**CIS 1115 Assignment 4**

Write a complete Java program, including comments in both the main program and in each method, which will do the following:

The main program will start by calling a method named **introduction()** which will print out a description of what the program will do at the top of the first output page. This first method is called just once.

This method will not be sent any parameters, and it will not return a value. It will print several lines of output explaining what the program does, and it will explain how the program should end (see step 4).

1. The main program will ask the user to type in four integer values, positive, negative, or zero (see step 4). The main program will print the four numbers right after they are read in.

2. The main program will call a method named **findsum(),** sending it the four integer values. The method will determine the sum of the larger 3 of the four values and subtract the smallest value, sending the answer back to the main program.

The main program will print the value returned. For example, if you send 5 9 6 4 to the method, the method will return 16, (5+9 +6 -4) and the main program will print that 16 is the sum.

3. The main program will call a method named **printmyname()**, sending it one parameter -- the sum of the integer values calculated in step 2. The method will print your name that many times, and the method will not return a value. (Be careful – see below.)

For example, if the parameter to the method is 2, then the method should print your name 2 times. If the parameter is 4, the method should print your name 4 times.

However, if the parameter sent in is less than or equal to 0, or if the parameter sent in is greater than 12, the method will say it is not possible to print the name. For example, if the parameter is -2, the method will say in this case it is not possible to print the name since the number is negative. If the parameter is 15, the method will say it is not possible to print the name because it has to print it too many times.

4. The main program will call a method named **howmanyeven**, sending it all four integers. The method will determine how many (0,1,2,3 or 4 ) of the values are even and return that value to the main program. (Hint: Even means divisible by 2 with no remainder, and there is an operator that will tell you the remainder.)

The main program will print an appropriate message: There is/are \_\_ even number(s).

5. Then the main program will skip a few lines and go back to step 1

At step 1, if the user types in a special combination, the program will go to after step 5.

You must determine what this combination is, and you must explain it--in a prompt or in a comment, **and** in the introduction at the beginning--to the person using the program.

6. At the end, print how many sets of four data values were entered and processed. Make sure that this value is at least 8.

**DATA**: **You will be judged on the quality of your data.**

Type in a total of at least 8 sets of data values.

Have at least two sets where the sum is less than 1.

Have at least two sets where the sum is more than 14.

Have at least four sets where you do print your name one or more times.

Be sure to include both 1 and 10 as valid sets.

**STYLE**: Be sure that each method has a good comment explaining two things: exactly what parameter(s) the method will receive, and exactly what the method will do—for example, if it returns an answer or prints. For each method, mention the method's parameters by name in the comment.

**OUTPUT:** Here is some sample output (ignoring the introduction):

the four original integers are -5, 3, 8, 6

the sum is 22 (remember minus a minus 5 adds five)

it is not possible to print the name because it will print too many times

there is/are 2 even number(s)

the three original integers are 4, -1, -3, 0

the sum is 6

your name

your name

your name

your name

your name

your name (your name printed6 times)

there is/are 2 even number(s)

Optional

1. At the end, print the total number of times your name was printed.

2. At the end, print the largest value ever computed for the sum. Print the smallest value ever computed for the sum.