Lab 17

Instructions: Complete the steps below. Be sure to upload a copy of all your source code (.java) files to the link on Brightspace by its deadline, so that you can receive credit for this lab.

1. (Triangle.java and Triangle.jpg) Design a class named Triangle.java that extends GeometricObject from the Inheritance lecture notes with:
    - Three double data fields named side1, side2, and side3 with default values 1.0 to denote three sides of the triangle.
    - A no-arg constructor that creates a default triangle.
    - A constructor that creates a triangle with the specified side1, side2, and side3.
    - A constructor that creates a triangle with the specified side1, side2, and side3, color and filled (from GeometricObject).
    - The accessor methods for all three data fields.
    - A method named getArea() that returns the area of this triangle(use Heron formula for the area of a triangle).
    - A method named getPerimeter() that returns the perimeter of this triangle.
    - A method named toString() that returns a string description for the triangle.

    Draw the UML diagram that involves the class Triangle and GeometricObject. Implement the class. Write a main method in Triangle that creates a Triangle object with sides 1, 1.5, 1, sets color yellow and filled true, and displays the area, perimeter, color, and whether filled or not.

2. (Person.java, Student.java, Employee.java, Faculty.java, Staff.java, TestPerson.java and PeopleUML.jpg) Design a class named Person.java and two of its subclasses named Student.java and Employee.java. Make Faculty.java and Staff.java subclasses of Employee.
    - A person has a name, address, phone number, age and email address and implement a method getAge():int.
    - A student has a class status (freshman, sophomore, junior, or senior - define these various statuses as constants).
    - An employee has a salary and date-hired of type MyDate. Define separately a class named MyDate.java that contains the fields year, month, and day.
    - A faculty member has office hours (as a String, e.g., "Mondays and Wednesdays 8:30-10am") and a rank (represented by a positive integer between 1 and 10), and it implements a method salaryRange():int that returns:
    - 1 if rank*years<10 (where years is current year minus date-hired year)
    - 2 if 10 <= rank*years < 20
    - 3 if 20 <= rank*years
    - A staff member has a title.

    Override the toString method in each class to display the class name and the person's name.
    Write a test program TestPerson.java with a main method that creates persons, students, employees, faculties, and staff, and invokes their toString() methods.
    Draw the UML diagram for the classes.

Grading Guidelines: This lab is graded on a scale of **0-3 points**, assigned as follows:

- **0 -** The student did not attend the lab,
- **3 -** The solutions are complete OR the student spent the entire lab solving the required lab problems (in this case, the students may not arrive at the lab after the lab started and may not leave until the lab ends).