Lab Session #10

Part 1: Theory

1. Consider the following code: What will print? After you have come up with an answer, check your answer by putting it into the computer and running the program. Make sure you understand *why* the results are what they are.  
     
   public class Methods2 {

public static void main(String[] args) {

int x = 0;

System.out.println(x);

confusing(x);

System.out.println(x);

}

public static int confusing(int y) {

System.out.println(y);

y = y + 1;

System.out.println(y);

return y – 1;

}

}

1. Consider the following code: What will print? After you have come up with an answer, check your answer by putting it into the computer and running the program. Make sure you understand *why* the results are what they are.  
     
   public class Methods2 {

public static void main(String[] args) {

int x = 0;

System.out.println(x);

x = confusing(x);

System.out.println(x);

}

public static int confusing(int y) {

System.out.println(y);

y = y + 1;

System.out.println(y);

return y – 1;

}

}

1. Consider the following code: What will print? After you have come up with an answer, check your answer by putting it into the computer and running the program. Make sure you understand *why* the results are what they are.  
     
   public class Methods2 {

public static void main(String[] args) {

int x = 0;

int y = 2;

confusing(x + y);

}

public static int confusing(int x) {

return more(x \* x);

}

public static int more(int y) {

System.out.println(y);

return y;

}

}

1. Consider the following code: What will print? After you have come up with an answer, check your answer by putting it into the computer and running the program. Make sure you understand *why* the results are what they are.  
     
   public class Methods {

public static void main(String[] args) {

int x = 0;

int y = 1;

confusing(y);

}

public static void confusing(int x) {

if (x < 0) {

System.out.println(-x);

}

else {

System.out.println(x);

}

}

}

**Part 2: Application**

* 1. Write a program that takes as input one double value and prints the perimeter of a square, a regular pentagon, a regular hexagon and a regular octagon. You must create a method for each perimeter you calculate. The method must take 1 double value as parameter and return a double value containing the perimeter. That is, your program must contain four methods called *perimeterSquare, perimeterPentagon, perimeterHexagon, perimeterOctagon*.

In the main method, you will read an input from the user, write commands to call the four methods and get the four perimeter values in four variables and finally print the four values. Note that the printing statements must be in the main method and NOT the perimeter methods.

A sample execution of your program must look like the following:

Enter a value: 2

The perimeter of a square of side 2 is: 8.0

The perimeter of a regular pentagon of side 2 is: 10.0

The perimeter of a regular hexagon of side 2 is: 12.0

The perimeter of a regular octagon of side 2 is: 16.0

* 1. In lab 2, you wrote a flowchart to find the smallest of two numbers. Convert that algorithm into a program today. Write a method *minOfTwoIntegers* that takes two integer values as parameters and returns one integer value that contains the minimum of the two. Do NOT use the Math.min method. Use the if-else statements that we saw in class today to do so.
  2. In one of the previous labs you wrote a method *perimeterRightTriangle*. The method took two doubles as input, which are set from the main method. The method returned the perimeter of a right triangle with those two lengths as its side. (If you haven’t completed this, finish your previous lab first!)

Now add an *if-else* statement to your main method. You should only calculate the perimeter of the triangle if the first value the user entered is positive. If it is a negative number or zero, you should not do the calculation and should simply print “Invalid”  
  
Hint: You will need to check if the value they typed is > 0. This means you need an if statement and almost your entire code will be within that if statement. The else statement should have just 1 print statement to print Invalid.

Lastly, add a *second* if-else statement (within the first if statement), to check if the 2nd value they type is greater than 0.