- **(The Rectangle class)** 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.

  Implement the class. 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.

- **(Student grade averages)** Create Student class and Student objects for the following data:

	Grade 1	Grade 2	Grade 3	Grade 4	Grade 5
Jack	76	54	89	76	98
Robert	34	65	23	87	100
Edward	80	65	97	54	94
Eddie	63	75	33	75	87

- Create a method which returns average of the student.
  - `public int getAverage()`
- Create a method that prints Student information
  - `public void printStudent()`
- `Output: Name Grade1 Grade2 Grade3 Grade4 Grade5 Average`
- Create a method outside the class which sorts all the student objects according to the average
  - `public static void sort(Student[] students)`

- **(Books)**Create Book class with the following fields:

ISBN

Name

Author

Publisher

Price

Write the respective constructors, getters and setters for the class
Create another class with the main method; create an array of Book objects

Write methods that:

Sort and print

a. by ISBN
b. by Name
c. by Author
d. by Publisher
e. by Price

Search a book according to their fields (ISBN, Name, Author, Publisher)

- **(The MyInteger class)** Design a class named MyInteger. The class contains:

  An int data field named value that stores the int value represented by this object.

  A constructor that creates a MyInteger object for the specified int value.

  A get method that returns the int value.

  Methods isEven(), isOdd(), and isPrime() that return true if the value is even, odd, or prime, respectively.

Static methods isEven(int), isOdd(int), and isPrime(int) that return true if the specified value is even, odd, or prime, respectively.

Static methods isEven(MyInteger), isOdd(MyInteger), and isPrime(MyInteger) that return true if the specified value is even, odd, or prime, respectively.

Methods equals(int) and equals(MyInteger) that return true if the value in the object is equal to the specified value.

A static method parseInt(char[]) that converts an array of numeric characters to an int value.

A static method parseInt(String) that converts a string into an int value. Draw the UML diagram for the class. Implement the class. Write a client program that tests all methods in the class.