Lab Performance-1: (5 Marks)

Create a Car class to represent a fan.

- The car speed is represented by three constants labeled LOW, NORMAL, and FAST, with values 1,2, and 3 respectively.
- The car speed is specified by an int data field called speed (default LOW).
- A boolean data field called car on that indicates whether or not the car is turned on (default false).
- The length and width of the car is specified by a double data field called car len (default 4) and car width (default 3).
- A color data field that describes the color of the car (default white).
- A constructor that builds a default car with no arguments.
- The car objects are initialized to provided values using a parameterized constructor.
- The car description will be shown using the display() function. When the car is turned on, the display() function shows the speed, color, and area of the car. If the car is turned off, the method returns the color and area of the car, as well as the message "car engine is switched off."

Create two Car objects with a test program. One with default values and the other with normal speed, length 6, width 4, brown color, and status true turned on. Show the descriptions for two Car objects that you've built.

Lab Performance-2: (5 Marks)

Create a Rectangle class that extends Geometric class. The class has three double data fields named length, width, and depth that have default values of 1.0.

- A default rectangle is created using a no-arg constructor.
- A rectangle with the given length, width, and depth is created with a parameterized constructor.
- Getter and Setter methods for all data fields.
- getArea() is a method that returns the area of this rectangle.
- The perimeter of this rectangle is returned by the getPerimeter() function.
- toString() is a method that returns a string description of the rectangle.

Create a Rectangle object with sides of 1, 1.5, 1, color yellow, and filled true, and display the area, perimeter, color, and whether it is filled or not.

Lab Performance-3: (5 Marks)

Create the User class and its two subclasses, Pupil and Job. Make Teacher and Administrative class for Job subclasses.

A user's information includes his or her name, address, phone number, and email address. A pupil's position is determined by his or her class (freshman, sophomore, junior, or senior). Assign a constant to the status. Each job has a title, a pay, and a start date. A professor has office hours and is assigned a rank. A title is given to a member of the administrative job. To show the class name and the user's name, override the toString function in each class.

Create a test program that calls the toString() methods from the objects of all classes.

Submission Instruction/ Answering Format:

- Write the Problem
- Complete Program
- Screenshot of Input/Output
- Rename file as **ID_Name_LP1_3.pdf**