**Directions**: Launch BlueJ and load the project Z:\csa\csa\_exercises. *Create the project if it does not already exist.*

Do not forget to test!!!

***Java Directions****: If you are doing this as a Java exercise, you will need to create a class file for each class and test it in a main method. Sample Java main method follows:*

*Public static void main(String[] args) {*

*}*

**WhileLoops Class**

In this exercise, you will create the WhileLoops class that takes two arrays as input. You will create several methods to work with those arrays. One array represents villains and the other contains their level of evilness:

String[] villainArray = {"Joker", "Doc Oc", "Scorpion", "Illusive Man",

"Darth Maul", "Ganondorf", "Alduin"};

int[] evilRatingArray = {5, 6, 4, 7, 9, 6, 11};

***Note****: Typically these values would be stored in a 2d array since they have a 1 to 1 relationship with each other. In other words element [0] of villainArray contains “Joker”; element [0] of evilRatingArray is 5, which represents how evil the Joker is.*

1. Place the following line of code at the very beginning of the file. We will use it to copy the arrays into our instance variable arrays.
   1. import java.util.Arrays;
2. Create an instance variable for each array (give them appropriate names)
3. The constructor method must receive both arrays as parameters and set their respective instance variables (which are arrays). In order to do this, you must use the copyOf() method of the Arrays class *(which you imported previously)*.

**Note2:** Dynamically retrieve the array lengths as you need them. This will come from the .length variable. DO NOT USE the number 7 inside your methods! Your methods need to work no matter how many elements are in the array.

1. **printArrays()** **method**
   1. Prints out each array one after another using a while loop.
   2. Nothing is returned from this method.
   3. Test your method thoroughly.

***Continued on Next Page***

1. **PrintReverse()** **method**
   1. Print out the reverse of each array one after another using a while loop.
   2. Nothing is returned from this method.
   3. Test your method thoroughly.
2. **calcAverage()** **method**
   1. Use a while loop to calculate the average of the values in the integer array.
   2. Return the average from your method
   3. Print the value returned from the method.
   4. Test this method the completely *(all possible combinations)*
      1. How many tests must be created to adequately test this method?
3. **villainReport()** **method**
   1. Use a while loop to print the related values from both arrays. The output should look like:

Villain Report

Villain: Joker \*\* Evil Rating: 5

Villain: Doc Oc \*\* Evil Rating: 6

…

* 1. Nothing is returned from this method.