**ECE25100 Object Oriented Programming**

|  |  |
| --- | --- |
| **Lab 1: Hello, Java World** |  |

**Description:**The purpose of this lab is to help you:

* get familiar with JDK and JCreator/Eclipse IDE.
* understand how to perform simple computations, format numbers for display purposes and make use of the **IF** statement.

To get credit for the lab, you need to demonstrate to the instructor/lab assistant that you have completed all the requirements, and sign your name on the grade sheet.

**QUESTION 1:**

Compile, run and test the examples from the lecture notes.

**QUESTION 2:**

Complete the code below (i.e., copy/paste into a new file called **StarDisplay.java**) to display a banner like this

\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*

\*\*

\*\*

\*\*

\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*

\*\*

\*\*

\*\*

\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*

**public class** StarDisplay {

public void displayStar( ){

// display the banner

}

}

Copy/paste the source code below into a new file called **StarDisplayTest.java** and test the class **StarDisplay**.

**public class** StarDisplayTest {

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

StarDisplay s = new StarDisplay( );

s.displayStar();

}  
}

**QUESTION 3:**

The area of an equilateral triangle is computed as follows (given that all sides have length **a**):

Complete the code below (i.e., copy/paste into new files called **AreaProgram.java** and **AreaProgramTest.java**) to compute the area of the triangle.   Test your code to see whether it is correct.   When **a=1** the area is around **0.433**, when **a=5** the area is around **10.825**.

**public class** AreaProgram {

**double** a;

**double** area;

public double computeArea( ){

// compute the area

}

}

**public class** AreaProgramTest {

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

AreaProgram areaProgram = new AreaProgram( );

System.out.println("The area is "+areaProgram.computeArea( ));

}

}

**QUESTION 4:**

Below is some code to display some monetary values.  Create new files called **MoneyProgram.java** and **MoneyProgramTest.java** and paste the code below in them, then compile and run the code:

**public class** MoneyProgram {

**float**a = 10.00f;  
**float**b = 20.20f;  
**float**c = 3.01f;  
**float**d = 0.12f;  
**float**e = 67.78923f;  
  
**public void** display(){

System.out.println(a);

System.out.println(b);

System.out.println(c);

System.out.println(d);

System.out.println(e);

}  
}

**public class** MoneyProgramTest {

**public static void** main(String args[]) {  
   MoneyProgram m = new MoneyProgram( );

m.display();  
   }  
}

The code produces the following output:

10.0  
20.2  
3.01  
0.12  
67.78923

Adjust the code so that it produces the nicely formatted output as shown below.   You may use the **printf** (see below for hint).

$10.00  
$20.20  
$3.01  
$0.12  
$67.79

Hint: Here are some examples of how you may do this:  
System.out.printf("printf displays it as %.2f   \n", 67.78923);

**QUESTION 5:**

Here is a simple piece of code that asks the user whether they are male or female.   Save, compile, run and test the code in two file called

**UnderstandingIFProgram.java and UnderstandingIFProgramTest.java**:

**public class** UnderstandingIFProgram{

**private char** input;

**private boolean** male;

public void question(){

System.out.print("Are you (M)ale or (F)emale ? ... ");  
input = **new** java.util.Scanner(System.in).nextLine().charAt(0);  
         
**if** ((input == 'M') || (input == 'm'))  
   male = **true**;  
**else**  
   male = **false**;  
              
**if** (male == **false**)  
   System.out.println("OK, now I know that you are a female.");  
**else**  
   System.out.println("OK, now I know that you are a male.");

}

}

**public class** UnderstandingIFProgramTest {

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

UnderstandingIFProgram u = new UnderstandingIFProgram();

u.question();

  }  
}

The code is not efficient.  Replace all of this code:

**if** ((input == 'M') || (input == 'm'))  
    male = **true**;  
**else**  
    male = **false**;

with this:

male = (input == 'M') || (input == 'm');

Can you explain why the code still works the same?   Now replace this line:

**if** (male == **false**)

with this one:

**if** (!male)

Do you understand why the code works the same way?   It is important to understand how booleans can be used efficiently with **IF** statements as shown here.

**QUESTION 6:**

Write a class named Robot that takes a point (x,y) to represent a robot's initial location (in centimeters) as well as its orientation (an angle called **theta** which is a value between 0 and 360 degrees).   You can either set these values in your code directly or ask the user for them (using the **Scanner** class).   They could all be integers.

Assume the robot moves forward **d** centimeters. Compute and display the final location of the robot after it moves. The new location is computed as point

(, ).

Remember that the sine and cosine functions in JAVA use radians, not degrees.

To convert degrees to radians, you need to multiply the degrees by **Math.PI/180** and if you need to convert from radians to degrees you can multiply the radians by **180/Math.PI**.

Write a class called RobotTest to test your code...remembering that 0 degrees is to the right (as shown below in red), 90 degrees is up, 180 degrees is left and the angle **theta** shown below is around 135 degrees.

