SOFE 2710U Object-Oriented Programming and Design
Fall 2017 (Dr. Q. Mahmoud & Dr. A. Abdalbari)

Individual Programming Assignment #1

The purpose of this assignment is to get you started with Java programming. You will design and develop Java programs to solve mathematical and engineering problems.

Due Date: By 11:59pm on Thursday, Oct 12, 2017
No late assignments will beaccepted!

Academic Integrity & Zero Tolerance for Plagiarism

This assignment is to be done individually. Academic integrity is everyone's responsibility and you should familiarize yourself with UOIT policy: http://academicintegrity.uoit.ca

The penalty for academic misconduct is a **mark of zero on the assignment** and following the academic misconduct process as outlined at: http://academicintegrity.uoit.ca/students/misconduct-process.php

It is important to note that the assignment is the same for all both sections of SOFE 2710, and the same Teaching Assistant will be marking assignments across both sections. To avoid academic misconduct, we highly recommend you follow the following instructions/guidelines:

- You may discuss assignment issues with other students in high-level and broad fashion.
- Do not look at anyone else's code on screen or paper.
- Do not consult other resources: Internet resources, people, books, etc.
- If you're stuck, consult the TAs or instructors during tutorials or office hours or make an appointment.

The assignment source code is to be submitted electronically, and the instructors and TAs may use online tools to check for plagiarism. We have zero tolerance for plagiarism.

IMPORTANT NOTE: **You must provide your own solutions to all three problems, but note that only two of your solutions will be selected randomly for grading. Plagiarism detection tools will be used against all of your solutions and not just the ones selected for grading.**

Question 1 (10 marks)

Figure 1 shows an electric circuit designed to measure the temperature of the water in a beaker.

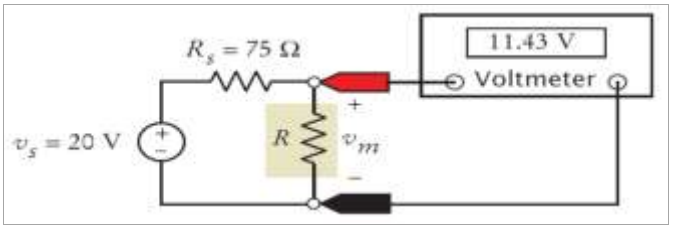

Figure 1: Electric circuit

The resistor *R* represents a temperature sensor enclosed in the beaker. The resistance *R*, in Ω, is related to the temperature *T*, in °C, by the equation

$$R = R_0 + kT$$

where $R_0 = 50$ and $k = 0.5$.

The voltmeter displays the value of the voltage across the sensor

$$v_m = \left(\frac{R}{R_s + R}\right) v_s$$

The voltage v_m indicates the temperature, *T*, of the water according to the equation

$$T = \frac{\frac{R_s}{k} v_m}{v_s - v_m} - \frac{R_0}{k} = \frac{150 v_m}{20 - v_m} - 100$$

For example, when $v_m = 11.43$ V, as shown in the figure, the temperature of the water is

$$T = \frac{150(11.43)}{20 - 11.43} - 100 = 100°C$$

Create a new Java class PrintVoltageWater that prints a table showing the meter voltage corresponding to water temperatures varying from 0 °C to 100 °C in increments of 10 degrees.

```
public class PrintVoltageWater {
   public static void main(String[] args) {
      // Write your code here
   }
}
```

Here is a sample output with no description message:

```
  T     vm
------  ------
  0     8.00
 10     8.46
 20     8.89
 30     9.29
 40     9.66
 50    10.00
 60    10.32
 70    10.63
 80    10.91
 90    11.18
100    11.43
```

Question 2 (10 marks)

The maximum load that can be placed at the end of a symmetrical wooden beam, such as the rectangular beam shown in Figure 2, can be calculated as the following:

$$L = \frac{S \times I}{d \times c}$$

L is the maximum weight in lbs of the load placed on the beam.

S is the stress in lbs/in^2.

I is the beam's rectangular moment of inertia in units of in^4.

d is the distance in inches that the load is placed from the fixed end of the beam (the "moment arm").

c is one-half the height in inches of the symmetrical beam.

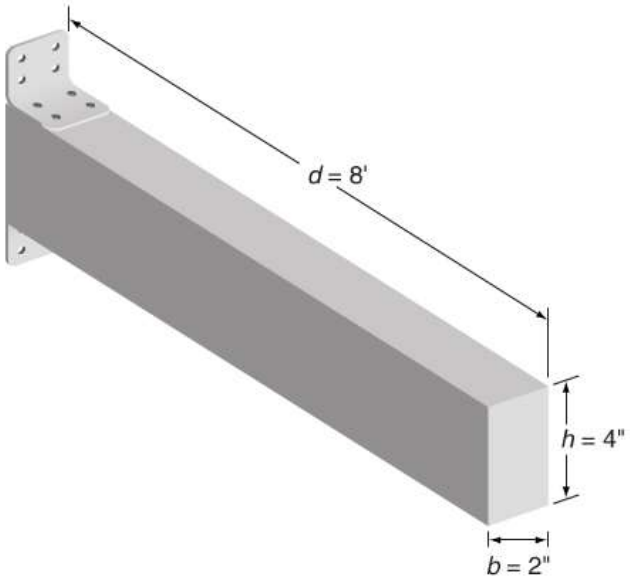

Figure 2: Calculating a symmetrical wooden beam's maximum load

For a 2" × 4" wooden beam, the rectangular moment of inertia is given by this formula:

$$I = \frac{base \times height^3}{12} = \frac{2 \times 4^3}{12} = 10.67^4$$

c = ½(4 in) = 2 in

Using this information, create a Java class (see below) with at least one method to compute the maximum load in lbs that can be placed at the end of an 8-foot 2" × 4" wooden beam so that the stress on the fixed end is 3000 lb/in^2. Your Java class should contain at least one method that is capable to computing the maximum load in lbs that can be placed at the end of a variety of wooden beans (e.g. 8-foot 2" x 4" or 3" x 6", etc.). Use the code below, and note that you may add instance variables, at least one constructor, and other methods.

```
public class WoodenBeam {
   double computeMaxLoad(double stress) {
     // Write your code here
   }
   public static void main(String[] args) {
      // Create a beam object using it's length, bade and height
      WoodenBeam wb1 = new WoodenBeam(8.0, 2.0, 4.0);
      WoodenBeam wb2 = new WoodenBeam(8.0, 3.0, 6.0);
      double maxload1 = wb1.computeMaxLoad(3000.0);
      double maxload2 = wb2.computeMaxLoad(3000.0);
      // print out details
   }
}
```

Sample Output with no description message:

For Beam #1:
The length of the beam is d = 8 feet which is equal to 96 inches
The length of the base is b = 2 inches
The height of the beam is h = 4 inches
The stress on the fixed end of the beam is S = 3000 lbs per square inch
The rectangular moment of inertia is: 10.67 in^4
The centroid of the beam is: 2 inches
Using the formula L = (S*I)/(d*c) = (3000*10.67)/(96*2) , the maximum load that can be placed at the end of the beam is: 166.67 lbs

For Beam #2:
The length of the beam is d = 8 feet which is equal to 96 inches
The length of the base is b = 3 inches
The height of the beam is h = 6 inches
The stress on the fixed end of the beam is S = 3000 lbs per square inch
The rectangular moment of inertia is: 54 in^4
The centroid of the beam is: 3 inches
Using the formula L = (S*I)/(d*c) = (3000*54)/(96*3) , the maximum load that can be placed at the end of the beam is: 562.5 lbs

Question 3 (10 marks)

Electric wire, like that in the photo, is a cylindrical conductor covered by an insulating material. The resistance of a piece of wire is given by the formula

$$R = \frac{\rho L}{A} = \frac{4\rho L}{\pi d^2}$$

where ρ is the resistivity of the conductor, and L, A, and d are the length, cross-sectional area, and diameter of the wire. The resistivity of copper is 1.678 x 10^{-8} Ω m. The wire diameter, d, is commonly specified by the American wire gauge (AWG), which is an integer, n. The diameter of an AWG n wire is given by the formula

$$d = 0.127 \times 92^{\frac{36-n}{39}} \text{ mm}$$

Write a Java class to compute and print out the resistance of copper and aluminum wire. Use the following class, and you add instance variables, constructors, and other methods.

```
public class WireResistance {
    // Takes the write gauge and returns the corresponding wire diameter
    double computeDiameter(int wireGauge) {
    }

    // Takes the length and gauge of a piece of copper wire and returns the resistance of that wire.
    double computeCopperWireResistance(double length, int wireGauge) {
    }
    // Takes the length and gauge of a piece of aluminum wire and returns the resistance of that wire. The resistivity
    // of aluminum is 2.82 x 10^-8 Ω m
    double computeAlumWireResistance(double length, int wireGauge){
    }

    public static void main(String[] args) {
    }
}
```

Your program should validate user input for the wire gauge and wire length (i.e. prevent computation of zero or negative wire gauge or length) and displays the precision of the output resistance to three decimal points. Here is a sample output with no description message:

```
Enter the wire gauge: 30
Enter the wire length in inches: 65
The resistance of a 65 inch piece of 30 gauge copper wire is 0.544 Ohms.
The resistance of a 65 inch piece of 30 gauge aluminum wire is 0.914 Ohms.
```

Grading Guidelines and Marking Rubrics

1) No late assignments will be accepted, so don't wait until the last minute to start working on your assignment. Students with no submissions will receive a mark of zero for this assignment.
2) If a program doesn't compile successfully, a mark of zero is assigned for that program; no partial marks for coding style or documentation for programs that don't compile.
3) Formatting and indenting your source code is important. Follow the programming style presented during lectures and tutorial#3. You must provide full Javadoc documentation for each of your classes.

Criteria	Marks	Description
Program Source Code	3	• Program is properly documented (including Javadoc) • Meaningful variable names, properly formatted source code • Appropriate programming techniques as discussed in class
User Friendliness	2	• Program description message • Program output is presented neatly on the screen
Functional and Correct	5	• Program is functional and produces correct output
Total	**10 x 2**	**Two programs will be randomly selected and graded**. • *Question 1 [10 marks]* • *Question 2 [10 marks]* • *Question 3 [10 marks]*

Submission Guidelines

The source code must be submitted on Github as per the instructions below. No email submissions will be accepted.

1) Create a Github account (https://github.com/join).
2) Register for Github Devpack in order to get private repositories (https://education.github.com/pack).
3) Go to the following link for Assignment1: https://classroom.github.com/a/AdaDTxZP
4) For additional instructions on how to use Github and submit your assignment, go to: https://github.com/UOITEngineering/student-classroom-assignments