Q1) design a class named **Rectangle** to represent a rectangle.
The class contains:
Two **double** data fields named **width** and **height** that specify the width and height of the rectangle. The default values are **1** for both **width** and **height**.

- A no-arg constructor that creates a default rectangle.
- A constructor that creates a rectangle with the specified **width** and **height**.
- A method named **getArea()** that returns the area of this rectangle.
- A method named **getPerimeter()** that returns the perimeter.

  Write a test program that creates two **Rectangle** objects—one with width **4** and height **40** and the other with width **3.5** and height **35.9**. Display the width, height, area, and perimeter of each rectangle in this order.

Q2) *The* Fan *class*) Design a class named **Fan** to represent a fan.
The class contains:
Three constants named **SLOW**, **MEDIUM**, and **FAST** with the values **1**, **2**, and **3** to denote the fan speed.

- A private **int** data field named **speed** that specifies the speed of the fan (the default is **SLOW**).
- A private **boolean** data field named **on** that specifies whether the fan is on (the default is **false**).
- A private **double** data field named **radius** that specifies the radius of the fan (the default is **5**).
- A string data field named **color** that specifies the color of the fan (the default is **blue**).
- The accessor and mutator methods for all four data fields.
- A no-arg constructor that creates a default fan.
- A method named **toString()** that returns a string description for the fan. If

  the fan is on, the method returns the fan speed, color, and radius in one combined string. If the fan is not on, the method returns the fan color and

radius along with the string “fan is off” in one combined string.

Write a test program that creates two **Fan** objects. Assign maximum speed, radius **10**, color **yellow**, and turn it on to the first object. Assign medium speed, radius **5**, color **blue**, and turn it off to the second object. Display the objects by invoking their **toString** method.

Q3) Implement a class **Student**. For the purpose of this exercise, a student has a name and a total quiz score. Supply an appropriate constructor and methods getName(), addQuiz(int score), getTotalScore(), and getAverageScore(). To compute the latter, you also need to store the number of quizzes that the student took.

Q4) Modify the Student class of Exercise E8.7 to compute grade point averages. Methods are needed to add a grade and get the current GPA. Specify grades as elements of a class Grade. Supply a constructor that constructs a grade from a string, such as "B+". You will also need a method that translates grades into their numeric values (for example, "B+" becomes 3.3).