today? 3

Enter the first number: 10

Enter the next number: 2

Enter the next number: 24

The average of the values is 12

The smallest of the values is 2

The largest of the values is 24

The range of the values is 22
```

   Let's think about how we want to do this. If we set the first value to be the largest and smallest, then we can take a look at every new value that the user enters and compare it with the largest and smallest values. If the new value is larger than the largest, we can update the largest variable, and vice versa with the smallest. Since this program wants us to allow the user to enter how many numbers they want to use, we can use a while loop that runs until we reach the number that the user specified. Let's start programming!

First, we should prompt the user to enter the number of values they want to enter:

```
n = int(input("How many numbers do you want to use today?"))
```

Now we want to prompt the user for the first value. Like we discussed, we can set the total, largest, and smallest values to be this first value, and then for any other values the user enters, we can compare them to see if any of the variables should be updated:

```
firstValue = int(input("Enter the first number:"))
```

```
largest = firstValue
smallest = firstValue
total = firstValue
```

Now we want to loop as many times as the user requested. We can do this using a while loop. Let’s initialize a counter for the loop to start at 0. Now we need to think about what the loop condition will be. Let's say the user wants to enter 5 numbers, so n=5. We already prompted for the first value, so we want to ask 4 more times. If we make the loop condition

while counter < n

then the loop will run for n=0,1,2,3,4 because those are all less than 5, which means the loop will run for a total of 5 times. We only need it to run for 4 more times, therefore we need to make sure that we subtract 1 from n, because we already asked for the first value:

```
counter = 0
while counter<(n-1):
```

Currently, all our values are equal to the user entered first value. Let's prompt for the next number to see if we need to update any of the values:

```
current = int(input("Enter the next number:"))
```

We should update the total, because that will always be updated with every new number. We should also update the counter because that will be updated with every iteration of the loop:

```
total=total+current
counter=counter+1
```

Now, we need to compare the new input with the smallest and largest. If the new input is smaller than the smallest, then the smallest variable is updated. If the new input is larger than the largest, then the largest is updated.

```
if current<smallest:    #find minimum
    smallest = current
elif current>largest:  #find maximum
    largest = current
```

Now we are done with the loop, because this will continue to run until all of the values are entered. We can now print the statements with the results. We do this outside of the loop, because we only want to print once after all the values are entered.

```
print("the average of the values is: ", total/n)
print("the smallest of the values is {}".format(smallest))
print("the largest of the values is {}".format(largest))
print("the range of the values is {}".format(largest-smallest))
```

Let’s test the program. What happens if we ask to enter 0 numbers? We are prompted to enter the first number. We need to fix this because we only want the program to run when we have at least one number; put an if statement after prompting for n that will only do the computations if n>0. This will also avoid a division by 0 problem when we are printing the average.

```
n = int(input("How many numbers do you want to use today?"))
if n>0:

  firstValue = int(input("Enter the first number:"))
  largest = firstValue
  smallest = firstValue
  total = firstValue

  counter = 0
  while counter<(n-1):
      current = int(input("Enter the next number:"))
      total=total+current
      counter=counter+1
      if current<smallest:    #find minimum
          smallest = current
      elif current>largest:  #find maximum
          largest = current

  print("the average of the values is: ", total/n)
  print ("the smallest of the values is {}".format(smallest))
  print ("the largest of the values is {}".format(largest))
  print ("the range of the values is {}".format(largest-smallest))
else:
  print("You did not want to use any numbers today.")
```

And we are done!

Design some test cases and test the program.

2. Identify errors and fix the code.

The following code segment is a short program that will notify the user when their bank account total has reached $20. It will start with a balance of $50, and will decrement by a dollar each time, printing the total every time. The variable accountTotal should have value 20 at the end of the program. There are 3 errors in this program.   Find the errors and correct the code.

```
accountTotal = 50
while accountTotal =< 20:
    print(accountTotal)

print('Your account has reached $20.')
```