**Lab #4: Repeating Code**

**Objective:** This lab is intended to give you practice working with for loops

1. What must go in the top line of a for loop?
2. Determine, by hand tracing, what would be printed out by the following lines of code.

for i in range(0,10):

print(i)

1. Determine, by hand tracing, what would be printed out by the following lines of code.

for i in range(0,2):

for j in range(0,5):

print(j)

print(i)

1. Determine, by hand tracing, what would be printed out by the following lines of code.

word = “hello”

for i in range(0,len(word)):

print(word[0:i])

1. Determine, by hand tracing, what would be printed out by the following lines of code.

i = 0

while ( i < 6 ) {

for j in range(0,2):

print(j)

I += 1

print(i)

print(j)

1. Determine, by hand tracing, what would be printed out by the following lines of code.

result = 0

shouldLoop = True

while shouldLoop:

for i in range(0,3):

print(result)

if result < 5:

shouldLoop = False

else:

result += 1

**\*Note:** For each of the following questions, you should actually write the methods described in a Main class in IntelliJ and call them in a Main method in order to test them. Then, copy and paste your methods below each question.

1. Create a method titled ***printTriangle***, that accepts one integer argument that represents an amount of rows. You may assume the amount entered will always be greater than or equal to 3. The method should return nothing but should print out a triangle with the input amount of rows.

| Input of 7 | Input of 6 |
| --- | --- |
| \*  \*\*  \*\*\*  \*\*\*\*  \*\*\*  \*\*  \* | \*  \*\*  \*\*\*  \*\*\*  \*\*  \* |

1. You must write two overloaded methods both titled ***instancesOf*** one that accepts two String arguments one called *target* and the other *key*. Both methods should return an integer that represents the amount of times an instance of the key is found in the target string.

| Target | Key | Result |
| --- | --- | --- |
| hello | l | 2 |
| Over there | e | 3 |
| I like butter better | tt | 2 |
| I love bread! | x | 0 |

1. Create a method titled ***dividesSelf*** that accepts one integer as an argument. The method should return a boolean value. The method should determine if the input integer is divisible by all of the digits that make up that number and return true if it does. If the input integer contains a zero the method should immediately return false.

EX: Input 128

128 / 1 = 128

128 / 2 = 64

128 / 8 = 16

| Input | Output |
| --- | --- |
| 128 | true |
| 120 | false |
| 12 | true |
| 173 | false |