MIT ART DESIGN & TECHNOLOGY UNIVERSITY, PUNE

MIT COLLEGE OF MANAGEMENT (MITCOM) , PUNE

PROGRAMME: MASTER OF COMPUTER APPLICATION (MCA)

SEMESTER - II (January - April 2024)

ACADEMIC YEAR 2023-24

Laboratory Course Assignment

Course Code: 23MCAL201   Course Name: Lab I Web Technology Using Java

PRN NO: ADT23MGTM0813 Student Name: DANEJ MANSI AJAY

MCA Year: 1ST YEAR   SEM: 2ND   DIVISION: A

Dr. Sangita Phunde	**Dr. Vijaya Gondane**	**Prof. Dr. Sunita Karad**
HOD, MCA	**PG HEAD Director-MITCOM**	**Executive**

Name and sign of External Examiner:

Name and sign of Internal Examiner:

Date of Examination:

DECLARATION

I hereby declare that the Journal of assignments solved by me and it is executed as per the course requirement of MCA program of MIT-ADT University, Pune.

The practical assignment work in this report has not been submitted to any other University or Institute for the award of any degree or diploma. This is my own and original work.

Place: MITCOM, Pune

Name of the Student: DANEJ MANSI AJAY

Signature of the Student:

Date:

MIT Art, Design & Technology University, Pune

MIT College of Management
MCA Department
JAVA ASSIGNMENT LIST
MCA – I SEM – II (January – April 2024)

PRN No: ADT23MGTM0813 **Name: - DANEJ MANSI AJAY**

Sr.No	Assignment no-1	DATE
1.	Write a program to list all even numbers less than or equal to the number n. Take the value of n as input from user.	
2.	Define a class Rectangle with its length and breadth. Provide appropriate constructor(s), which gives facilty of constructing rectangle object with default values of length pf breadth as 0 or passing value of length and breadth externally to constructor. Provide appropriate accessor & mutator methods to Rectangle class. Provide methods to calculate area & to display all information of Rectangle. Design different class TestRectangle class in separate source file, which will contain main function. From this main function, create 5 Rectangle objects by taking all necessary information from the user.	
3.	Create a class Book which describes its Book_title and Book_price. Use getter and setter methods to get & set the Books description. Implement createBooks and showBooks methods to create n objects of Book in an array. Display the books along with its description as follows:- Book Title Price Java Programming Rs.350.50 Let Us C Rs.200.00 Note: createBooks & showBooks should not be member functions of Book class.	
4.	Modify the program which is created in assignment 2 as follows The class has attributes **length** and **width**, each of which defaults to 1. It should have member functions that calculate the perimeter and area of the rectangle. It should have set and get functions for both length and width. The set functions should verify that length and width are each floating-point numbers larger than 0.0 and less than 20.0	
5.	Create a class Date for manipulating dates. Provide a constructor that enables an object of this class to be initialized when it is declared (You can select any	

default values for the day, month & year, e.g. your birth date). Provide the necessary functionality to perform error checking on the initializer values for data members day, month, and year. Also, provide a member function to add an integer in a date to obtain a new date.

Design separate class Employee which will have following information.

Employee Number	Number
Employee Name	Text
Joining Date	Date

Provide appropriate constructor(s) & methods to this class. Provide main function which will create 5 objects of Employee class.

6. Write a program that takes a String through Command Line argument and display the length of the string. Also display the string into uppercase and check whether it is a palindrome or not. (Refer Java API Documentation)

7. Write a program that accepts two numbers from the Command Line and prints them out. Then use a *for loop* to print the next 13 numbers in the sequence where each number is the sum of the previous two. For example:

```
input> java prob2 1 3
     output> 1 3 4 7 11 18 29 47 76 123 322 521 843 1364
```

8. Write a program that accepts two numbers in the range from 1 to 40 from the Command Line. It then compares these numbers against a single dimension *array* of five integer elements ranging in value from 1 to 40. The program displays the message *BINGO* if the two inputted values are found in the array element. For example:

```
      input>java prob3 3 29
      output>Your first number was 3
      Your second number was 29
Its Bingo! // this message if 3 and 29 is found in the array
Not  Found!  //  this  message  if  3  and  29  is  not  found  in
  the         //array
The array was 7 25 5 19 30
```

9. Write a program that allows you to create an integer *array* of 18 elements with the following values: *int A[]={3,2,4,5,6,4,5,7,3,2,3,4,7,1,2,0,0,0}*. The program computes the sum of element 0 to 14 and stores it at element 15, computes the average and stores it at element 16 and identifies the smallest value from the array and stores it at element 17

10.	Create a class Term. This class represents a term of a polynomial such as $2x^4$ where 2 is coefficient and 4 is exponent of the term. <u>Data members</u>:- int coefficient int exponent Create another class Polynomial. The internal representation of a polynomial is an array of Terms. The size of this array should be fixed. Provide a constructor for this class that will set all terms of a polynomial object as zero (where coefficient is 0 and exponent is 0). Provide following functions: setTerm(int, int) – Setting a term of a polynomial object. Each successive call of this function should set next term of the polynomial object. It should do the following validations:- Whether the exponent of the term being set is already used. Whether the array size limit is exceeded. Whether the exponent is negative. In all the cases it should not set the term and display an appropriate message. sort() – to arrange the terms in ascending order of exponents. Provide a function to print a polynomial object	
11.	Create a class Matrix. Internal representation of this class will be a two dimensional array of size 10 by 10. In addition, the class should have following data members and member functions: <u>Data members</u>:- int rows int columns Constructors - The default constructor Matrix() - This should set each of the array element to zero. Overloaded constructor Matrix(int, int) - This constructor should call the default constructor first. It should then assign the value of first parameter to variable rows, and the value of the second parameter to variable columns. You can assume that the values of both the parameters will be less than or equal to 10. Member functions - void setElement(int r, int c, int value) - This function should set the array element at row r and column c to the value val. This assignment should be done only if val is positive r and c are valid else the element should be set to zero. Matrix transpose () – This function should transpose the matrix. Transpose of	

	a matrix is another matrix where the elements in rows of the first matrix become elements of the corresponding columns in the new matrix. Provide a function to print a Matrix object.	
12.	Create a class called complex for performing arithmetic operations with complex numbers. Use floating point variables to represent the private data of the class. Provide a default constructor that initializes the object with some default values. Provide public member functions for each of the following ● Addition of two complex numbers: It returns the result obtained by adding the respective real parts and the imaginary parts of the two complex numbers. ● Subtraction of two complex numbers: It returns the result obtained by subtracting the respective real parts and the imaginary parts of the two complex numbers. ● display() – It displays the complex number in a+bi format. The output should be displayed as follows: - Sum of a1+b1 i & a2+b2 i is : a3+b3 i	

SR.NO	Assignment no-2	DATE
1.	Create an abstract class Instrument which is having the abstract function play. Create three more sub classes from Instrument which is Piano, Flute, Guitar. Override the play method inside all three classes printing a message “Piano is playing tan tan tan tan ” for Piano class “Flute is playing toot toot toot toot” for Flute class “Guitar is playing tin tin tin ” for Guitar class You must not allow the user to declare an object of Instrument class. Create an array of 10 Instruments. Assign different type of instrument to Instrument reference. Check for the polymorphic behavior of play method. Use the instanceof operator to print that which object stored at which index of instrument array.	
2.	Create an abstract class Compartment to represent a rail coach. Provide a abstract function notice in this class. Derive FirstClass, Ladies, General, Luggage classes from the compartment class. Override the notice function in each of them to print notice suitable to the type of the compartment. Create a class TestCompartment . Write main function to do the following: Declare an array of Compartment pointers of size 10. Create a compartment of a type as decided by a randomly generated integer in the range 1 to 4. Check the polymorphic behavior of the notice method.	
3.	Create a class Medicine to represent a drug manufactured by a pharmaceutical company. Provide a function displayLabel() in this class to print Name and address of the company. Derive Tablet, Syrup and Ointment classes from the Medicine class. Override the displayLabel() function in each of these classes to print additional information suitable to the type of medicine. For example, in case of tablets, it could be “store in a cool dry place”, in case of ointments it could be “for external use only” etc. Create a class TestMedicine . Write main function to do the following: Declare an array of Medicine references of size 10 Create a medicine object of the type as decided by a randomly generated integer in the range 1 to 3. Refer Java API Documentation to find out random generation feature. Check the polymorphic behavior of the displayLabel() method.	
4.	Write a program that accepts two numbers and a operator like (+,-,*, /) as command line arguments and perform the appropriate operation indicated by operator. If the user enters any other character the appropriate message will be displayed.	

	The output of the program should be displayed to the user.	
5.	Create a class Car which contains members speed, noOfGear. The class has a method drive() which is responsible to provide starting speed and noOfGears to a Car. Implement display() method which will display all attributes of Car class. The class SportCar is derived from the class Car which adds new features AirBallonType. When this method is invoked, initial speed and gear status must be displayed on console. Override the display method which display all attribute of the SportCar. Make use of super class display() method.	

Sr.No	Assignment-No. 3	DATE

1.	Create a package techm.itp.hyd<your batch id>.cs/ncs<your emp id>.
	<your first name>.
	For e.g., techm.itp.hyd10001.cs35123.Anu
	Now create a Greeter class in this package having the following features:
	
	Attributes:
	name string //indicates name of the person to be greeted
	Member functions:
	Greeter(aName) //constructor to initialize the name of the
	//person to be greeted by this greeter.
	sayHello() //returns a hello message with the name of the
	//person initialized earlier.
	sayGoodBye() //bids goodbye to the person named earlier.
	
	Create another class in the same package called Advisor that has the following features:
	Attributes:
	message string[5] //contains five advice messages
	Member functions:
	Advisor() //default constructor to initialize an array of
	//strings with atleast five advice messages
	getAdvice() //randomly selects an advice from the available
	//list of messages and returns it to the caller of //this method
	Outside the package, from your working directory, create a class GreeterTest that constructs Greeter objects for all command-line arguments and prints out the results of calling sayHello().
	The program should then display an advice and finally bid goodbye to each of the persons/entities in reverse order of the names entered at the command line.
	
	For e.g.,
	java GreeterTest Mars Venus
	then the program should print
	Hello, Mars!
	Hello, Venus!
	Advice: Never say No
	Goodbye Venus!
	Goodbye Mars!	

2.	Create a package esg.itp.shape containing the following classes and interface An interface Polygon containing the members as given below: area float perimeter float void calcArea( ); abstract method to calculate area of a particular polygon given its dimensions void calcPeri( ); abstract method to calculate perimeter of a particular polygon given its dimensions void display( ); method to display the area and perimeter of the given polygon Create a class Square that implements Polygon and has the following member: side float Square(float s); constructor to initialize side of square	

	Create another class Rectangle that implements Polygon and has the following member: length float breadth float Rectangle(int len, int bre); constructor to initialize length and breadth of a rectangle Outside the package, create a class that imports the above package an instantiates an object of the Square class and an object of the Rectangle class. Call the above methods on each of the classes to calculate the area and perimeter given the side and the length/breadth of the Square class and the Rectangle class respectively	
3.	Create a class called CalcAverage that has the following method: public double avgFirstN(int N) This method receives an integer as a parameter and calculates the average of first N natural numbers. If N is not a natural number, throw an exception IllegalArgumentException with an appropriate message.	
4.	Create a class Number having the following features: Attributes int first number int second number result double stores the result of arithmetic operations performed on a and b Member functions Number(x, y) constructor to initialize the values of a and b add() stores the sum of a and b in result sub() stores difference of a and b in result mul() stores product in result div() stores a divided by b in result Test to see if b is 0 and throw an appropriate exception since division by zero is undefined. Display a menu to the user to perform the above four arithmetic operations.	

5.	Create a class BankAccount having the members as given below:
	accNo integer
	custName string
	accType string (indicates ‘Savings’ or ‘Current’)
	balance float
	Include the following methods in the BankAccount class:
	void deposit(float amt);
	void withdraw(float amt);
	float getBalance();
	deposit(float amt) method allows you to credit an amount into the current balance. If amount is negative, throw an exception NegativeAmount to block the operation from being performed.
	withdraw(float amt) method allows you to debit an amount from the current balance. Please ensure a minimum balance of Rs. 1000/- in the account for savings account and	

	Rs. 5000/- for current account, else throw an exception InsufficientFunds and block the withdrawal operation. Also throw an exception NegativeAmount to block the operation from being performed if the amt parameter passed to this function is negative. getBalance() method returns the current balance. If the current balance is below the minimum required balance, then throw an exception LowBalanceException accordingly. Have constructor to which you will pass, accno, cust_name, acctype and initial balance. And check whether the balance is less than 1000 or not in case of savings account and less than 5000 in case of a current account. If so, then raise a LowBalanceException. In either case if the balance is negative then raise the NegativeAmount exception accordingly.	
6.	Create a class with following specifications. Class Emp empId int empName string designation string basic double hra double readOnly Methods printDET() calculateHRA() printDET() methods will show details of the EMP. calculateHRA() method will calculate HRA based on basic. There will 3 designations supported by the application. If designation is "Manager" - HRA will be 10% of BASIC if designation is "Officer" - HRA will be 12% of BASIC if category is "CLERK" - HRA will be 5% of BASIC Have constructor to which you will pass, empId, designation, basic and price. And checks whether the BASIC is less than 500 or not. If it is less than 500 raise a custom Exception as given below Create LowSalException class with proper user message to handle BASIC less than 500.	
7.	Create a class USERTRAIL with following specifications. val1, val2 type int Methods boolean show () will check if val1 and val2 are greater or less than Zero Have constructor which will val1, val2 and check whether if it is less than 0 then raise a custom Exception (name: Illegal value exception.)	

SR.NO	Assignment .no. 4	DATE
1.	Write a program which take source file and destination file as input as command line arguments. It copies the source file contents to destination file. If source file does not exist, it should give appriate message to use. If destination file does not exist, it should be created. If it exists, program should ask that, “whether you want to overwrite?(Yes/No”). On the basis of user choice, appropriate action should be taken. Note: Files may be any type of files like bitmap files, exe files, text files etc.	
2.	Write a stream based program which will accept Roll Number, Name, Age and Address from user Age and Roll-no should be numeric. Handle with built-in exception. None of the field should be blank. Handle with custom exception, Ask user ,whether to write the data in the file If answer is yes then data is saved into a file as an object(User can write many records in the file), otherwise terminate the current program Write another program to display all the records saved into the file	
3.	Write a program using java file system to copy the contents of one file into another. (This is Self Study Assignment. Refer Java API documentation.)	
4.	Write a program which will accept an input String from user (This is Self Study Assignment. Refer Java API documentation.) Write the input in the file io.txt Show size of the file Read contets from the file and display them on console Delete io.txt file using File class	
5.	Accept a directory name in form of String from user using proper IO stream. Store it in a variable. (This is Self Study Assignment. Refer Java API documentation.) Search if it exists in your system. If it exists then show all the files present in the directory otherwise print the message that directory Does not Exists	
6.	Create a class “CDR” with the following members: A_Number B_Number duration calculatedCharge Write a program which will accept the A_Number, B_Number and duration of Call from user(Duration is in minutes). Rate that call using 1 Rupee per min rate and store the calculated charge in “calculatedCharge.txt” file. Write this object to “rated_cdr.txt”. Note : CDR means Call Details Record	
7.	Identify that, what functionality need to be added in above assignment , so that it will require object externalization. Take approval for this functionality from the faculty. Then implement this functionality using Object Externalization.	

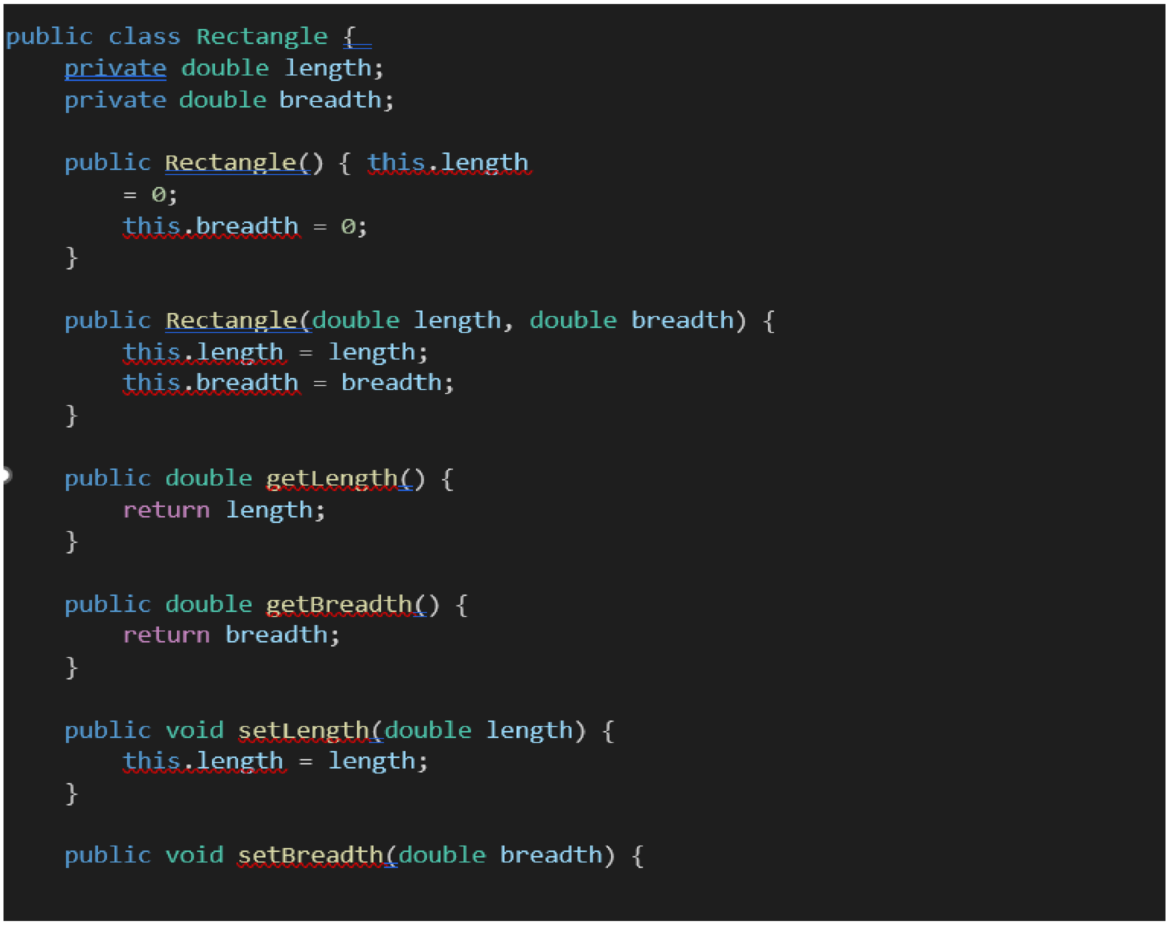
ASSIGNMENTS NO. 1

Objective: At the end of the assignments, participants will be able to create classes and write programs using objects, arrays, command-line arguments, Strings.

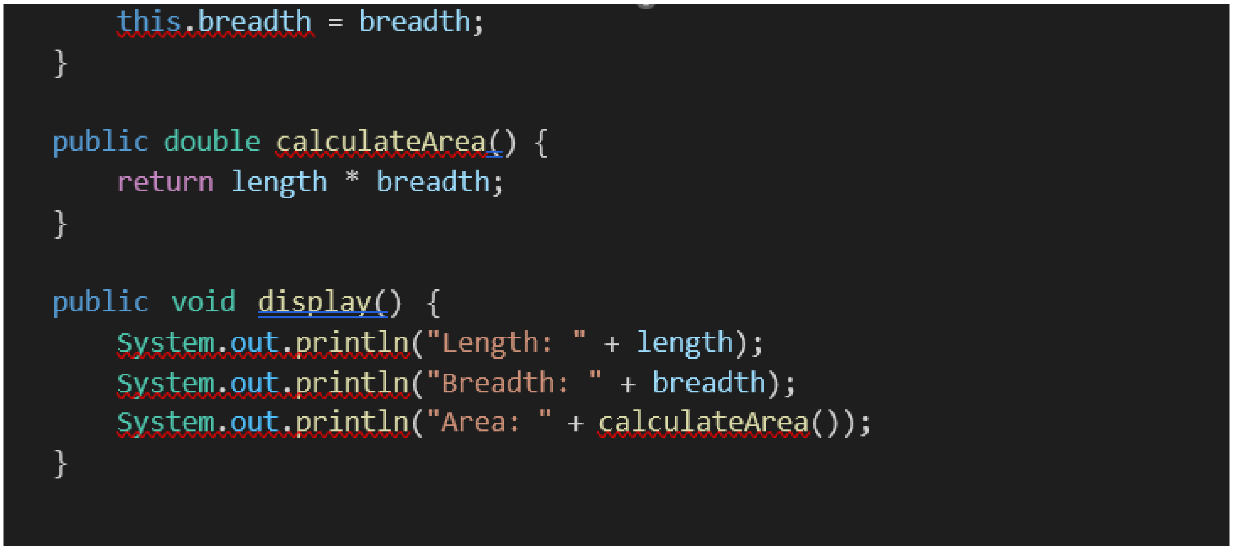
Java: Class, Object and Methods using Conditional statements and Loops.

Q.1 Write a program to list all even numbers less than or equal to the number n. Take the value of n as input from user.

CODE:

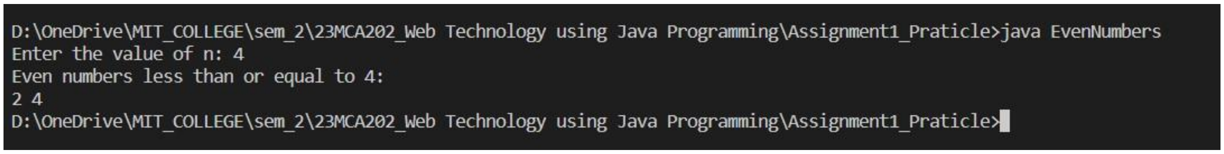


```
public class Rectangle {
    private double length;
    private double breadth;

    public Rectangle() { this.length
        = 0;
        this.breadth = 0;
    }

    public Rectangle(double length, double breadth) {
        this.length = length;
        this.breadth = breadth;
    }

    public double getLength() {
        return length;
    }

    public double getBreadth() {
        return breadth;
    }

    public void setLength(double length) {
        this.length = length;
    }

    public void setBreadth(double breadth) {
```



```java
        this.breadth = breadth;
    }

    public double calculateArea() {
        return length * breadth;
    }

    public void display() {
        System.out.println("Length: " + length);
        System.out.println("Breadth: " + breadth);
        System.out.println("Area: " + calculateArea());
    }
```

OUTPUT:



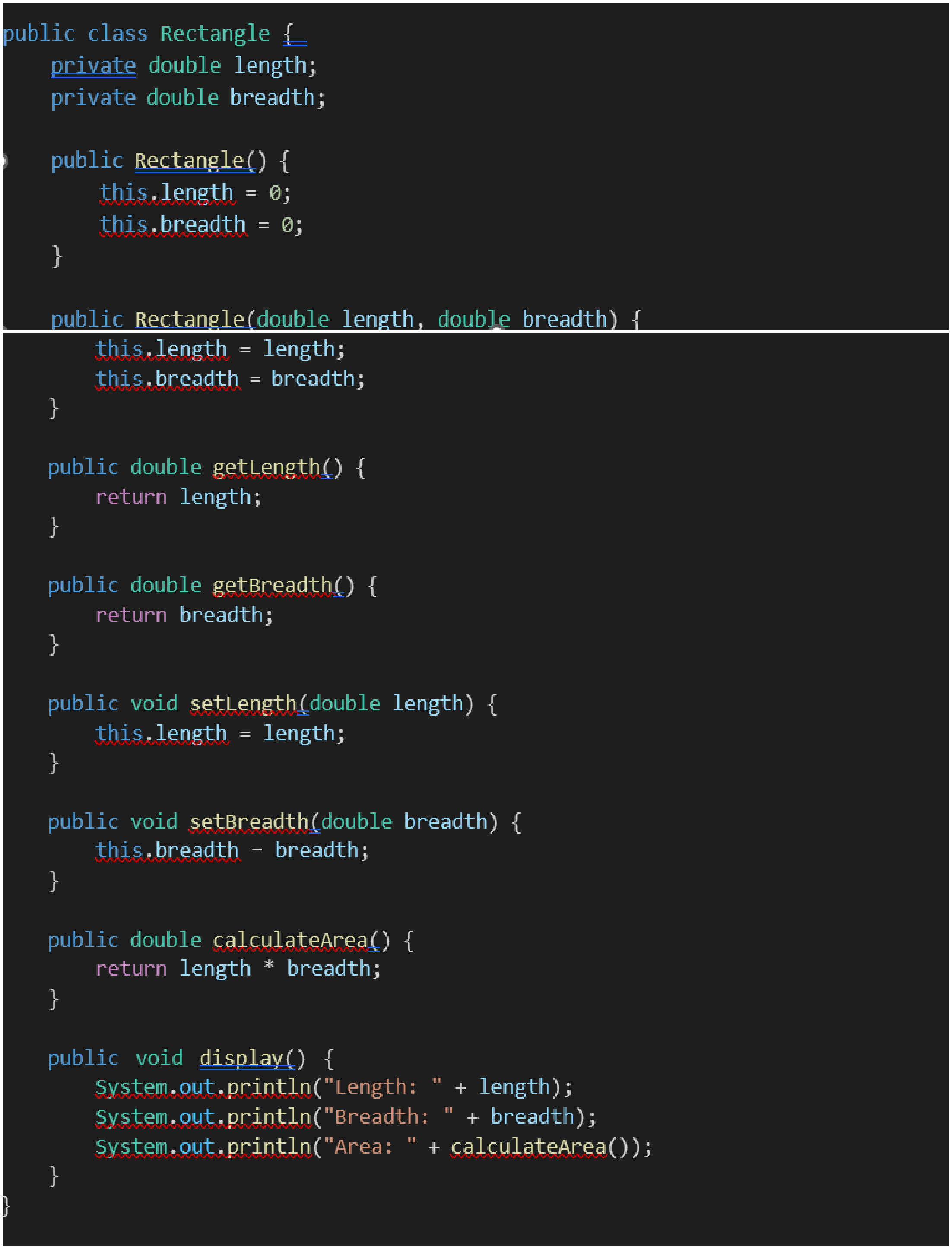
```
D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment1_Praticle>java EvenNumbers
Enter the value of n: 4
Even numbers less than or equal to 4:
2 4
D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment1_Praticle>
```

Q.2 Define a class Rectangle with its length and breadth.

Provide appropriate constructor(s), which gives facility of constructing rectangle object with default values of length pf breadth as 0 or passing value of length and breadth externally to constructor.

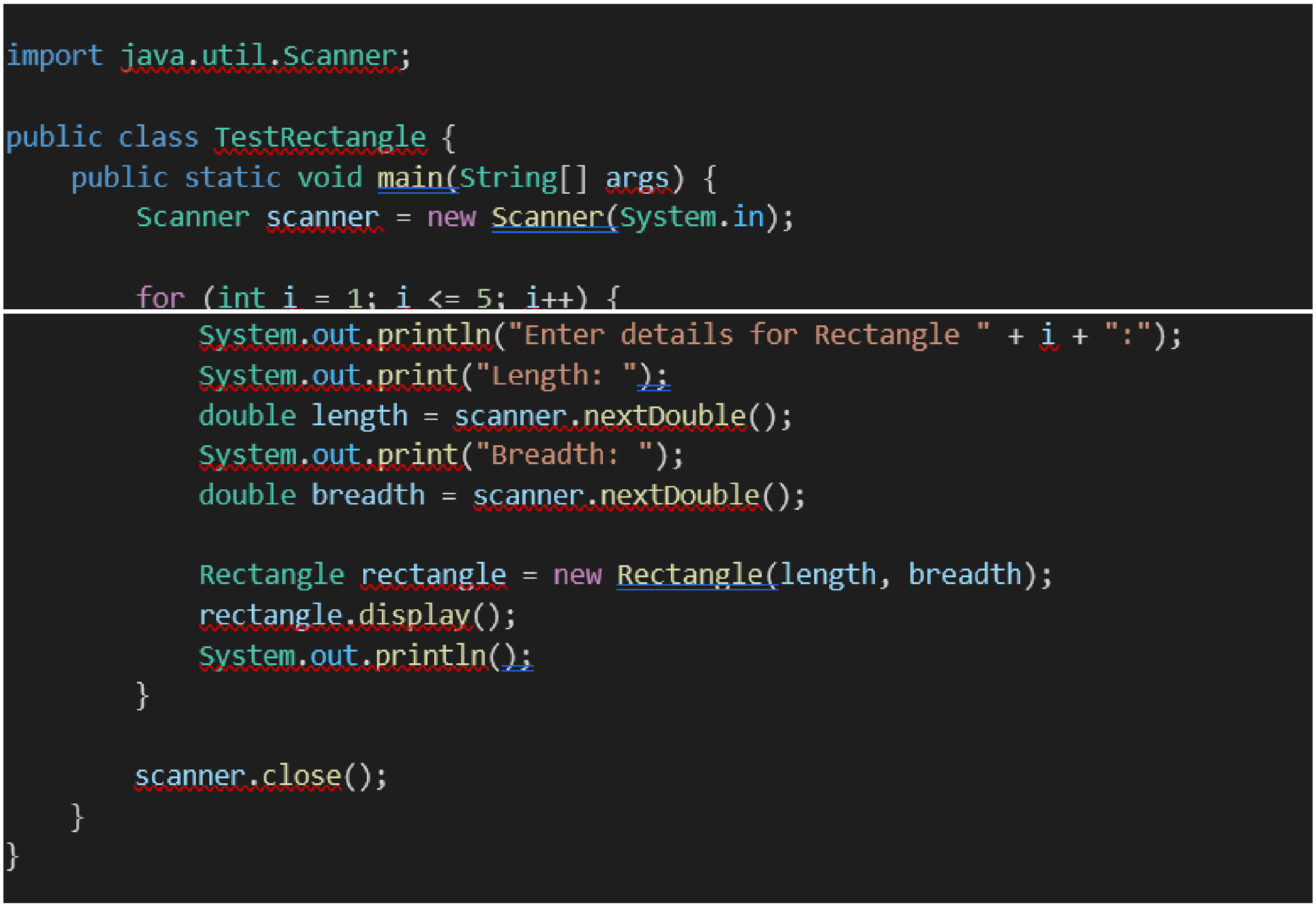
Provide appropriate accessor & mutator methods to Rectangle class. Provide methods to calculate area & to display all information of Rectangle.

CODE:



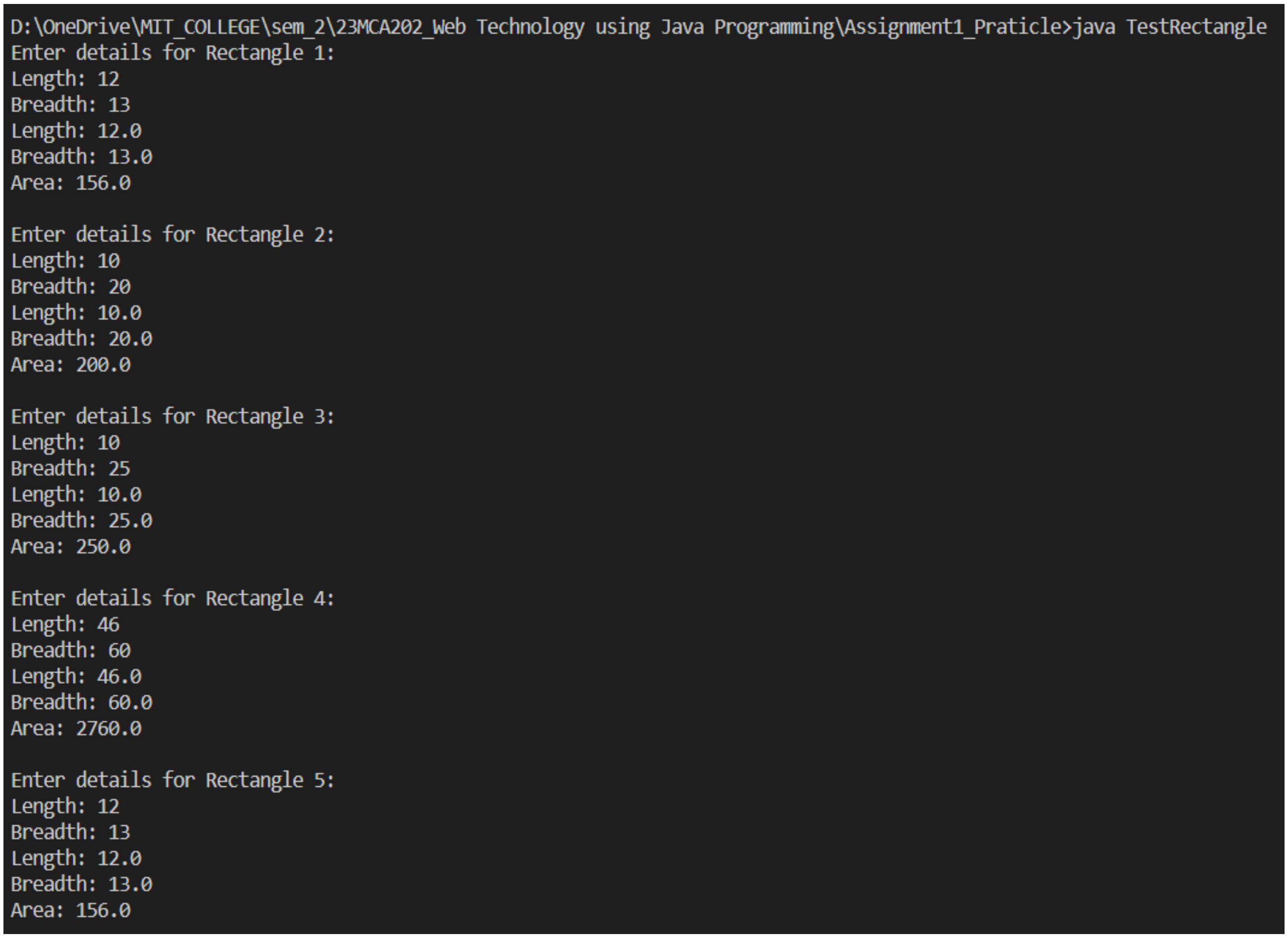
```java
public class Rectangle {
    private double length;
    private double breadth;

    public Rectangle() {
        this.length = 0;
        this.breadth = 0;
    }

    public Rectangle(double length, double breadth) {
        this.length = length;
        this.breadth = breadth;
    }

    public double getLength() {
        return length;
    }

    public double getBreadth() {
        return breadth;
    }

    public void setLength(double length) {
        this.length = length;
    }

    public void setBreadth(double breadth) {
        this.breadth = breadth;
    }

    public double calculateArea() {
        return length * breadth;
    }

    public void display() {
        System.out.println("Length: " + length);
        System.out.println("Breadth: " + breadth);
        System.out.println("Area: " + calculateArea());
    }
}
```

Design different class TestRectangle class in separate source file, which will contain main function. From this main function, create 5 Rectangle objects by taking all necessary information from the user.



```java
import java.util.Scanner;

public class TestRectangle {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        for (int i = 1; i <= 5; i++) {
            System.out.println("Enter details for Rectangle " + i + ":");
            System.out.print("Length: ");
            double length = scanner.nextDouble();
            System.out.print("Breadth: ");
            double breadth = scanner.nextDouble();

            Rectangle rectangle = new Rectangle(length, breadth);
            rectangle.display();
            System.out.println();
        }

        scanner.close();
    }
}
```

OUTPUT:



```
D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment1_Praticle>java TestRectangle
Enter details for Rectangle 1:
Length: 12
Breadth: 13
Length: 12.0
Breadth: 13.0
Area: 156.0

Enter details for Rectangle 2:
Length: 10
Breadth: 20
Length: 10.0
Breadth: 20.0
Area: 200.0

Enter details for Rectangle 3:
Length: 10
Breadth: 25
Length: 10.0
Breadth: 25.0
Area: 250.0

Enter details for Rectangle 4:
Length: 46
Breadth: 60
Length: 46.0
Breadth: 60.0
Area: 2760.0

Enter details for Rectangle 5:
Length: 12
Breadth: 13
Length: 12.0
Breadth: 13.0
Area: 156.0
```

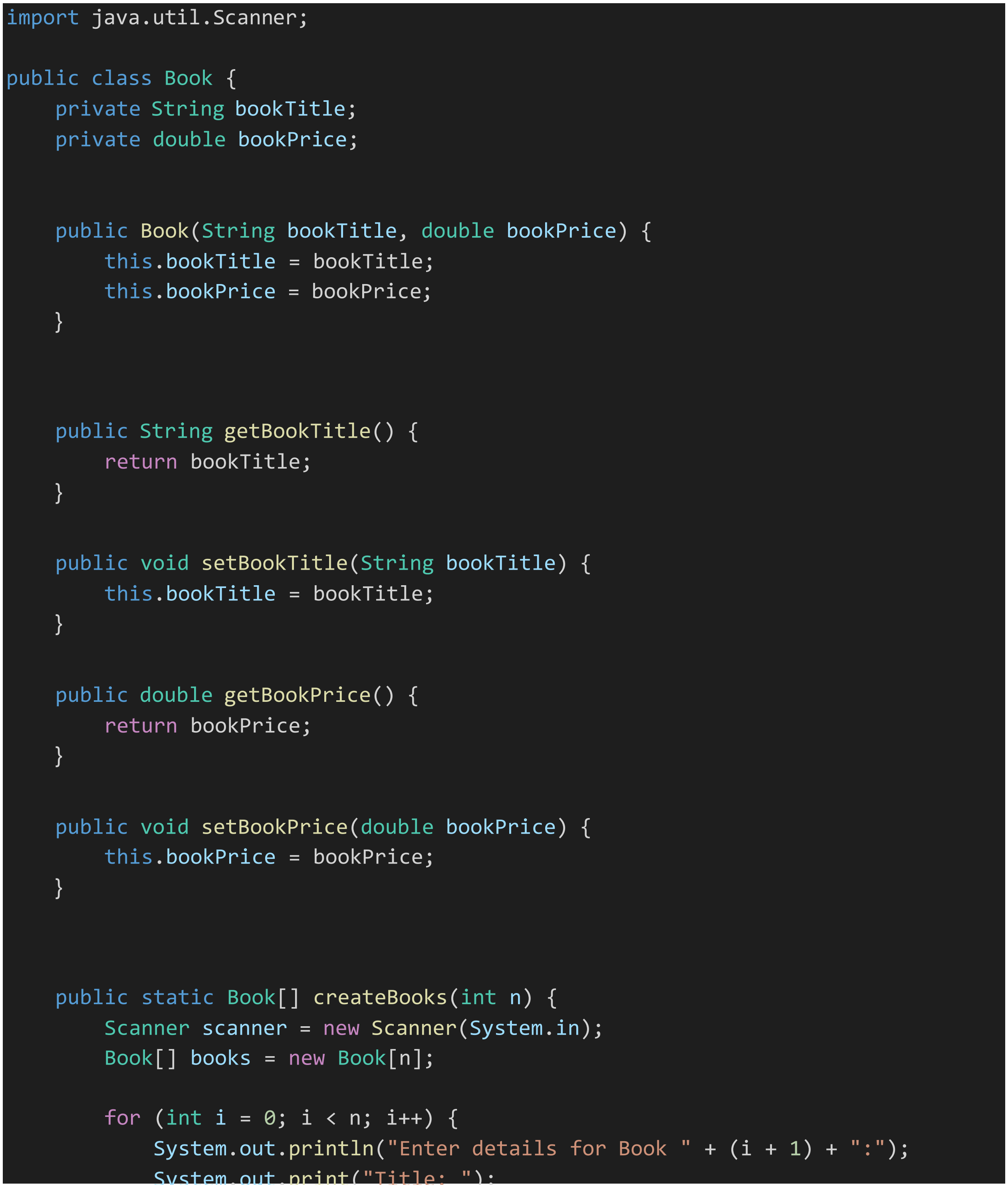
Q.3 Create a class Book which describes its Book_title and Book_price. Use getter and setter methods to get & set the Books description.

Implement createBooks and showBooks methods to create n objects of Book in an array. Display the books along with its description as follows: -

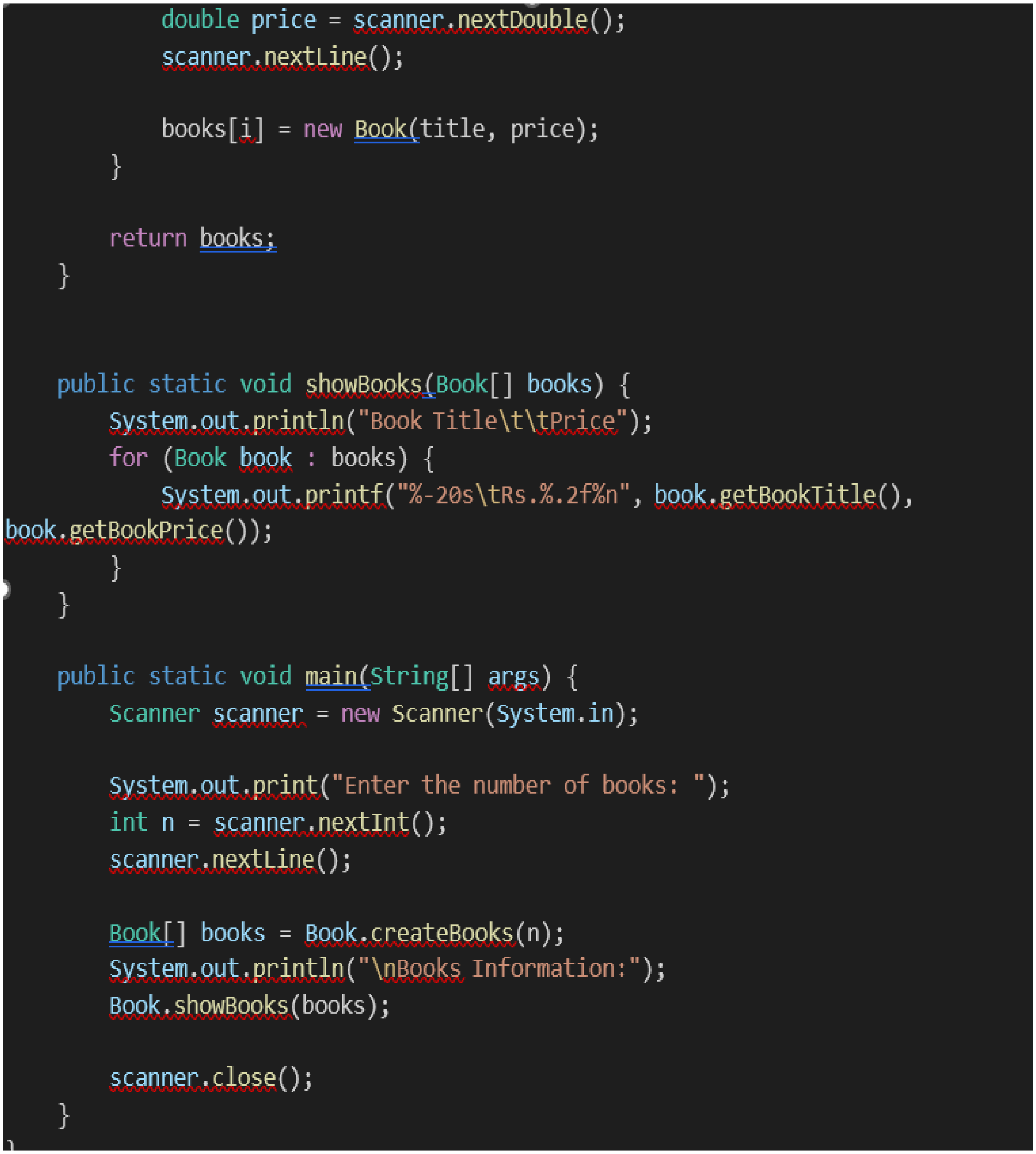
Book Title	**Price**
Java Programming	**Rs.350.50**
Let Us C	**Rs.200.00**

Note : create Books & show Books should not be member functions of Book class

CODE:

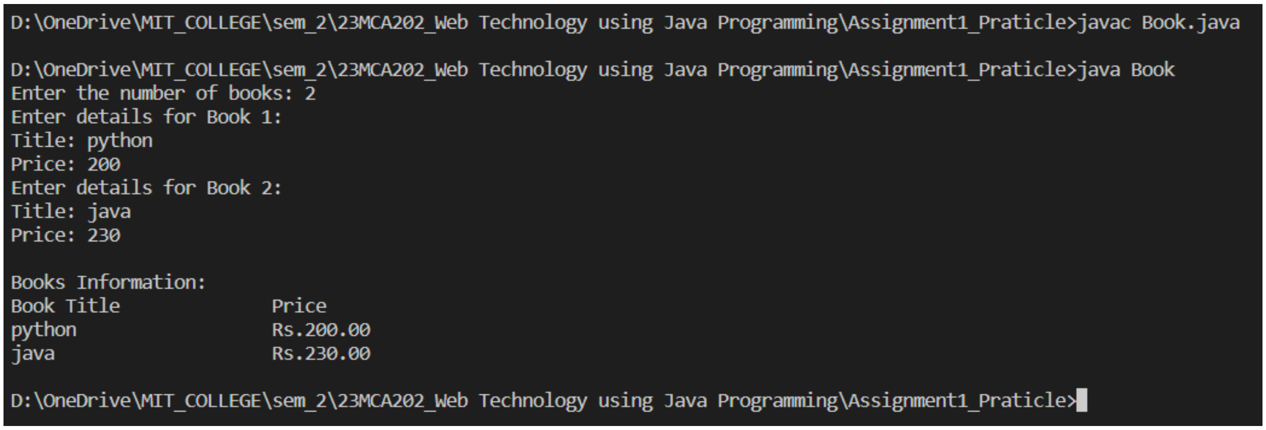


```
import java.util.Scanner;

public class Book {
    private String bookTitle;
    private double bookPrice;


    public Book(String bookTitle, double bookPrice) {
        this.bookTitle = bookTitle;
        this.bookPrice = bookPrice;
    }



    public String getBookTitle() {
        return bookTitle;
    }

    public void setBookTitle(String bookTitle) {
        this.bookTitle = bookTitle;
    }

    public double getBookPrice() {
        return bookPrice;
    }

    public void setBookPrice(double bookPrice) {
        this.bookPrice = bookPrice;
    }



    public static Book[] createBooks(int n) {
        Scanner scanner = new Scanner(System.in);
        Book[] books = new Book[n];

        for (int i = 0; i < n; i++) {
            System.out.println("Enter details for Book " + (i + 1) + ":");
            System.out.print("Title: ");
```



```
            double price = scanner.nextDouble();
            scanner.nextLine();

            books[i] = new Book(title, price);
        }

        return books;
    }


    public static void showBooks(Book[] books) {
        System.out.println("Book Title\t\tPrice");
        for (Book book : books) {
            System.out.printf("%-20s\tRs.%.2f%n", book.getBookTitle(),
book.getBookPrice());
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of books: ");
        int n = scanner.nextInt();
        scanner.nextLine();

        Book[] books = Book.createBooks(n);
        System.out.println("\nBooks Information:");
        Book.showBooks(books);

        scanner.close();
    }
}
```

OUTPUT:

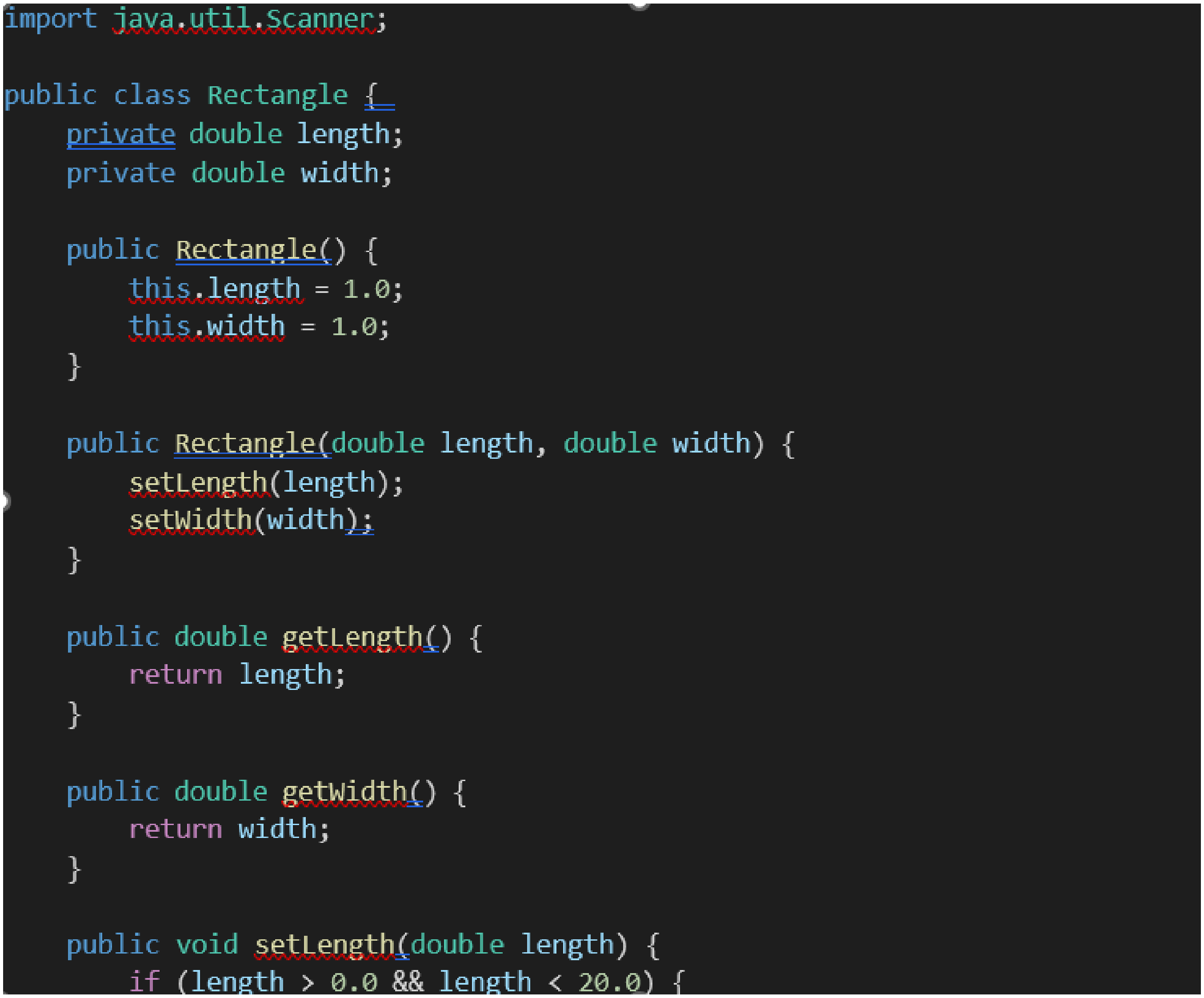


```
D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment1_Praticle>javac Book.java

D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment1_Praticle>java Book
Enter the number of books: 2
Enter details for Book 1:
Title: python
Price: 200
Enter details for Book 2:
Title: java
Price: 230

Books Information:
Book Title              Price
python                  Rs.200.00
java                    Rs.230.00

D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment1_Praticle>
```

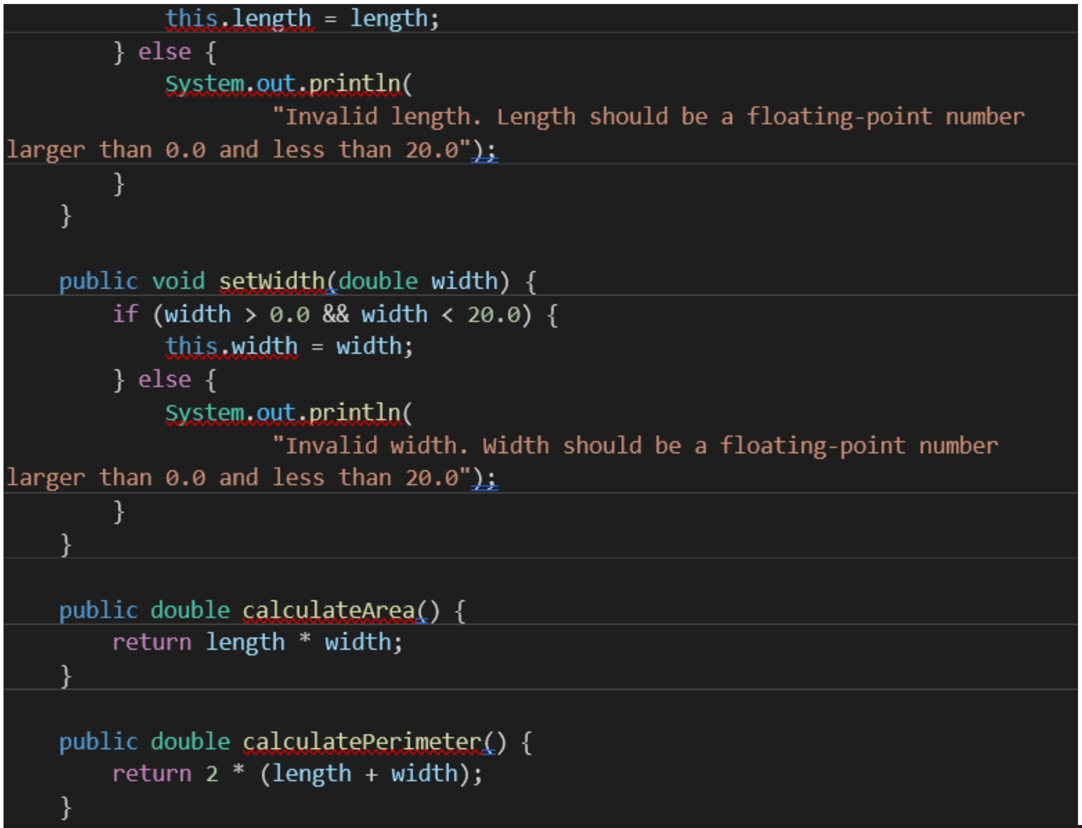
Q.4 **Modify the program which is created in assignment 2 as follows**

The class has attributes length and width, each of which defaults to 1. It should have member functions that calculate the perimeter and area of the rectangle. It should have set and get functions for both length and width. The set functions should verify that length and width are each floating-point numbers larger than 0.0 and less than 20.0

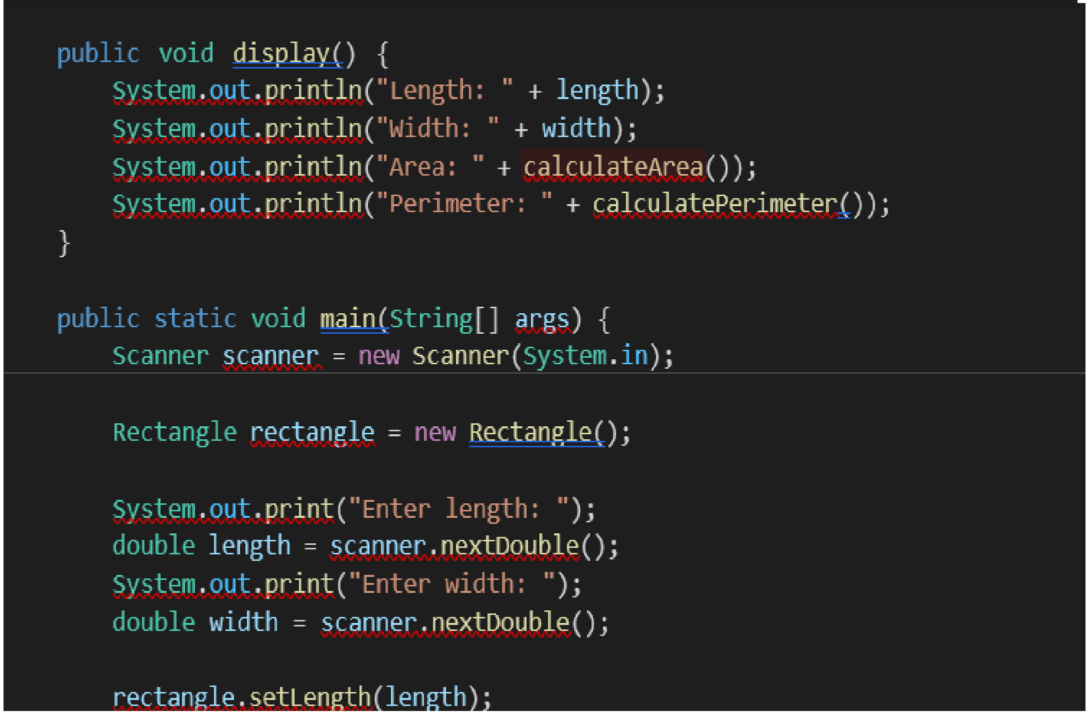
CODE:



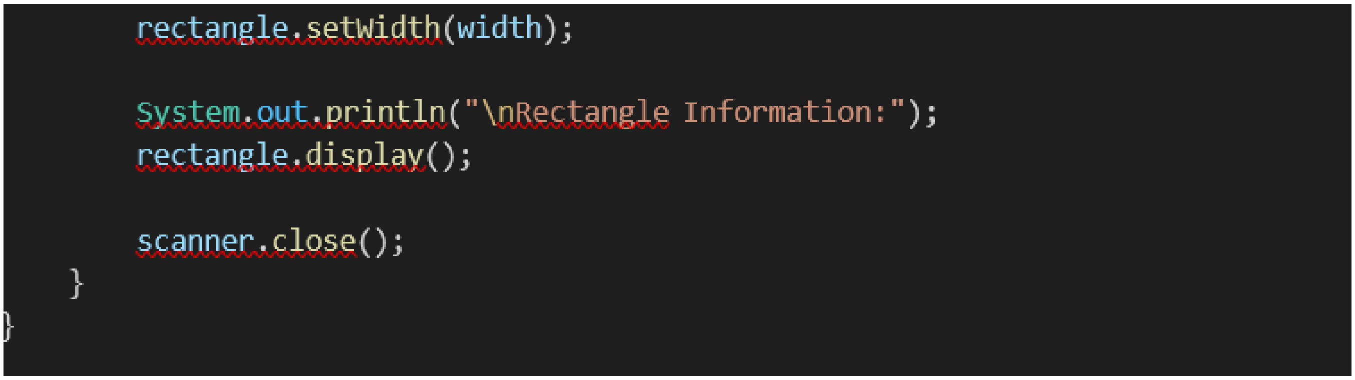
```
import java.util.Scanner;

public class Rectangle {
    private double length;
    private double width;

    public Rectangle() {
        this.length = 1.0;
        this.width = 1.0;
    }

    public Rectangle(double length, double width) {
        setLength(length);
        setWidth(width);
    }

    public double getLength() {
        return length;
    }

    public double getWidth() {
        return width;
    }

    public void setLength(double length) {
        if (length > 0.0 && length < 20.0) {
```



```
            this.length = length;
        } else {
            System.out.println(
                    "Invalid length. Length should be a floating-point number
larger than 0.0 and less than 20.0");
        }
    }

    public void setWidth(double width) {
        if (width > 0.0 && width < 20.0) {
            this.width = width;
        } else {
            System.out.println(
                    "Invalid width. Width should be a floating-point number
larger than 0.0 and less than 20.0");
        }
    }

    public double calculateArea() {
        return length * width;
    }

    public double calculatePerimeter() {
        return 2 * (length + width);
    }
```

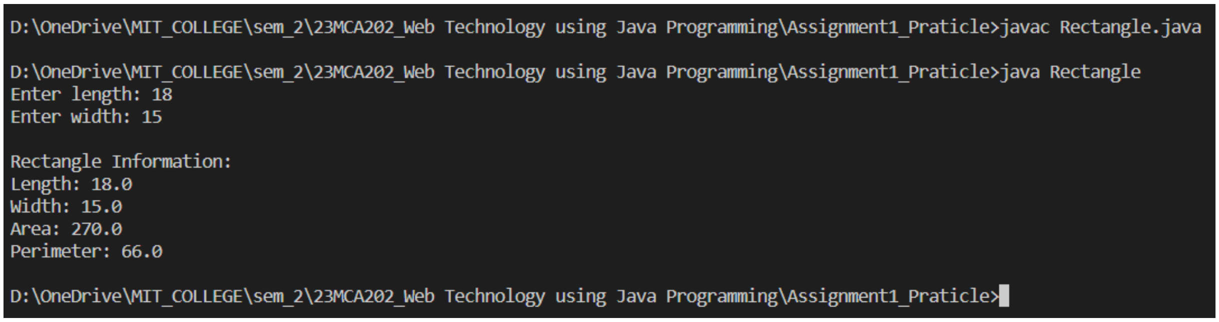


```
    public void display() {
        System.out.println("Length: " + length);
        System.out.println("Width: " + width);
        System.out.println("Area: " + calculateArea());
        System.out.println("Perimeter: " + calculatePerimeter());
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        Rectangle rectangle = new Rectangle();

        System.out.print("Enter length: ");
        double length = scanner.nextDouble();
        System.out.print("Enter width: ");
        double width = scanner.nextDouble();

        rectangle.setLength(length);
```



```
        rectangle.setWidth(width);

        System.out.println("\nRectangle Information:");
        rectangle.display();

        scanner.close();
    }
}
```

OUTPUT:



```
D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment1_Praticle>javac Rectangle.java

D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment1_Praticle>java Rectangle
Enter length: 18
Enter width: 15

Rectangle Information:
Length: 18.0
Width: 15.0
Area: 270.0
Perimeter: 66.0

D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment1_Praticle>
```

Q.5

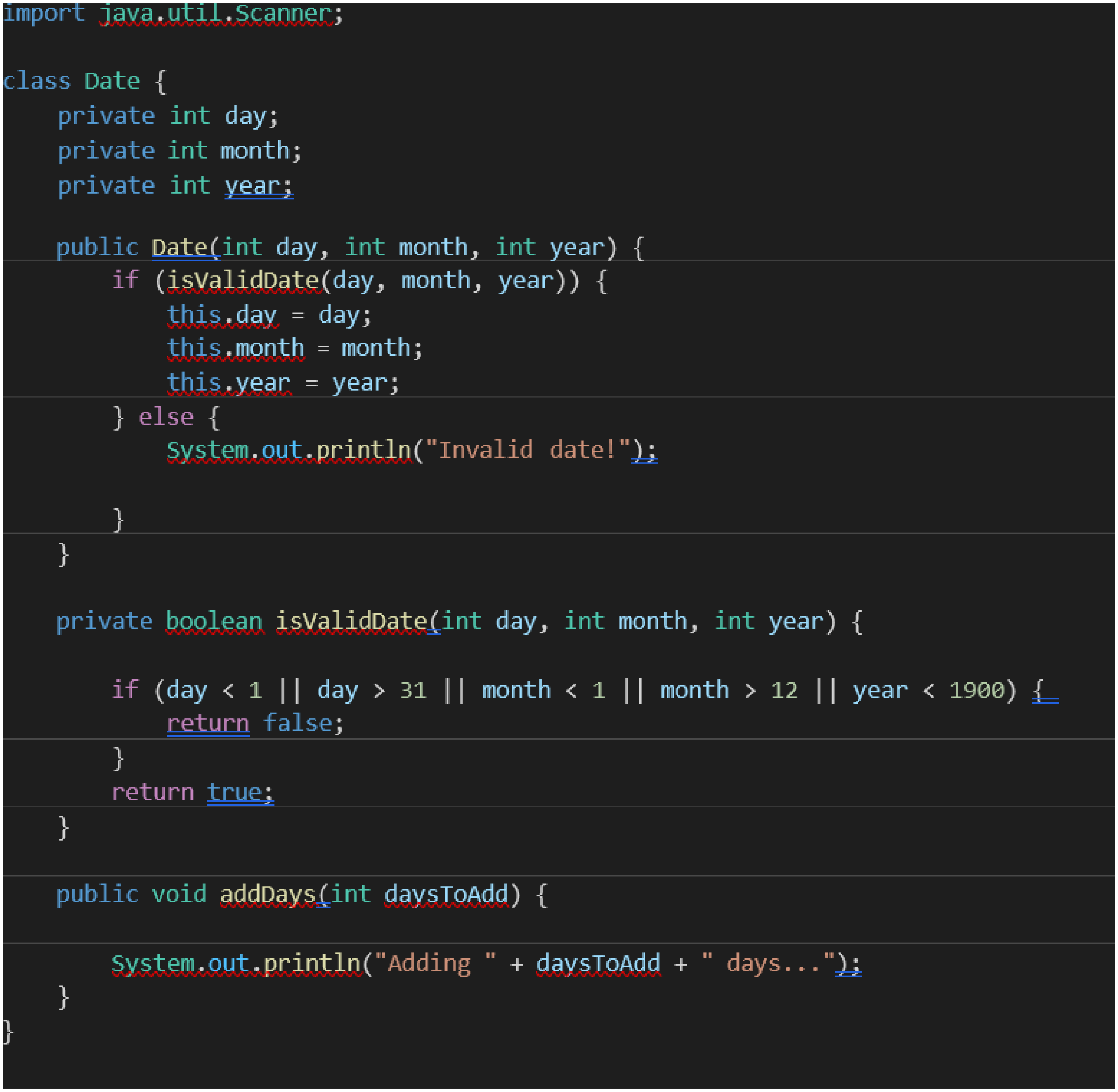
Create a class Date for manipulating dates. Provide a constructor that enables an object of this class to be initialized when it is declared (You can select any default values for the day, month & year, e.g. your birth date). Provide the necessary functionality to perform error checking on the initializer values for data members day, month, and year. Also, provide a member function to add an integer in a date to obtain a new date.

Design separate class Employee which will have following information.

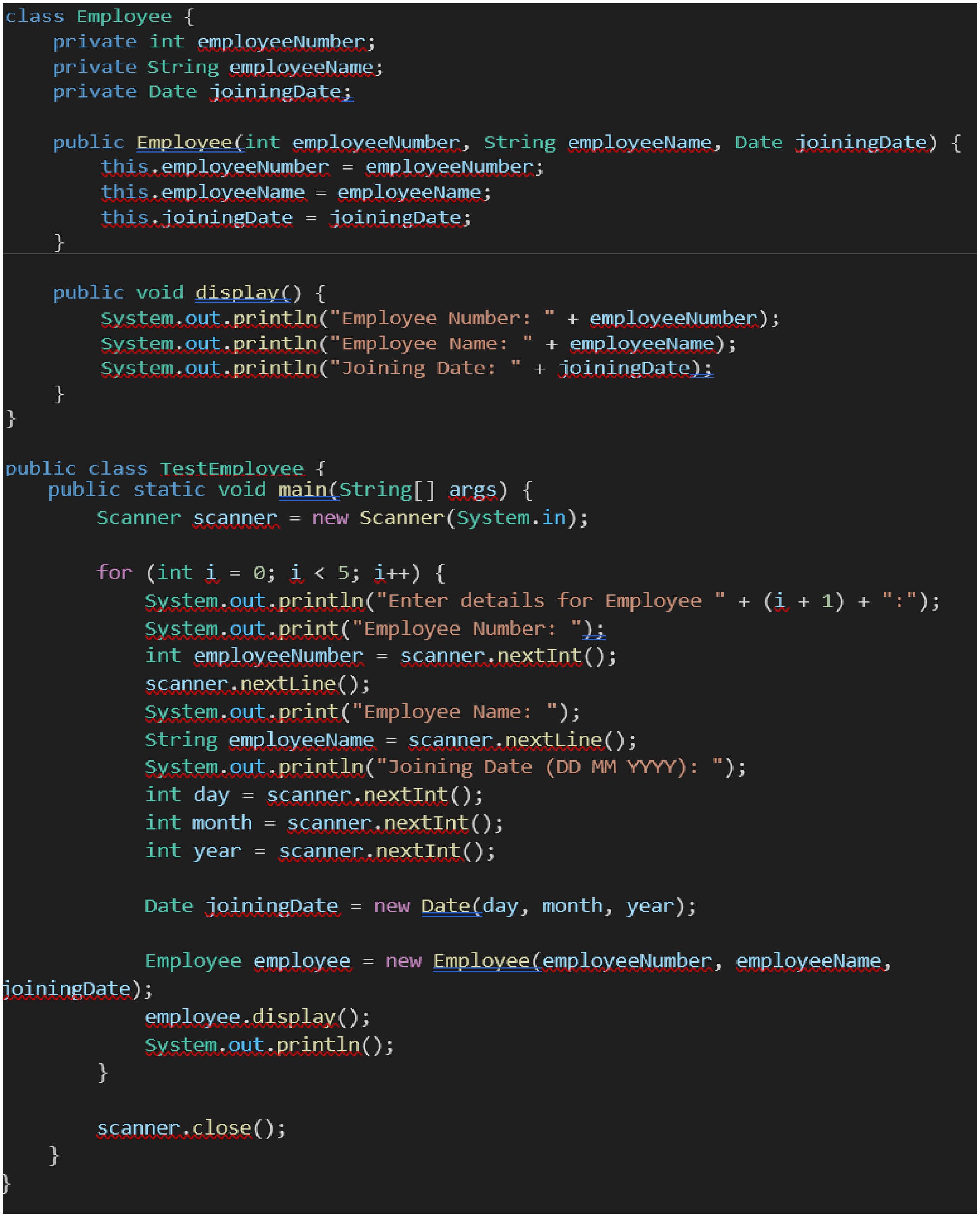
Employee Number	**Number**
Employee Name	**Text**
Joining Date	**Date**

Provide appropriate constructor(s) & methods to this class. Provide main function which will create 5 objects of Employee class.

CODE:

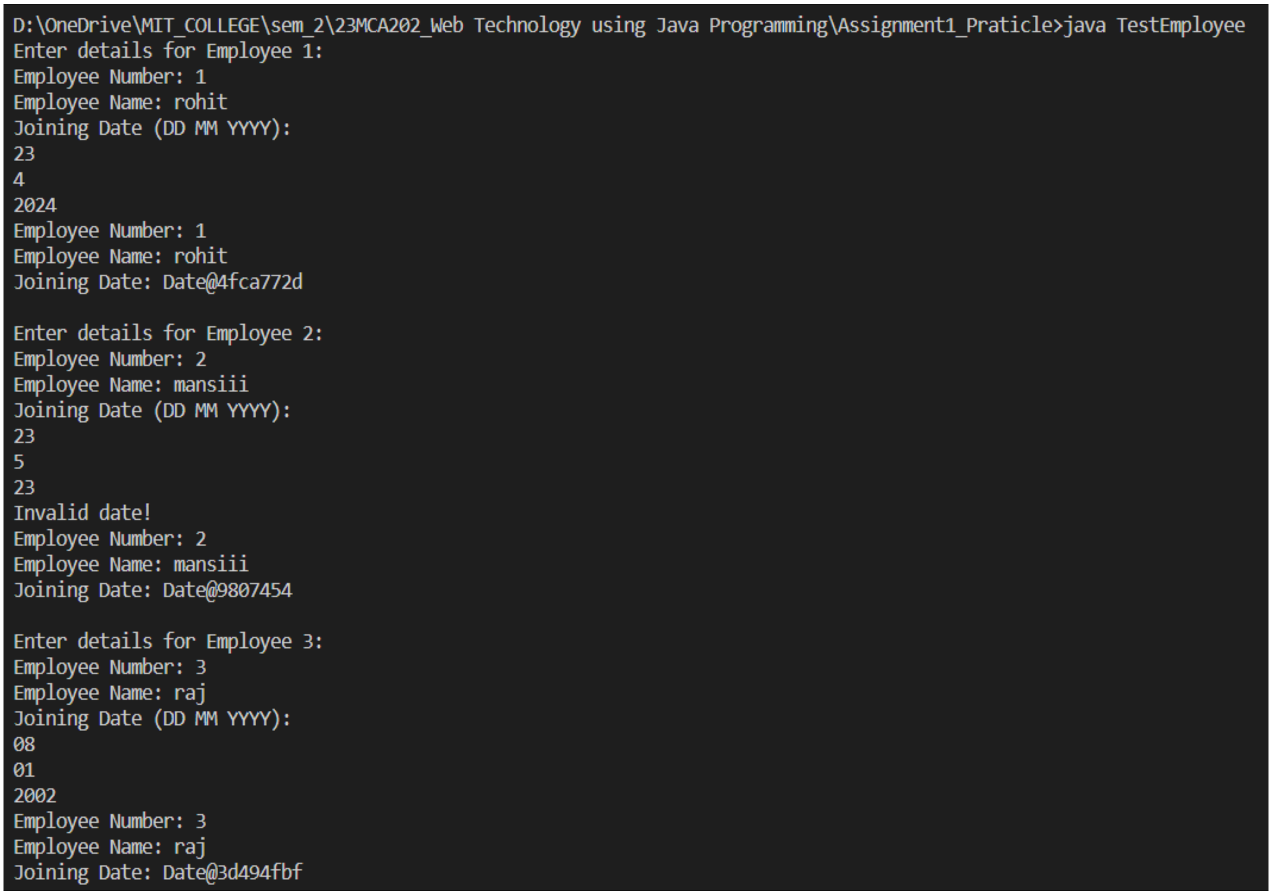


```
import java.util.Scanner;

class Date {
    private int day;
    private int month;
    private int year;

    public Date(int day, int month, int year) {
        if (isValidDate(day, month, year)) {
            this.day = day;
            this.month = month;
            this.year = year;
        } else {
            System.out.println("Invalid date!");

        }
    }

    private boolean isValidDate(int day, int month, int year) {

        if (day < 1 || day > 31 || month < 1 || month > 12 || year < 1900) {
            return false;
        }
        return true;
    }

    public void addDays(int daysToAdd) {

        System.out.println("Adding " + daysToAdd + " days...");
    }
}
```



```
class Employee {
    private int employeeNumber;
    private String employeeName;
    private Date joiningDate;

    public Employee(int employeeNumber, String employeeName, Date joiningDate) {
        this.employeeNumber = employeeNumber;
        this.employeeName = employeeName;
        this.joiningDate = joiningDate;
    }

    public void display() {
        System.out.println("Employee Number: " + employeeNumber);
        System.out.println("Employee Name: " + employeeName);
        System.out.println("Joining Date: " + joiningDate);
    }
}

public class TestEmployee {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        for (int i = 0; i < 5; i++) {
            System.out.println("Enter details for Employee " + (i + 1) + ":");
            System.out.print("Employee Number: ");
            int employeeNumber = scanner.nextInt();
            scanner.nextLine();
            System.out.print("Employee Name: ");
            String employeeName = scanner.nextLine();
            System.out.println("Joining Date (DD MM YYYY): ");
            int day = scanner.nextInt();
            int month = scanner.nextInt();
            int year = scanner.nextInt();

            Date joiningDate = new Date(day, month, year);

            Employee employee = new Employee(employeeNumber, employeeName,
joiningDate);
            employee.display();
            System.out.println();
        }

        scanner.close();
    }
}
```

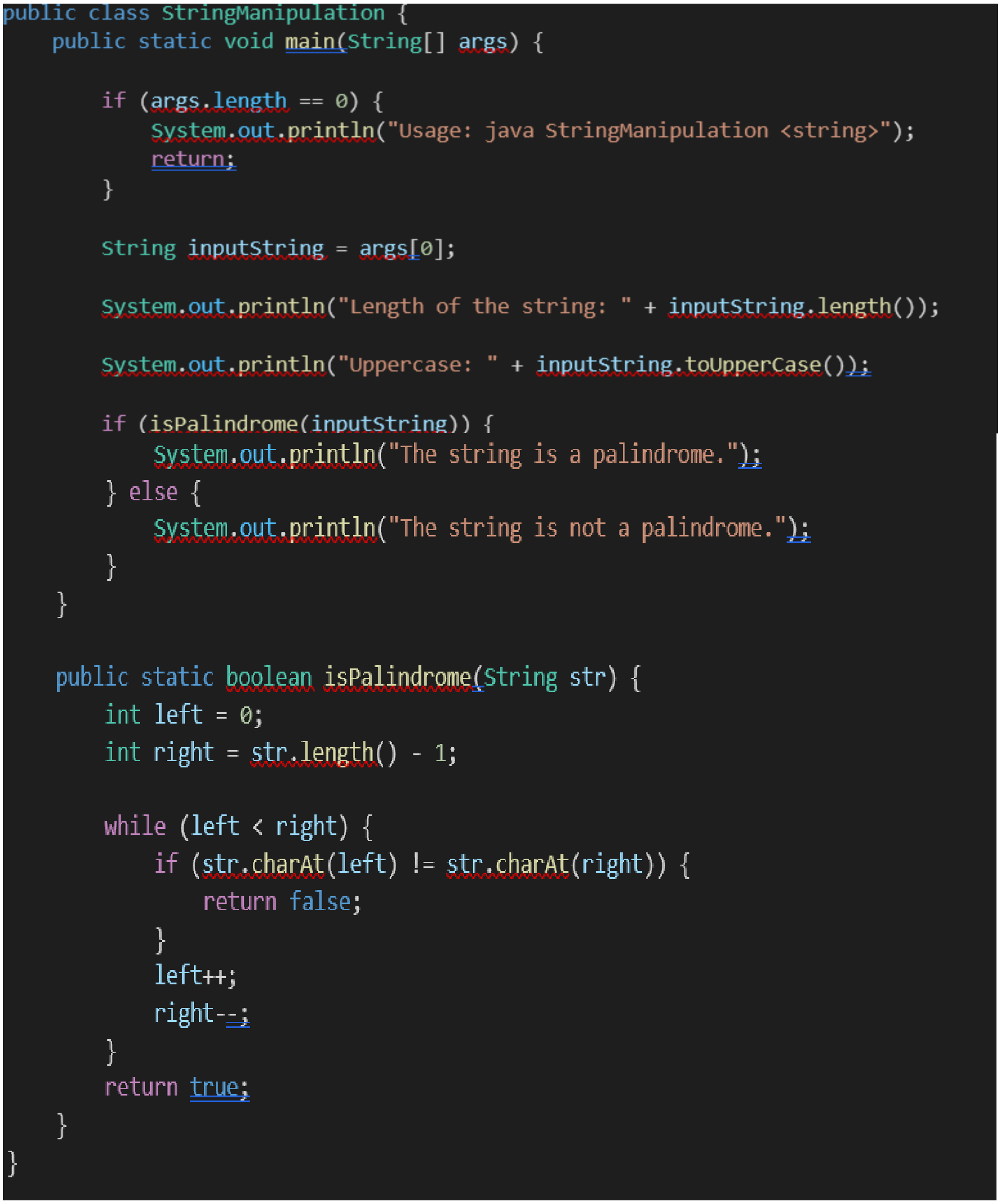
OUTPUT:



```
D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment1_Praticle>java TestEmployee
Enter details for Employee 1:
Employee Number: 1
Employee Name: rohit
Joining Date (DD MM YYYY):
23
4
2024
Employee Number: 1
Employee Name: rohit
Joining Date: Date@4fca772d

Enter details for Employee 2:
Employee Number: 2
Employee Name: mansiii
Joining Date (DD MM YYYY):
23
5
23
Invalid date!
Employee Number: 2
Employee Name: mansiii
Joining Date: Date@9807454

Enter details for Employee 3:
Employee Number: 3
Employee Name: raj
Joining Date (DD MM YYYY):
08
01
2002
Employee Number: 3
Employee Name: raj
Joining Date: Date@3d494fbf
```

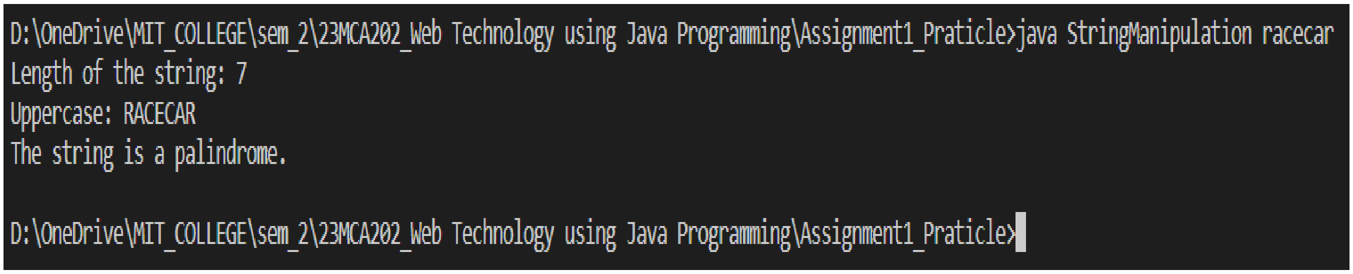
Q.6 Write a program that takes a String through Command Line argument and display the length of the string. Also display the string into uppercase and check whether it is a palindrome or not. (Refer Java API Documentation)

CODE:



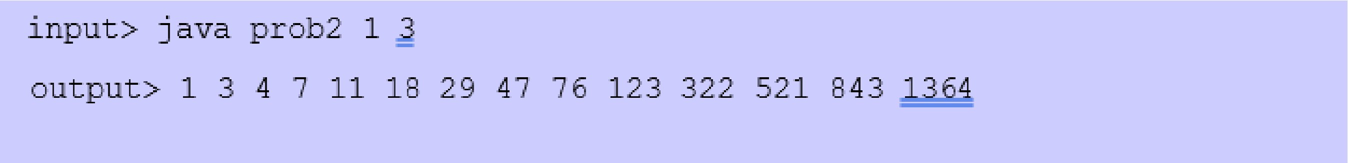
```
public class StringManipulation {
    public static void main(String[] args) {

        if (args.length == 0) {
            System.out.println("Usage: java StringManipulation <string>");
            return;
        }

        String inputString = args[0];

        System.out.println("Length of the string: " + inputString.length());

        System.out.println("Uppercase: " + inputString.toUpperCase());

        if (isPalindrome(inputString)) {
            System.out.println("The string is a palindrome.");
        } else {
            System.out.println("The string is not a palindrome.");
        }
    }

    public static boolean isPalindrome(String str) {
        int left = 0;
        int right = str.length() - 1;

        while (left < right) {
            if (str.charAt(left) != str.charAt(right)) {
                return false;
            }
            left++;
            right--;
        }
        return true;
    }
}
```

OUTPUT:



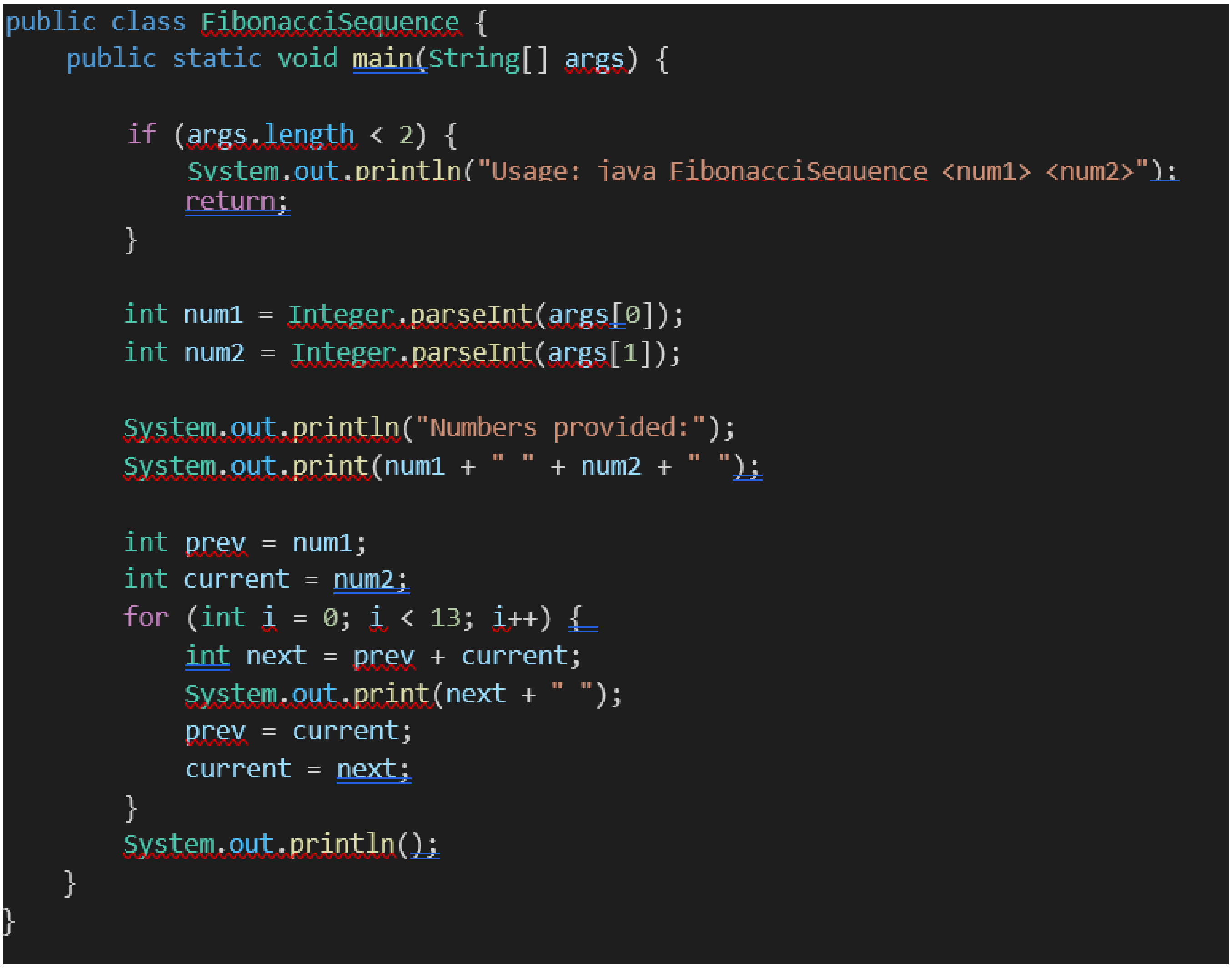
```
D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment1_Praticle>java StringManipulation racecar
Length of the string: 7
Uppercase: RACECAR
The string is a palindrome.

D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment1_Praticle>
```

Q.7 **Write a program that accepts two numbers from the Command Line and prints them out. Then use a *for loop* to print the next 13 numbers in the sequence where each number is the sum of the previous two. For example:**



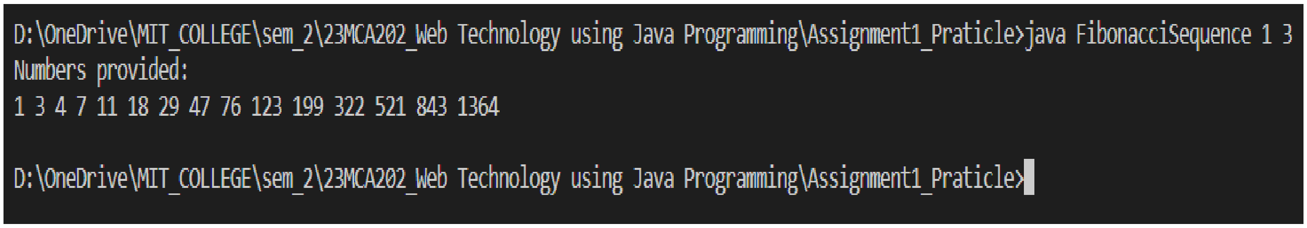
```
input> java prob2 1 3

output> 1 3 4 7 11 18 29 47 76 123 322 521 843 1364
```

CODE:



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

        if (args.length < 2) {
            System.out.println("Usage: java FibonacciSequence <num1> <num2>");
            return;
        }

        int num1 = Integer.parseInt(args[0]);
        int num2 = Integer.parseInt(args[1]);

        System.out.println("Numbers provided:");
        System.out.print(num1 + " " + num2 + " ");

        int prev = num1;
        int current = num2;
        for (int i = 0; i < 13; i++) {
            int next = prev + current;
            System.out.print(next + " ");
            prev = current;
            current = next;
        }
        System.out.println();
    }
}
```

OUTPUT:



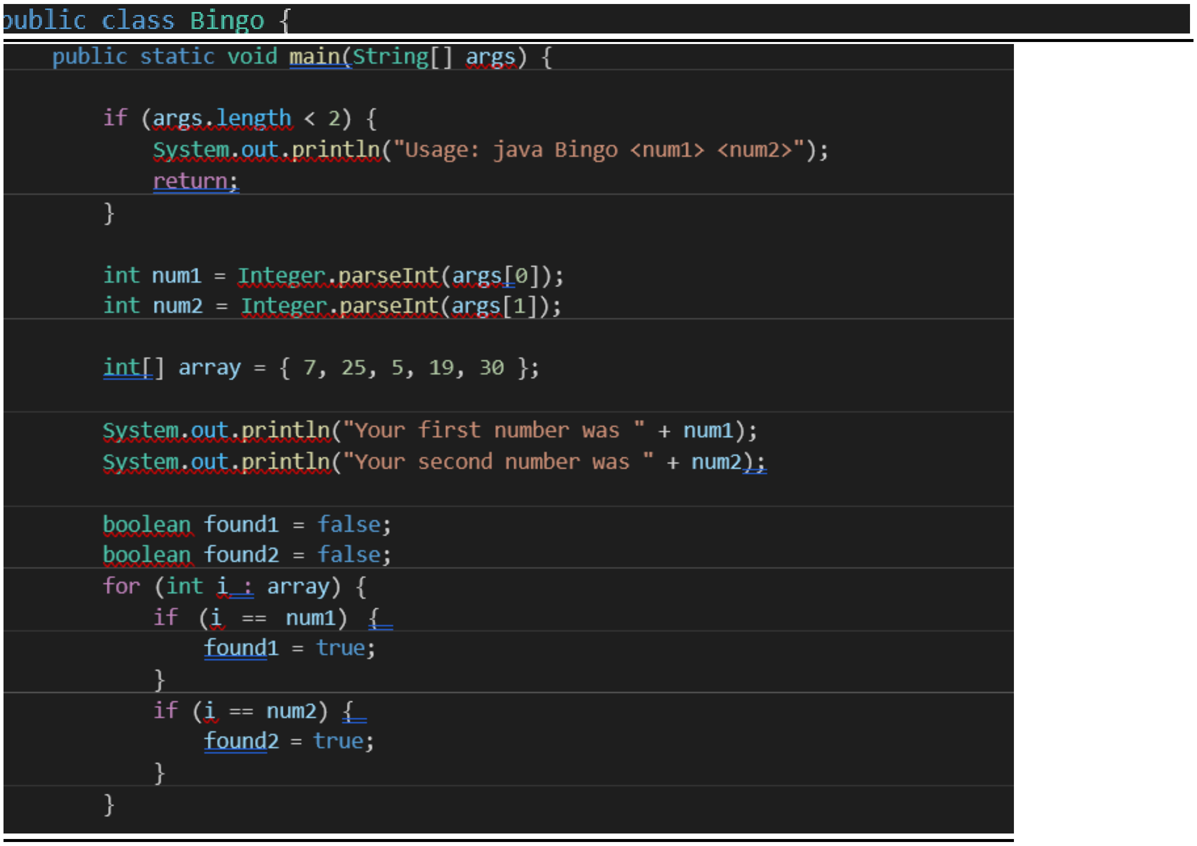
```
D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment1_Praticle>java FibonacciSequence 1 3
Numbers provided:
1 3 4 7 11 18 29 47 76 123 199 322 521 843 1364

D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment1_Praticle>
```

Q.8 Write a program that accepts two numbers in the range from 1 to 40 from the

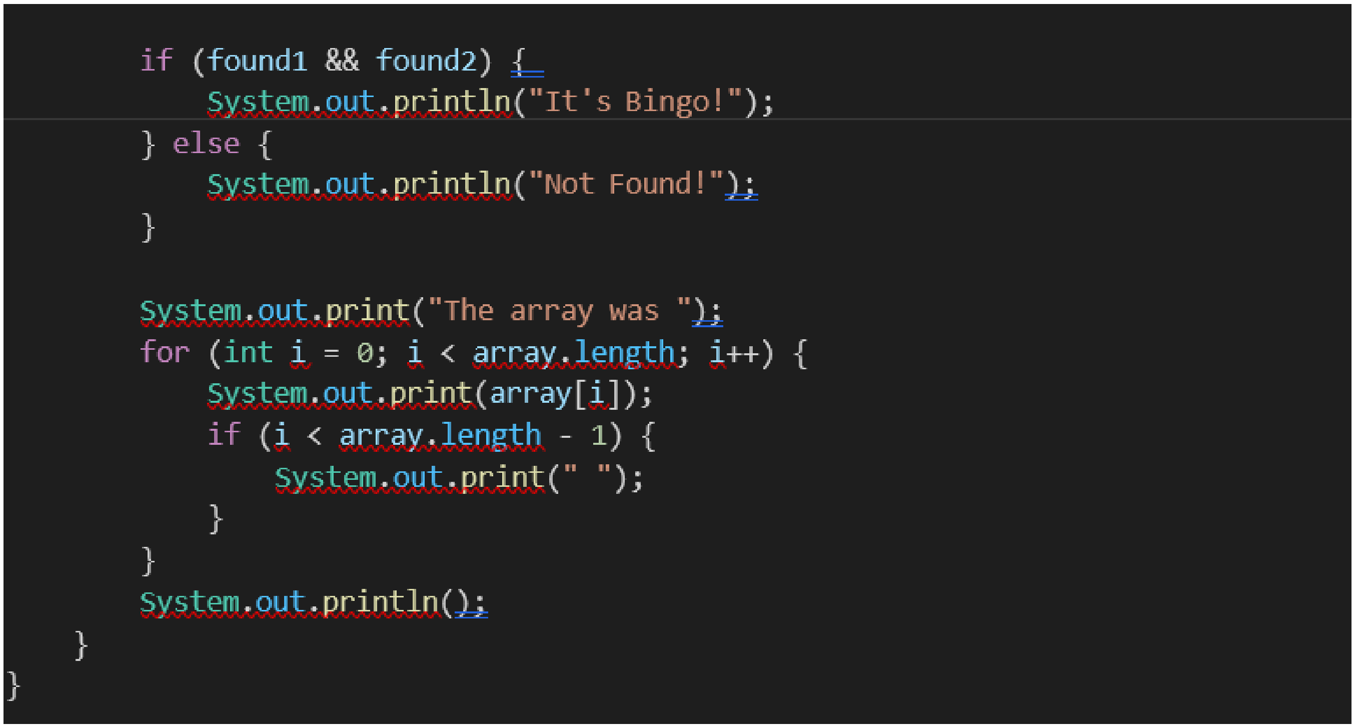
Command Line. It then compares these numbers against a single dimension *array* of five integer elements ranging in value from 1 to 40. The program displays the message *BINGO* if the two inputted values are found in the array element. For example:

```
 input>java prob3 3 29

 output>Your first number was 3

 Your second number was 29

Its Bingo! // this message if 3 and 29 is found in the array

Not Found! // this message if 3 and 29 is not found in the
 //array

The array was 7 25 5 19 30
```

CODE:

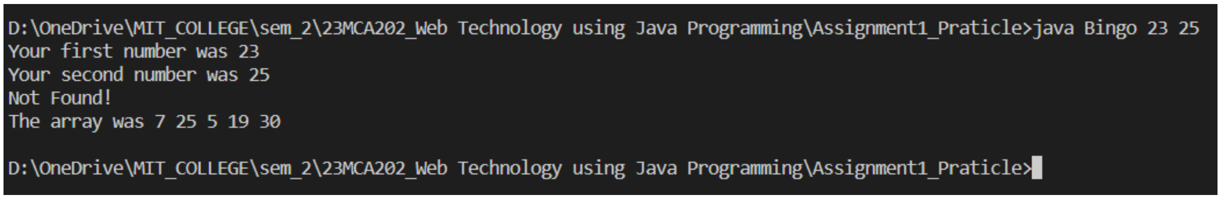


```
public class Bingo {
    public static void main(String[] args) {

        if (args.length < 2) {
            System.out.println("Usage: java Bingo <num1> <num2>");
            return;
        }

        int num1 = Integer.parseInt(args[0]);
        int num2 = Integer.parseInt(args[1]);

        int[] array = { 7, 25, 5, 19, 30 };

        System.out.println("Your first number was " + num1);
        System.out.println("Your second number was " + num2);

        boolean found1 = false;
        boolean found2 = false;
        for (int i : array) {
            if (i == num1) {
                found1 = true;
            }
            if (i == num2) {
                found2 = true;
            }
        }
```



```java
        if (found1 && found2) {
            System.out.println("It's Bingo!");
        } else {
            System.out.println("Not Found!");
        }

        System.out.print("The array was ");
        for (int i = 0; i < array.length; i++) {
            System.out.print(array[i]);
            if (i < array.length - 1) {
                System.out.print(" ");
            }
        }
        System.out.println();
    }
}
```

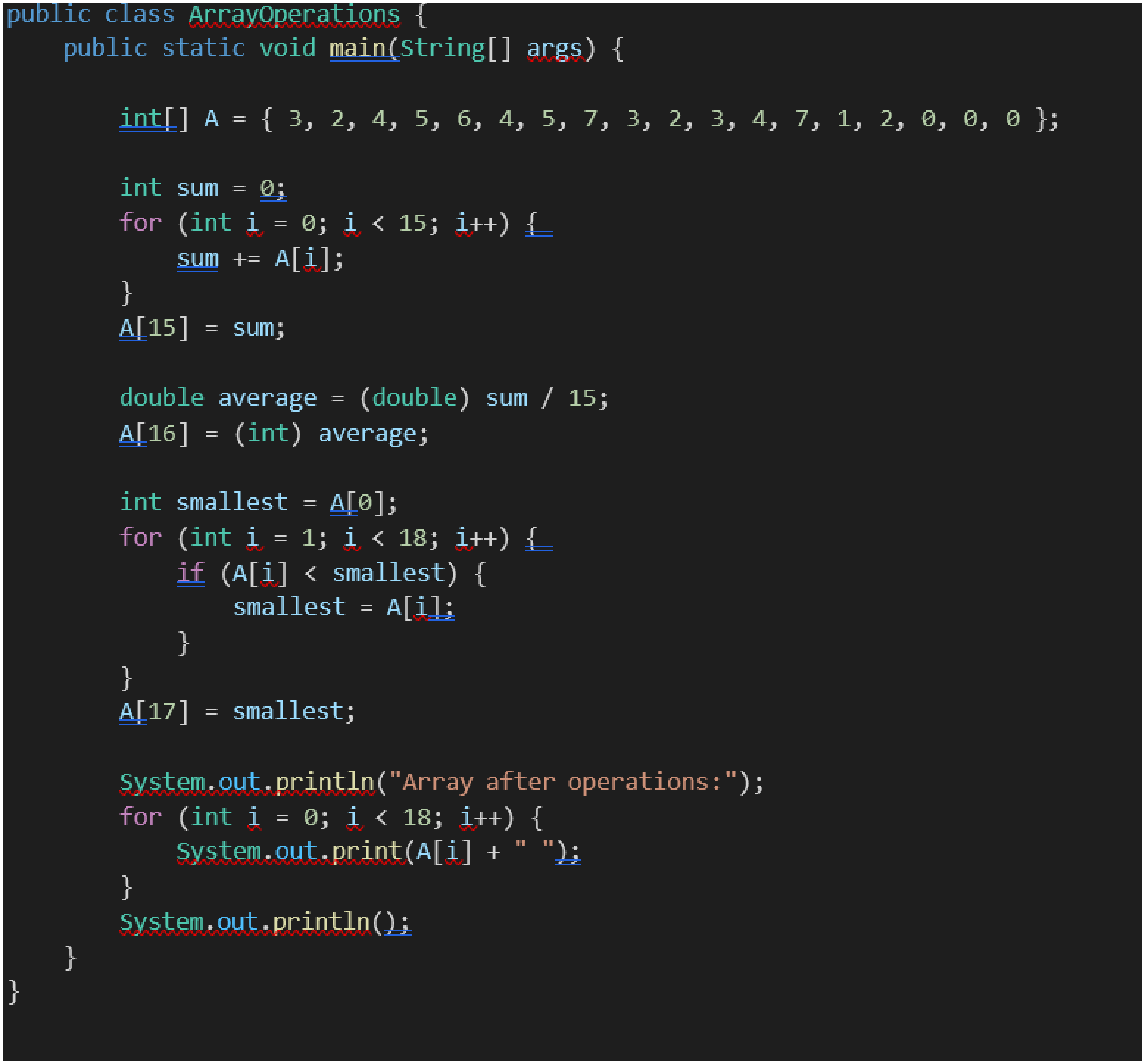
OUTPUT:



```
D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment1_Praticle>java Bingo 23 25
Your first number was 23
Your second number was 25
Not Found!
The array was 7 25 5 19 30

D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment1_Praticle>
```

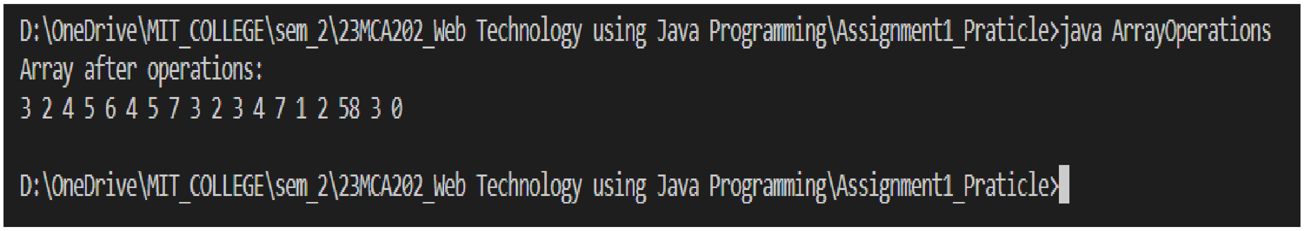
Q.9 **Write a program that allows you to create an integer *array* of 18 elements with the following values: *int A[]={3,2,4,5,6,4,5,7,3,2,3,4,7,1,2,0,0,0}*. The program computes the sum of element 0 to 14 and stores it at element 15, computes the average and stores it at element 16 and identifies the smallest value from the array and stores it at element 17.**

CODE:



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

        int[] A = { 3, 2, 4, 5, 6, 4, 5, 7, 3, 2, 3, 4, 7, 1, 2, 0, 0, 0 };

        int sum = 0;
        for (int i = 0; i < 15; i++) {
            sum += A[i];
        }
        A[15] = sum;

        double average = (double) sum / 15;
        A[16] = (int) average;

        int smallest = A[0];
        for (int i = 1; i < 18; i++) {
            if (A[i] < smallest) {
                smallest = A[i];
            }
        }
        A[17] = smallest;

        System.out.println("Array after operations:");
        for (int i = 0; i < 18; i++) {
            System.out.print(A[i] + " ");
        }
        System.out.println();
    }
}
```

OUTPUT:



```
D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment1_Praticle>java ArrayOperations
Array after operations:
3 2 4 5 6 4 5 7 3 2 3 4 7 1 2 58 3 0

D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment1_Praticle>
```

Q.10 Create a class Term. This class represents a term of a polynomial such as $2x^4$ where 2 is coefficient and 4 is exponent of the term.

<u>Data members</u>:
- int coefficient
int exponent

Create another class Polynomial. The internal representation of a polynomial is an array of Terms. The size of this array should be fixed.

Provide a constructor for this class that will set all terms of a polynomial object as zero (where coefficient is 0 and exponent is 0).

Provide following functions:

setTerm(int, int) – Setting a term of a polynomial object. Each successive call of this function should set next term of the polynomial object.

It should do the following validations:-

Whether the exponent of the term being set is already used. Whether the array size limit is exceeded.

Whether the exponent is negative.

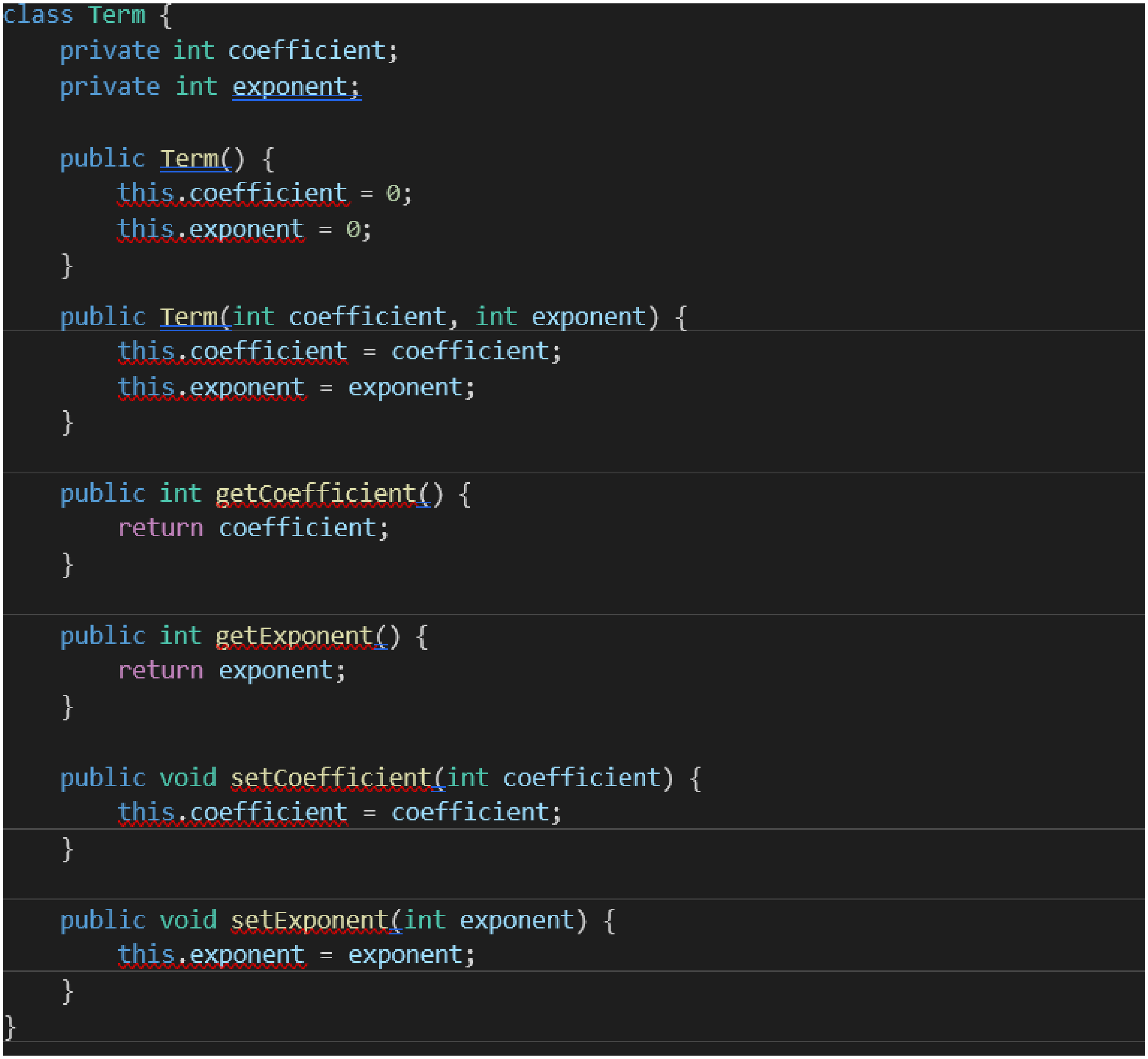
In all the cases it should not set the term and display an appropriate message.

sort() – to arrange the terms in ascending orderof

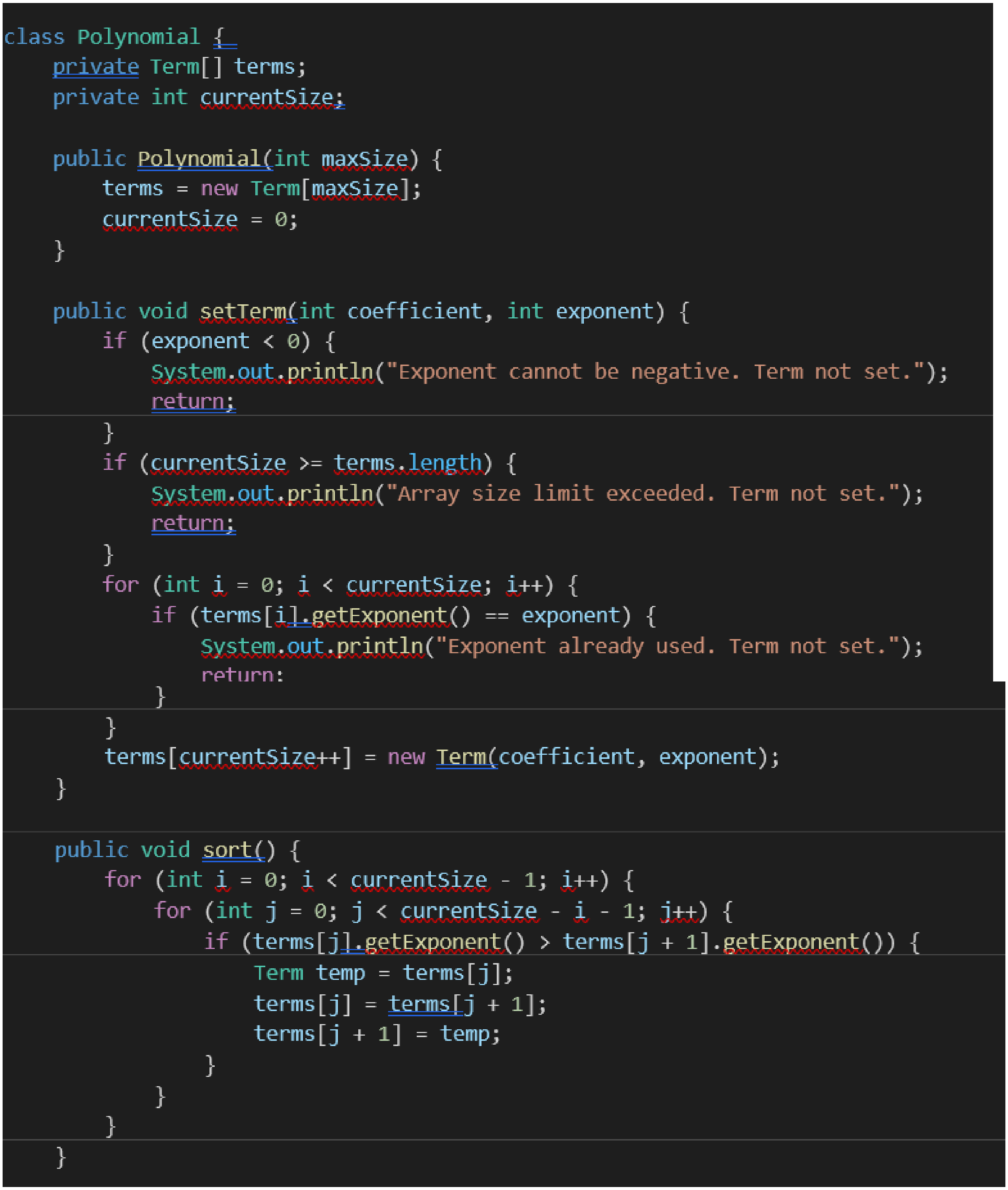
exponents. Provide a function to print a polynomial

object.

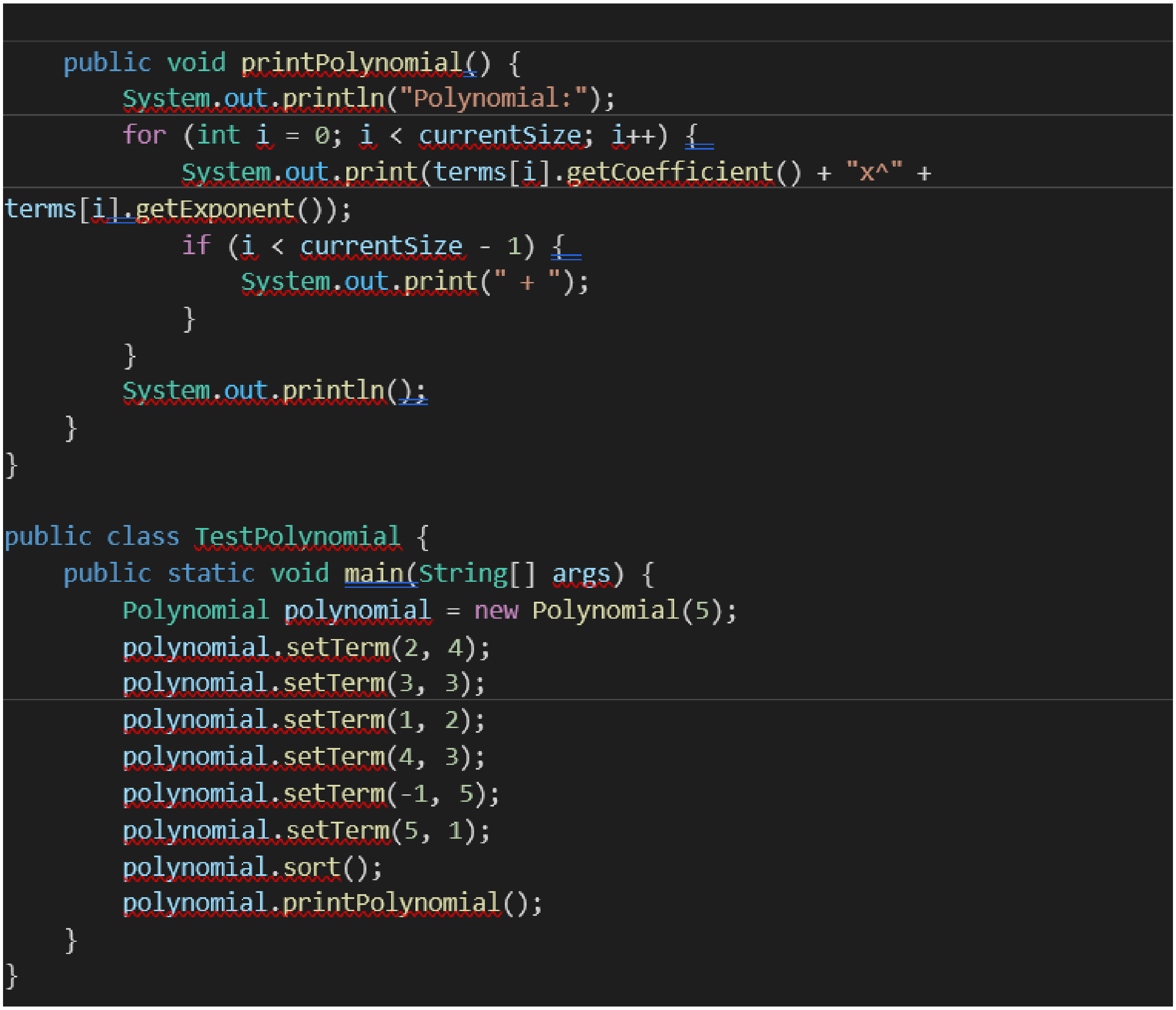
CODE:



```java
class Term {
    private int coefficient;
    private int exponent;

    public Term() {
        this.coefficient = 0;
        this.exponent = 0;
    }
    public Term(int coefficient, int exponent) {
        this.coefficient = coefficient;
        this.exponent = exponent;
    }

    public int getCoefficient() {
        return coefficient;
    }

    public int getExponent() {
        return exponent;
    }

    public void setCoefficient(int coefficient) {
        this.coefficient = coefficient;
    }

    public void setExponent(int exponent) {
        this.exponent = exponent;
    }
}
```

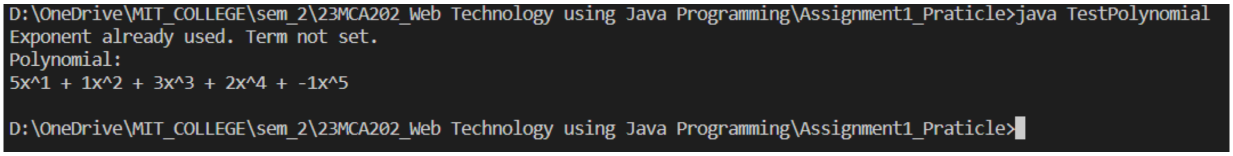


```
class Polynomial {
    private Term[] terms;
    private int currentSize;

    public Polynomial(int maxSize) {
        terms = new Term[maxSize];
        currentSize = 0;
    }

    public void setTerm(int coefficient, int exponent) {
        if (exponent < 0) {
            System.out.println("Exponent cannot be negative. Term not set.");
            return;
        }
        if (currentSize >= terms.length) {
            System.out.println("Array size limit exceeded. Term not set.");
            return;
        }
        for (int i = 0; i < currentSize; i++) {
            if (terms[i].getExponent() == exponent) {
                System.out.println("Exponent already used. Term not set.");
                return;
            }
        }
        terms[currentSize++] = new Term(coefficient, exponent);
    }

    public void sort() {
        for (int i = 0; i < currentSize - 1; i++) {
            for (int j = 0; j < currentSize - i - 1; j++) {
                if (terms[j].getExponent() > terms[j + 1].getExponent()) {
                    Term temp = terms[j];
                    terms[j] = terms[j + 1];
                    terms[j + 1] = temp;
                }
            }
        }
    }
```



```
    public void printPolynomial() {
        System.out.println("Polynomial:");
        for (int i = 0; i < currentSize; i++) {
            System.out.print(terms[i].getCoefficient() + "x^" +
terms[i].getExponent());
            if (i < currentSize - 1) {
                System.out.print(" + ");
            }
        }
        System.out.println();
    }
}

public class TestPolynomial {
    public static void main(String[] args) {
        Polynomial polynomial = new Polynomial(5);
        polynomial.setTerm(2, 4);
        polynomial.setTerm(3, 3);
        polynomial.setTerm(1, 2);
        polynomial.setTerm(4, 3);
        polynomial.setTerm(-1, 5);
        polynomial.setTerm(5, 1);
        polynomial.sort();
        polynomial.printPolynomial();
    }
}
```

OUTPUT:



```
D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment1_Praticle>java TestPolynomial
Exponent already used. Term not set.
Polynomial:
5x^1 + 1x^2 + 3x^3 + 2x^4 + -1x^5

D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment1_Praticle>
```

Passing objects to methods

Q.11 Create a class, Matrix. Internal representation of this class will be a two dimensional array of size 10 by 10. In addition, the class should have following data members and member functions:

Data members: -

int rows

int columns

Constructors -

The default constructor

Matrix () - This should set each of the array element to zero.

Overloaded constructor

Matrix (int, int) - This constructor should call the default constructor first. It should then assign the value of first parameter to variable rows, and the value of the second parameter to variable columns. You can assume that the values of both the parameters will be less than or equal to 10.

Member functions -

void set Element(int r, int c, int value) - This function should set the array element at row r and column c to the value val. This assignment should be done only if val is positive r and c are valid else the element should be set to zero.

Matrix transpose () – This function should transpose the matrix. Transpose of a matrix is another matrix where the elements in rows of the first

matrix become elements of the corresponding columns in the new matrix.

Provide a function to print a Matrix object.

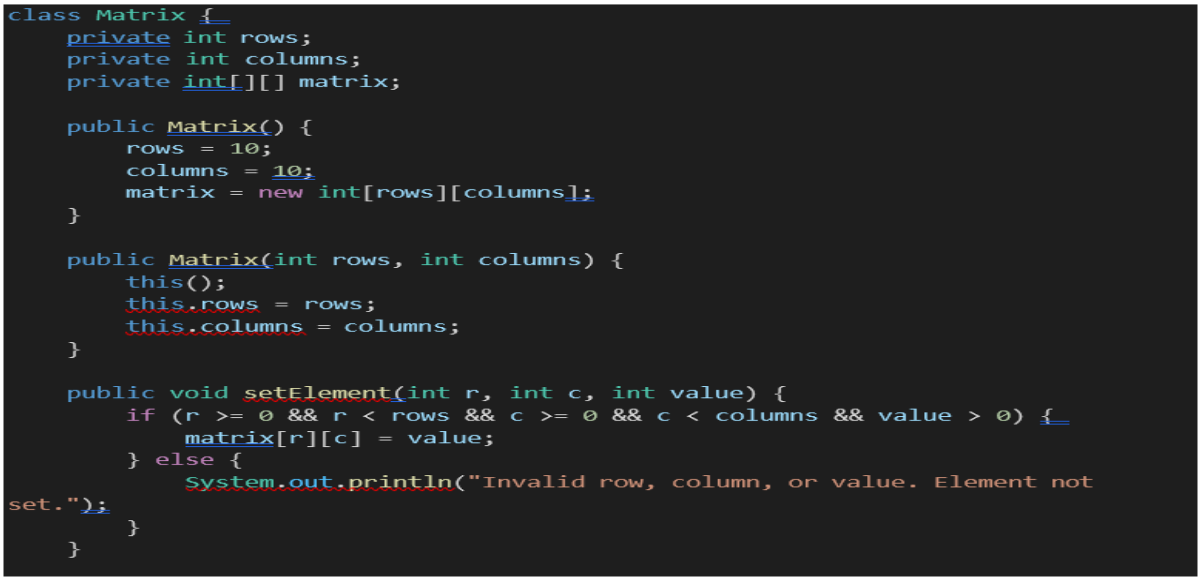
Q.12 Create a class called complex for performing arithmetic operations with complex numbers. Use floating point variables to represent the private data of the class. Provide a default constructor that initializes the object with some default values. Provide public member functions for each of the following.

- **Addition of two complex numbers**: It returns the result obtained by adding the respective real parts and the imaginary parts of the two complex numbers.
- **Subtraction of two complex numbers**: It returns the result obtained by subtracting the respective real parts and the imaginary parts of the two complex numbers.
- **display()** – It displays the complex number in **a+bi**

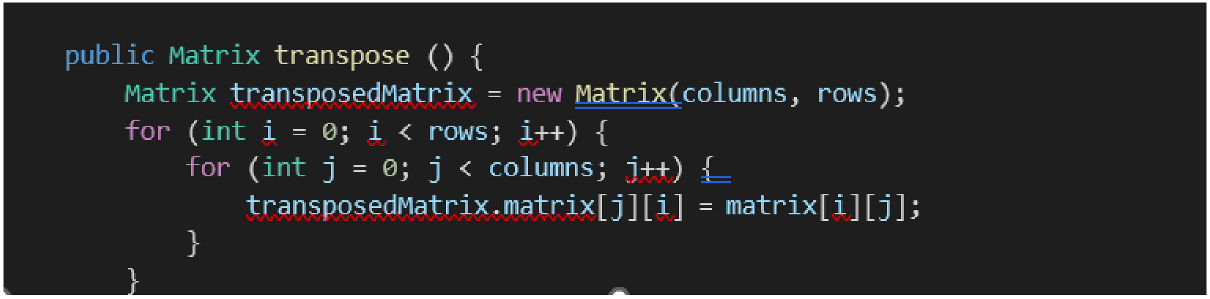
format. The output should be displayed as follows:-

Sum of a1+b1 **i &** a2+b2 **i** is **:** a3+b3 **i**

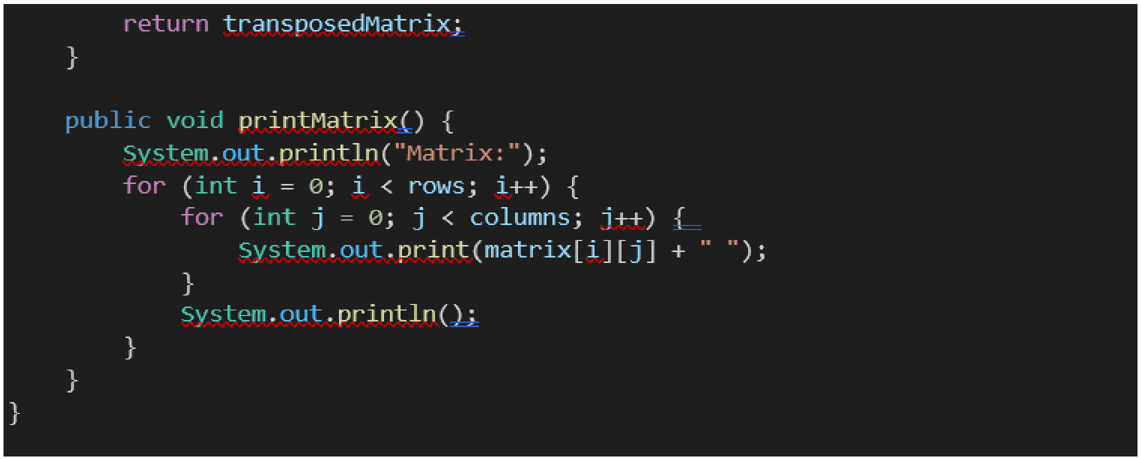
CODE:



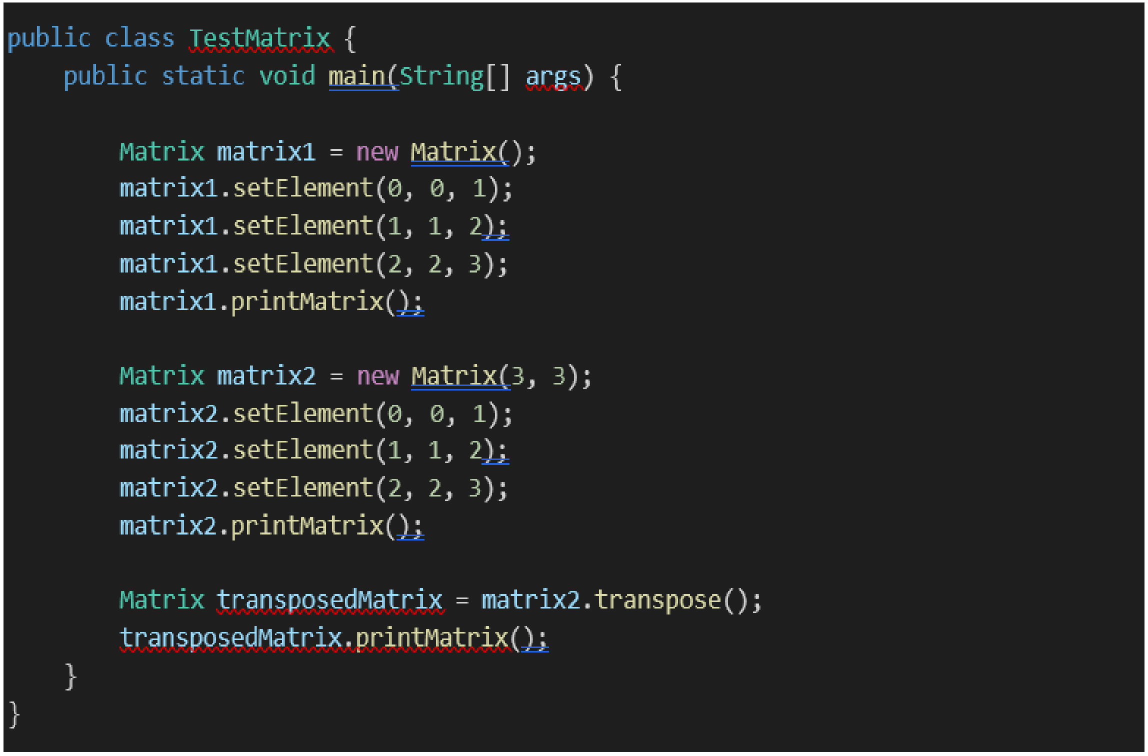
```
class Matrix {
    private int rows;
    private int columns;
    private int[][] matrix;

    public Matrix() {
        rows = 10;
        columns = 10;
        matrix = new int[rows][columns];
    }

    public Matrix(int rows, int columns) {
        this();
        this.rows = rows;
        this.columns = columns;
    }

    public void setElement(int r, int c, int value) {
        if (r >= 0 && r < rows && c >= 0 && c < columns && value > 0) {
            matrix[r][c] = value;
        } else {
            System.out.println("Invalid row, column, or value. Element not
set.");
        }
    }
```



```
    public Matrix transpose () {
        Matrix transposedMatrix = new Matrix(columns, rows);
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < columns; j++) {
                transposedMatrix.matrix[j][i] = matrix[i][j];
            }
        }
```

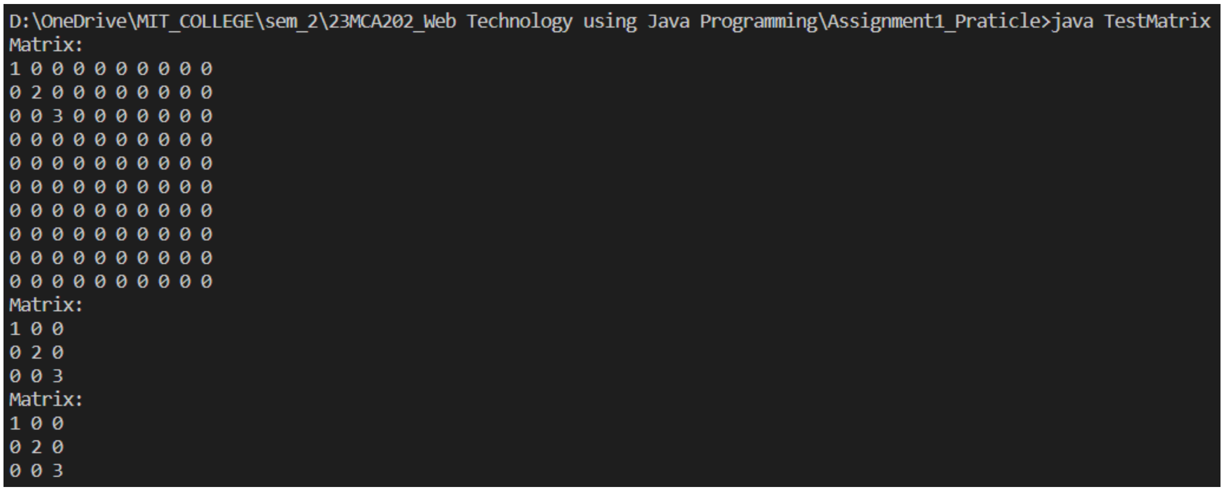


```
        return transposedMatrix;
    }

    public void printMatrix() {
        System.out.println("Matrix:");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < columns; j++) {
                System.out.print(matrix[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```



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

        Matrix matrix1 = new Matrix();
        matrix1.setElement(0, 0, 1);
        matrix1.setElement(1, 1, 2);
        matrix1.setElement(2, 2, 3);
        matrix1.printMatrix();

        Matrix matrix2 = new Matrix(3, 3);
        matrix2.setElement(0, 0, 1);
        matrix2.setElement(1, 1, 2);
        matrix2.setElement(2, 2, 3);
        matrix2.printMatrix();

        Matrix transposedMatrix = matrix2.transpose();
        transposedMatrix.printMatrix();
    }
}
```

OUTPUT:



```
D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment1_Praticle>java TestMatrix
Matrix:
1 0 0 0 0 0 0 0 0 0
0 2 0 0 0 0 0 0 0 0
0 0 3 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
Matrix:
1 0 0
0 2 0
0 0 3
Matrix:
1 0 0
0 2 0
0 0 3
```

ASSIGNMENTS NO. 2

Objective: At the end of the assignments, participants will be able to create abstract classes, create a new class by extending an existing class, write code to exhibit polymorphic behaviour of a method call, use of interfaces

Concept: Inheritance and Polymorphism.

Q.1 Create an abstract class Instrument which is having the abstract function play. Create three more sub classes from Instrument which is Piano, Flute, Guitar.
Override the play method inside all three classes printing a message

"Piano is playing tan tan tan tan " for Piano class
"Flute is playing toot toot toot toot" for Flute class "Guitar is playing tin tin tin " for Guitar class

You must not allow the user to declare an object of

Instrument class. Create an array of 10 Instruments.

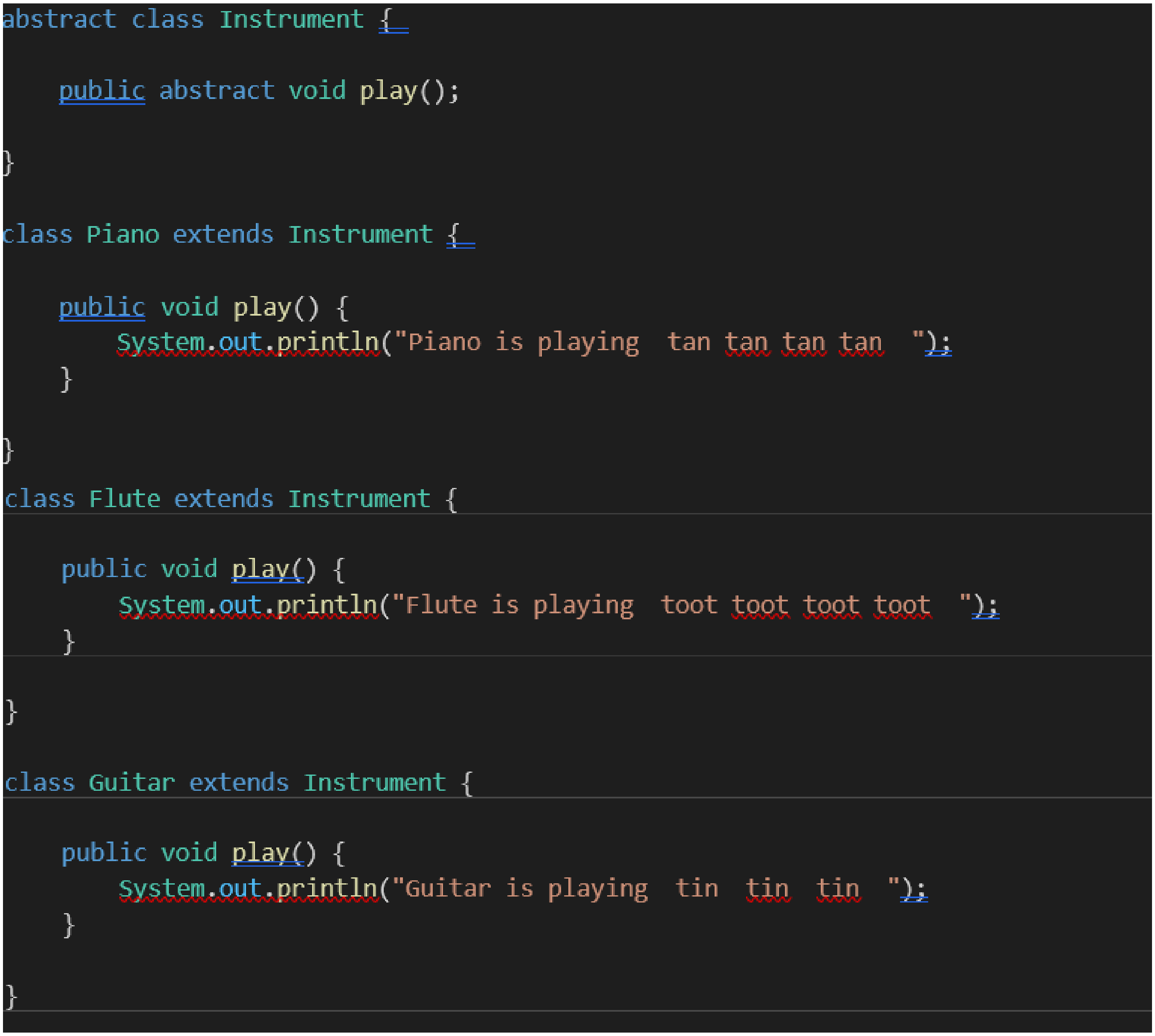
Assign different type of instrument to Instrument

reference. Check for the polymorphic behavior of

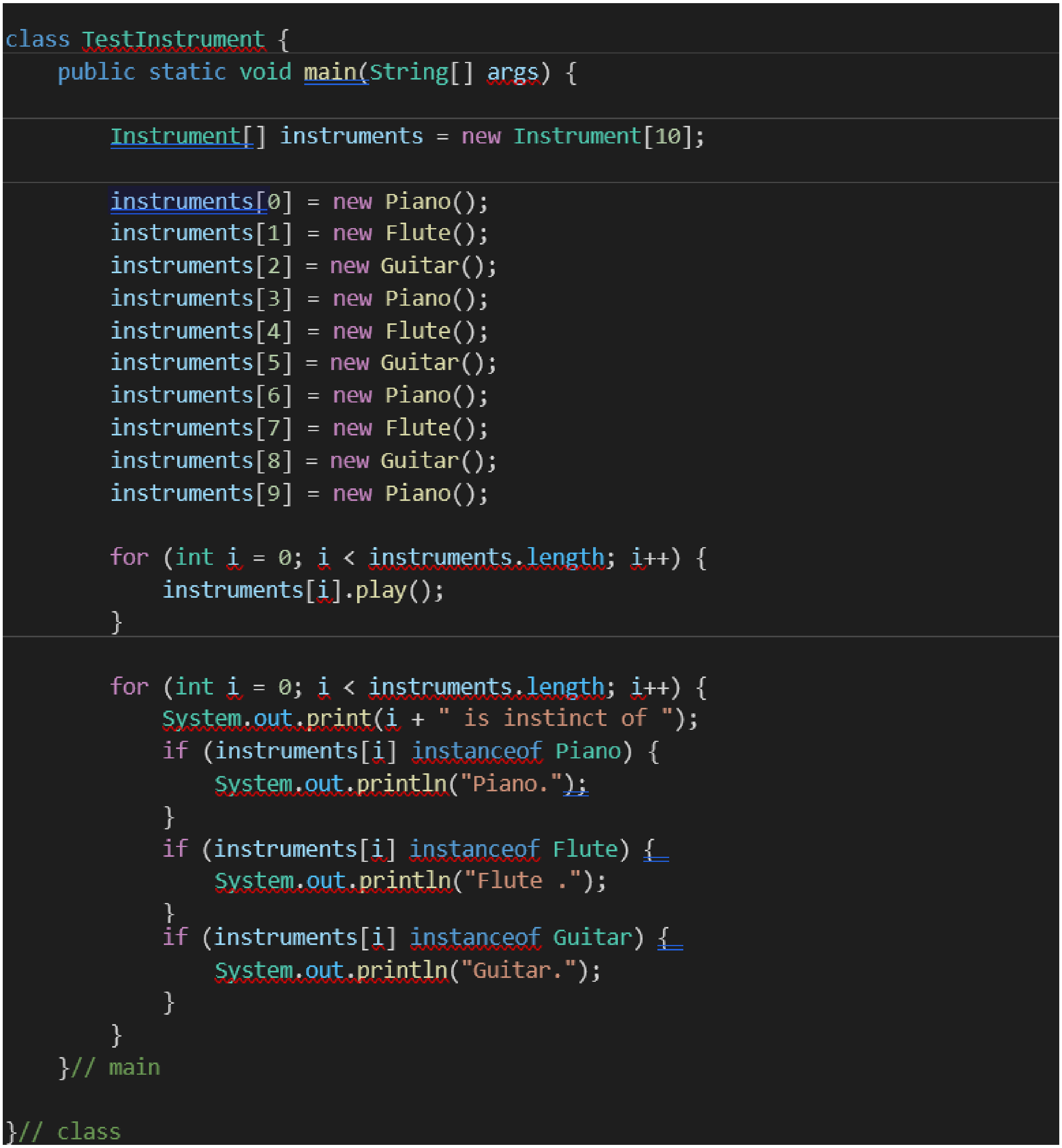
play method.

Use the instanceof operator to print that which object stored at which index of instrument array

CODE:



```
abstract class Instrument {

    public abstract void play();

}

class Piano extends Instrument {

    public void play() {
        System.out.println("Piano is playing  tan tan tan tan  ");
    }

}
class Flute extends Instrument {

    public void play() {
        System.out.println("Flute is playing  toot toot toot toot  ");
    }

}

class Guitar extends Instrument {

    public void play() {
        System.out.println("Guitar is playing  tin  tin  tin  ");
    }

}
```



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

        Instrument[] instruments = new Instrument[10];

        instruments[0] = new Piano();
        instruments[1] = new Flute();
        instruments[2] = new Guitar();
        instruments[3] = new Piano();
        instruments[4] = new Flute();
        instruments[5] = new Guitar();
        instruments[6] = new Piano();
        instruments[7] = new Flute();
        instruments[8] = new Guitar();
        instruments[9] = new Piano();

        for (int i = 0; i < instruments.length; i++) {
            instruments[i].play();
        }

        for (int i = 0; i < instruments.length; i++) {
            System.out.print(i + " is instinct of ");
            if (instruments[i] instanceof Piano) {
                System.out.println("Piano.");
            }
            if (instruments[i] instanceof Flute) {
                System.out.println("Flute .");
            }
            if (instruments[i] instanceof Guitar) {
                System.out.println("Guitar.");
            }
        }
    }// main

}// class
```

Q.2 Create an abstract class Compartment to represent a rail coach. Provide an

abstract function notice in this class. Derive FirstClass, Ladies, General, Luggage classes from the compartment class. Override the notice function in each of them to print notice suitable to the type of the compartment.

Create a class Test Compartment. Write main function to do the

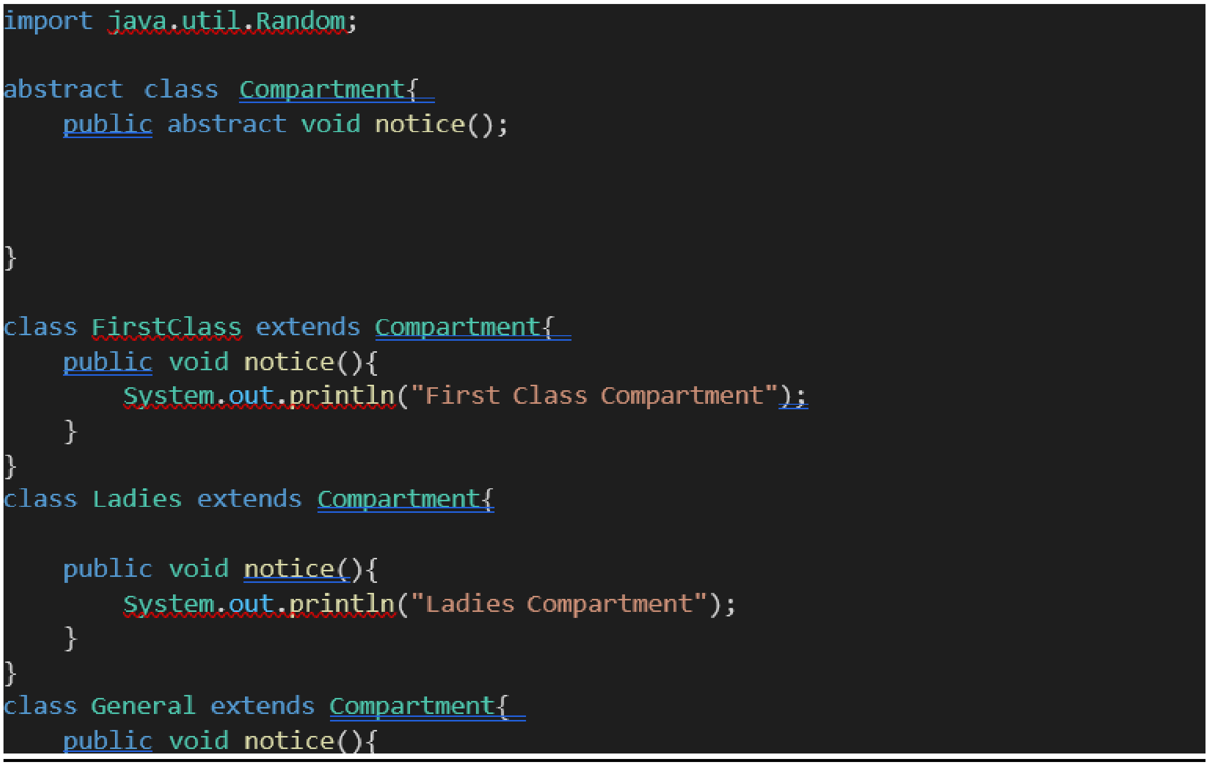
following: Declare an array of Compartment pointers of size 10.

Create a compartment of a type as decided by a randomly

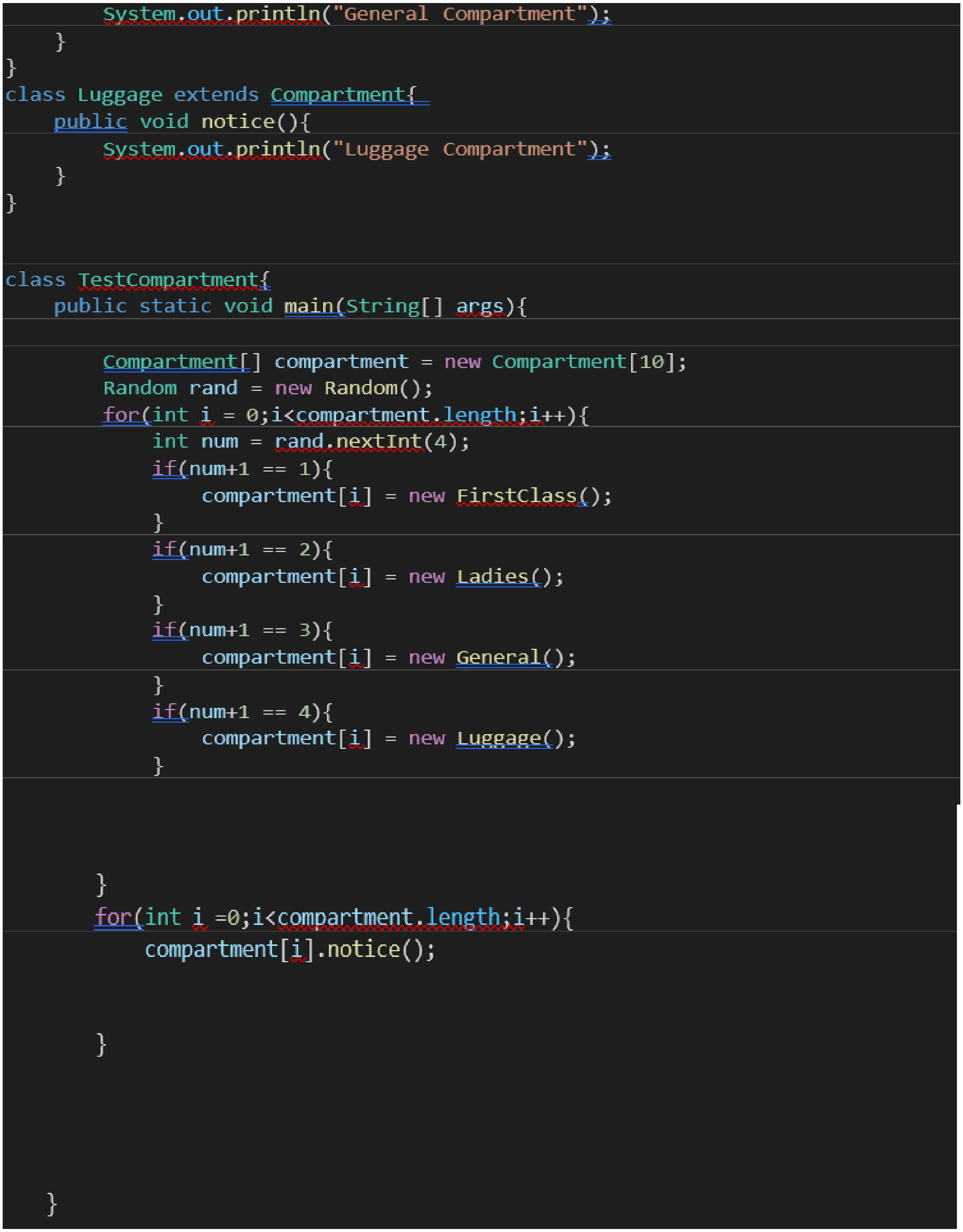
generated integer in the range 1 to 4. Check the polymorphic

behaviour of the notice method.

CODE:



```
import java.util.Random;

abstract  class  Compartment{
    public abstract void notice();



}

class FirstClass extends Compartment{
    public void notice(){
        System.out.println("First Class Compartment");
    }
}
class Ladies extends Compartment{

    public void notice(){
        System.out.println("Ladies Compartment");
    }
}
class General extends Compartment{
    public void notice(){
```



```
        System.out.println("General Compartment");
    }
}
class Luggage extends Compartment{ 
    public void notice(){
        System.out.println("Luggage Compartment");
    }
}


class TestCompartment{
    public static void main(String[] args){

        Compartment[] compartment = new Compartment[10];
        Random rand = new Random();
        for(int i = 0;i<compartment.length;i++){
            int num = rand.nextInt(4);
            if(num+1 == 1){
                compartment[i] = new FirstClass();
            }
            if(num+1 == 2){
                compartment[i] = new Ladies();
            }
            if(num+1 == 3){
                compartment[i] = new General();
            }
            if(num+1 == 4){
                compartment[i] = new Luggage();
            }



        }
        for(int i =0;i<compartment.length;i++){
            compartment[i].notice();


        }




    }
```

Q.3 Create a class Medicine to represent a drug manufactured by a pharmaceutical company. Provide a function displayLabel() in this class to print Name and address of the company.

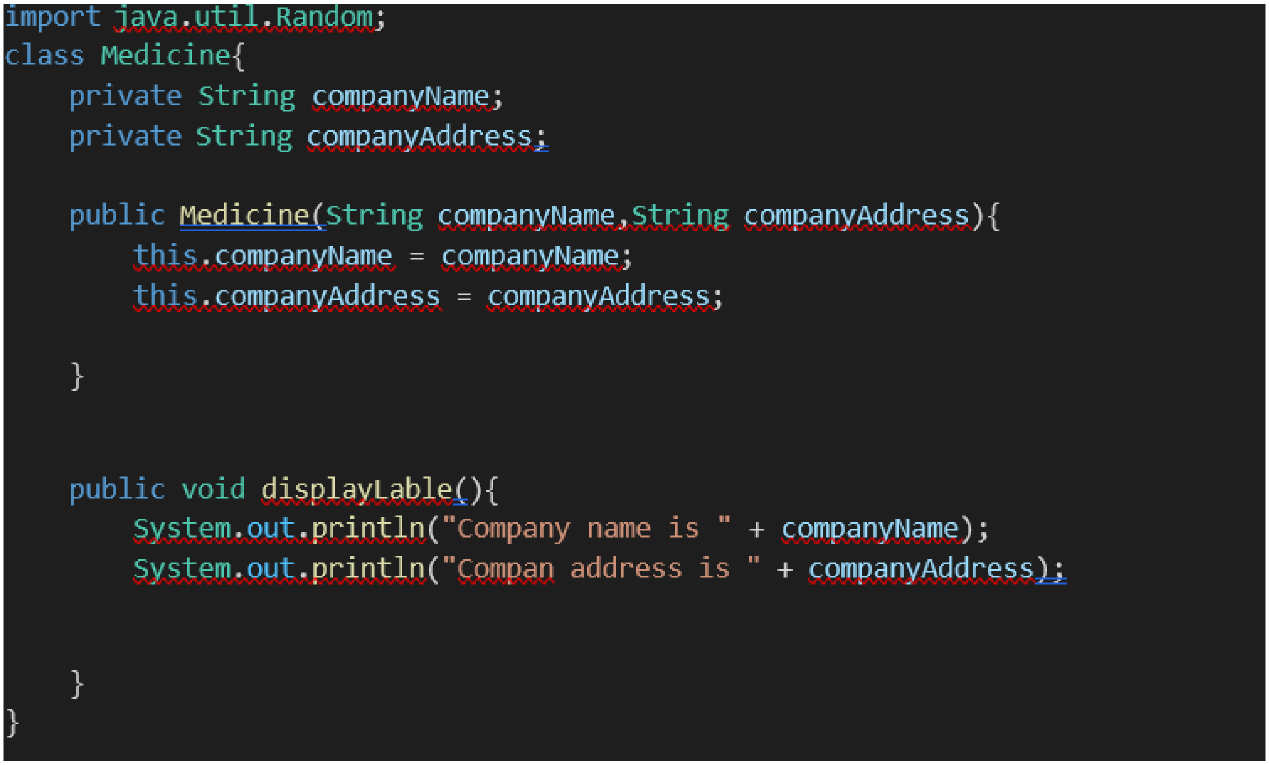
Derive Tablet, Syrup and Ointment classes from the Medicine class. Override the displayLabel() function in each of these classes to print additional information suitable to the type of medicine. For example, in case of tablets, it could be "store in a cool dry place", in case of ointments it could be "for external use only" etc.

Create a class TestMedicine . Write main function to do the following: Declare an array of Medicine references of size 10

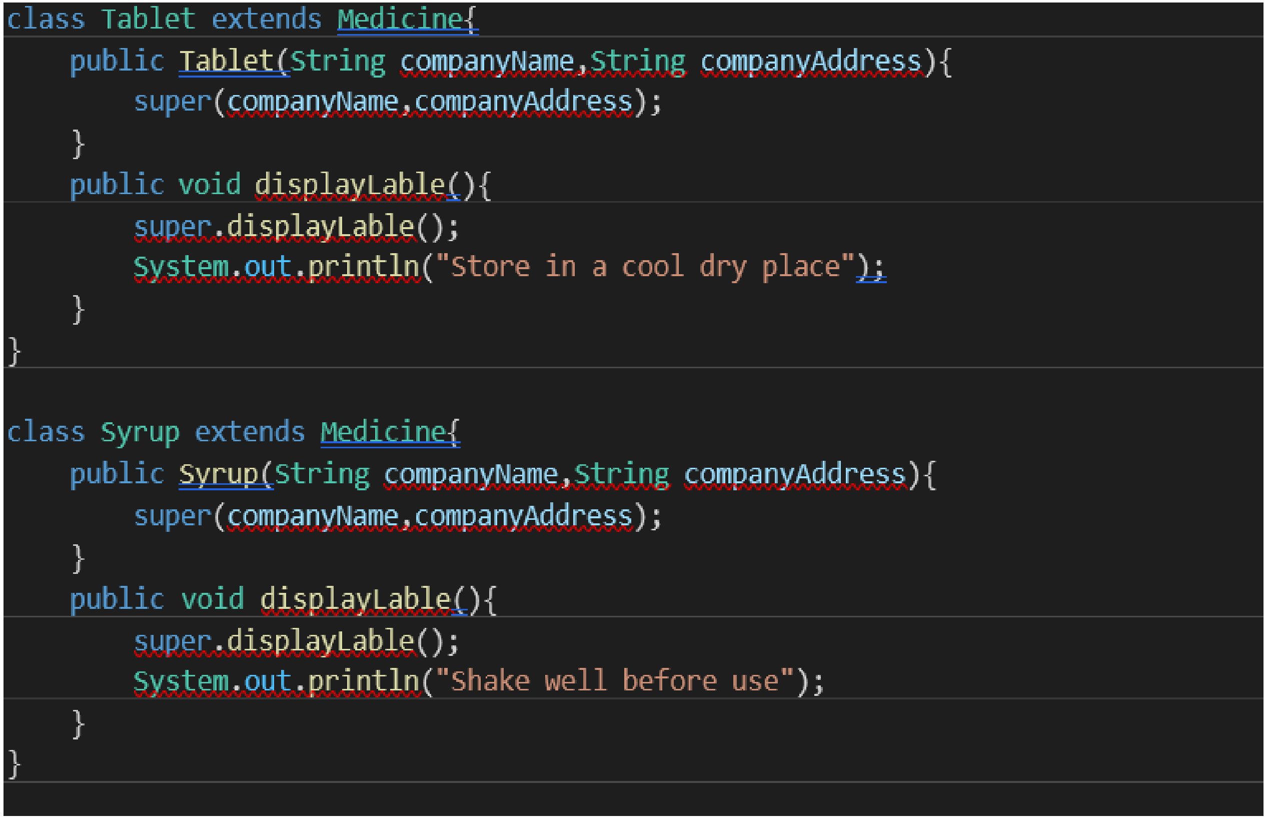
Create a medicine object of the type as decided by a randomly generated integer in the range 1 to 3.

Refer Java API Documentation to find out random generation feature. Check the polymorphic behaviour of the displayLabel() method.

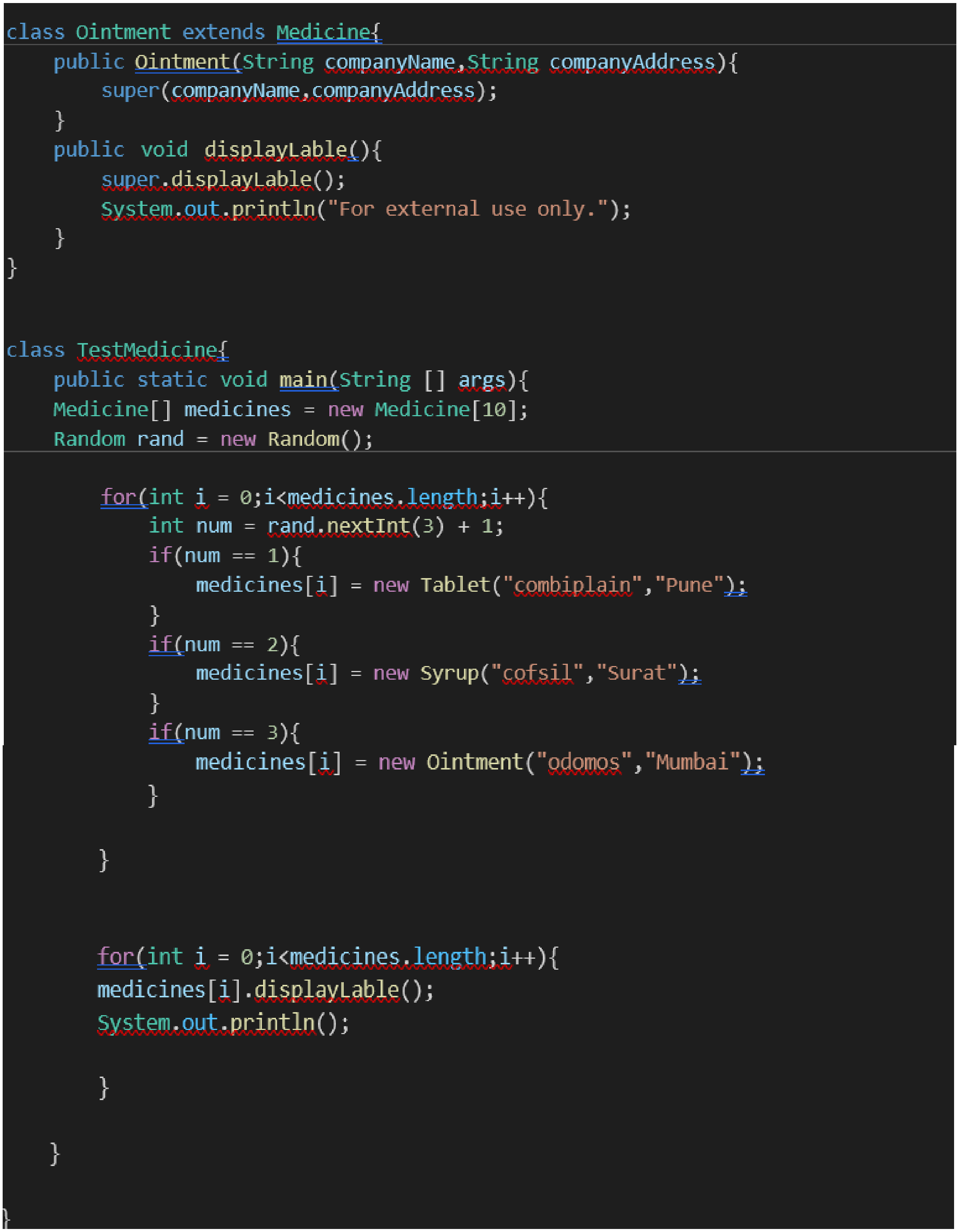
CODE:



```
import java.util.Random;
class Medicine{
    private String companyName;
    private String companyAddress;

    public Medicine(String companyName,String companyAddress){
        this.companyName = companyName;
        this.companyAddress = companyAddress;

    }


    public void displayLable(){
        System.out.println("Company name is " + companyName);
        System.out.println("Compan address is " + companyAddress);


    }
}
```



```
class Tablet extends Medicine{
    public Tablet(String companyName,String companyAddress){
        super(companyName,companyAddress);
    }
    public void displayLable(){
        super.displayLable();
        System.out.println("Store in a cool dry place");
    }
}

class Syrup extends Medicine{
    public Syrup(String companyName,String companyAddress){
        super(companyName,companyAddress);
    }
    public void displayLable(){
        super.displayLable();
        System.out.println("Shake well before use");
    }
}
```

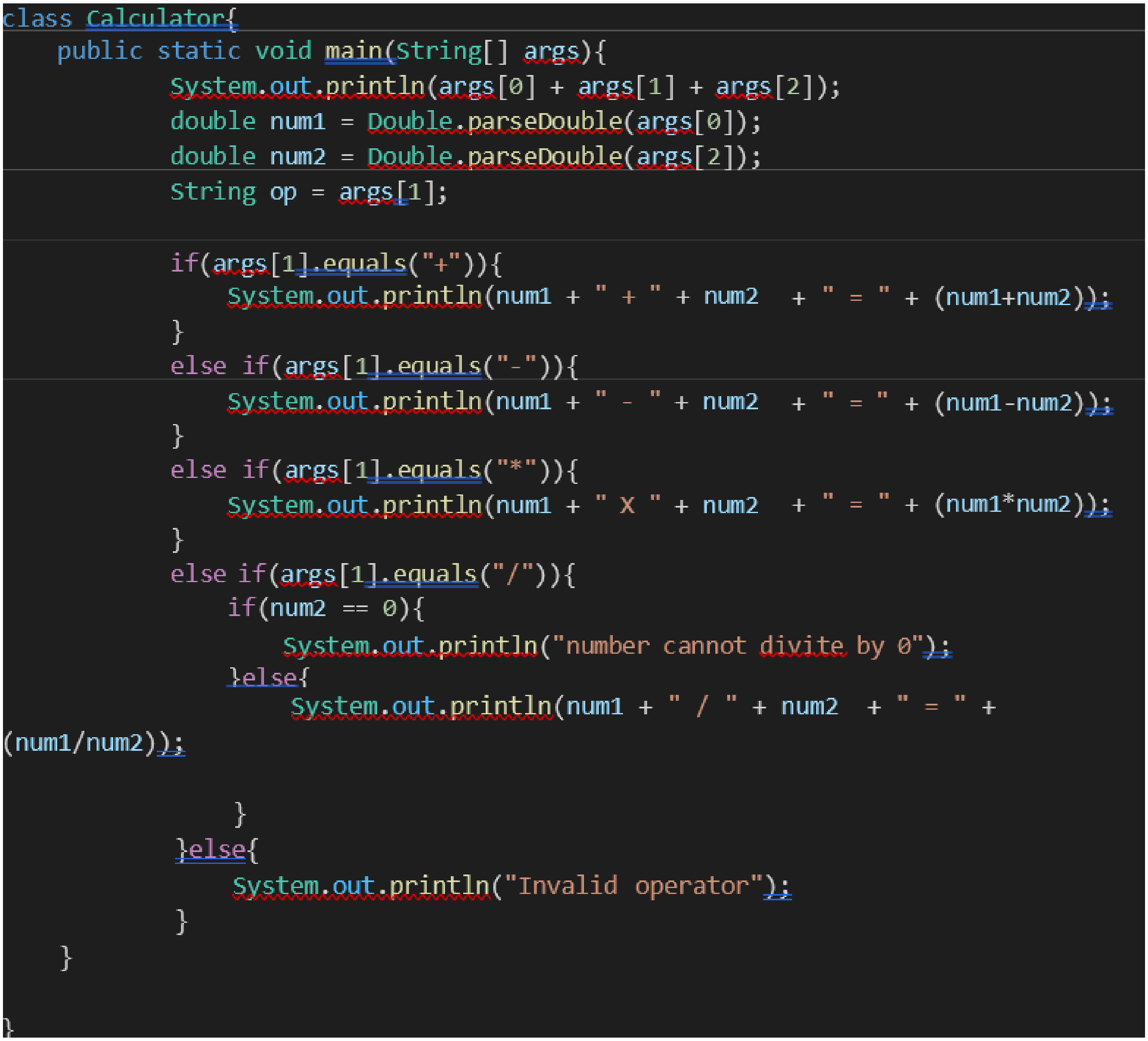


```
class Ointment extends Medicine{
    public Ointment(String companyName,String companyAddress){
        super(companyName,companyAddress);
    }
    public  void  displayLable(){
        super.displayLable();
        System.out.println("For external use only.");
    }
}


class TestMedicine{
    public static void main(String [] args){
    Medicine[] medicines = new Medicine[10];
    Random rand = new Random();

        for(int i = 0;i<medicines.length;i++){
            int num = rand.nextInt(3) + 1;
            if(num == 1){
                medicines[i] = new Tablet("combiplain","Pune");
            }
            if(num == 2){
                medicines[i] = new Syrup("cofsil","Surat");
            }
            if(num == 3){
                medicines[i] = new Ointment("odomos","Mumbai");
            }

        }


        for(int i = 0;i<medicines.length;i++){
        medicines[i].displayLable();
        System.out.println();

        }

    }

}
```

Q.4 Write a program that accepts two numbers and a operator like (+,-,*, /) as command line arguments and perform the appropriate operation indicated by operator.

If the user enters any other character the appropriate message will be displayed. The output of the program should be displayed to the user.

CODE:



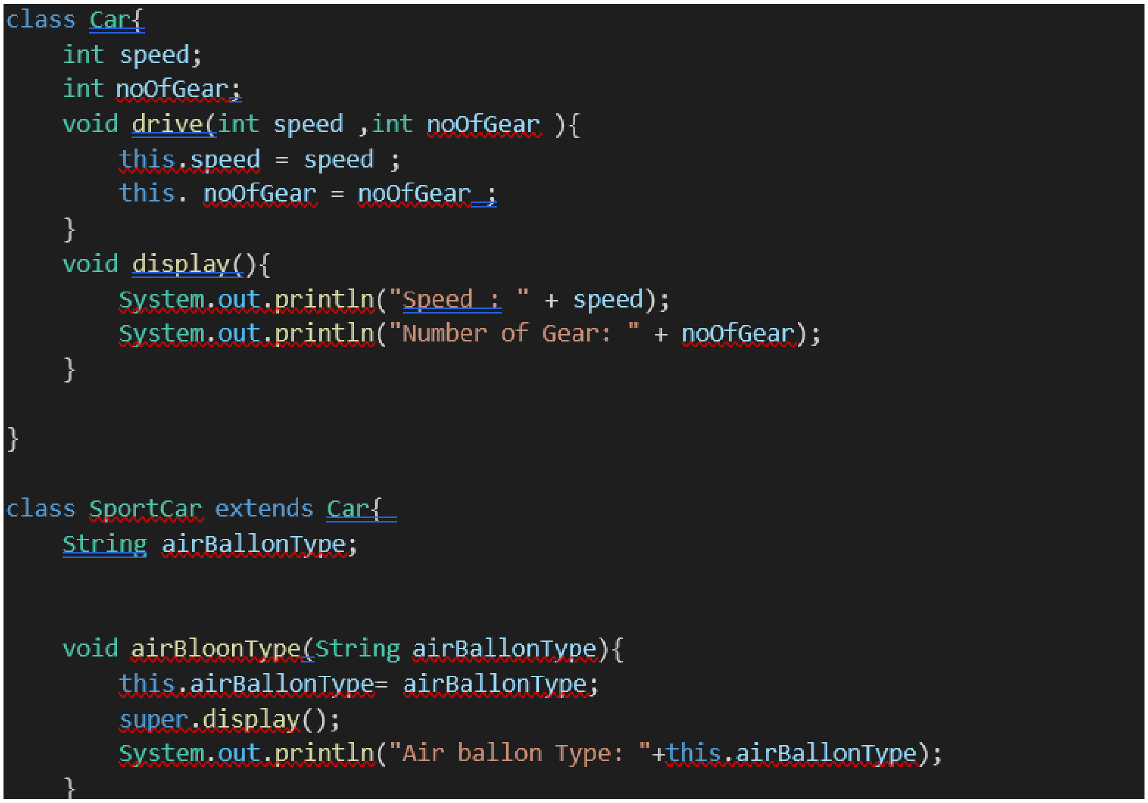
```
class Calculator{
    public static void main(String[] args){
            System.out.println(args[0] + args[1] + args[2]);
            double num1 = Double.parseDouble(args[0]);
            double num2 = Double.parseDouble(args[2]);
            String op = args[1];

            if(args[1].equals("+")){
                System.out.println(num1 + " + " + num2   + " = " + (num1+num2));
            }
            else if(args[1].equals("-")){
                System.out.println(num1 + " - " + num2   + " = " + (num1-num2));
            }
            else if(args[1].equals("*")){
                System.out.println(num1 + " X " + num2   + " = " + (num1*num2));
            }
            else if(args[1].equals("/")){
                if(num2 == 0){
                    System.out.println("number cannot divite by 0");
                }else{
                     System.out.println(num1 + " / " + num2   + " = " +
(num1/num2));

                }
            }else{
                System.out.println("Invalid operator");
            }
    }

}
```

Q.5 Create a class Car which contains members speed, noOfGear. The class has a method drive() which is responsible to provide starting speed and noOfGears to a Car. Implement display() method which will display all attributes of Car class.

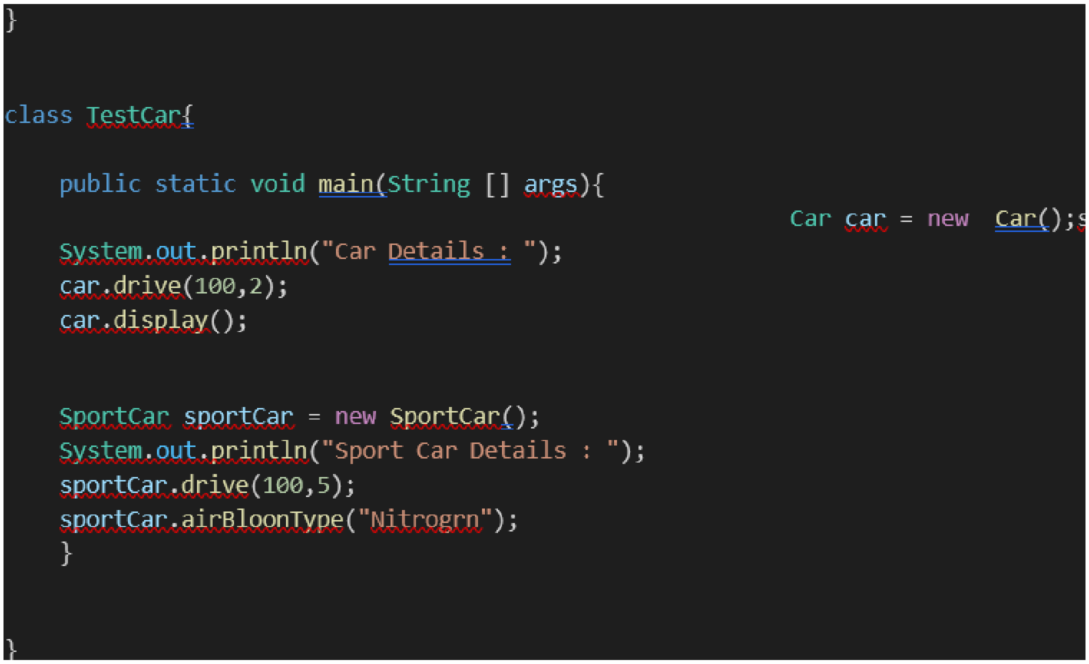
The class SportCar is derived from the class Car which adds new features AirBallonType. When this method is invoked, initial speed and gear status must be displayed on console. Override the display method which display all

attribute of the SportCar. Make use of super class display() method.

CODE:



```
class Car{
    int speed;
    int noOfGear;
    void drive(int speed ,int noOfGear ){
        this.speed = speed ;
        this. noOfGear = noOfGear ;
    }
    void display(){
        System.out.println("Speed : " + speed);
        System.out.println("Number of Gear: " + noOfGear);
    }

}

class SportCar extends Car{
    String airBallonType;


    void airBloonType(String airBallonType){
        this.airBallonType= airBallonType;
        super.display();
        System.out.println("Air ballon Type: "+this.airBallonType);
    }
```



```
}


class TestCar{

    public static void main(String [] args){
                                                                  Car car = new  Car();s
    System.out.println("Car Details : ");
    car.drive(100,2);
    car.display();


    SportCar sportCar = new SportCar();
    System.out.println("Sport Car Details : ");
    sportCar.drive(100,5);
    sportCar.airBloonType("Nitrogrn");
    }


}
```

ASSIGNMENTS NO. 3

Objective: At the end of the assignments, participants will be able to understand and create their own packages and also handle Exceptions.

Concept: Packages.

Q.1 Create a package techm.itp.hyd<your batch id>.cs/ncs<your emp id>.<your first name>.

For e.g., techm.itp.hyd10001.cs35123.Anu

Now create a Greeter class in this package having the following features:

Attributes:

name string //indicates name of the person to be greeted Member functions:

Greeter(aName) //constructor to initialize the name of the //person to be greeted by this greeter. sayHello()//returns a hello message with the name of the //person initialized earlier. sayGoodBye() //bids goodbye to the person named earlier.

Create another class in the same package called Advisor that has the following features: Attributes:

message string[5] //contains five advice messages Member functions:

Advisor() //default constructor to initialize an array of //strings with atleast five advice messages getAdvice() //randomly selects an advice from the available //list of messages and returns it to the caller of //this method

Outside the package, from your working directory, create a class GreeterTest that constructs Greeter objects for all command-line arguments and prints out the results of calling sayHello().

The program should then display an advice and finally bid goodbye to each of the persons/entities in reverse order of the names entered at the command line.

For e.g.,

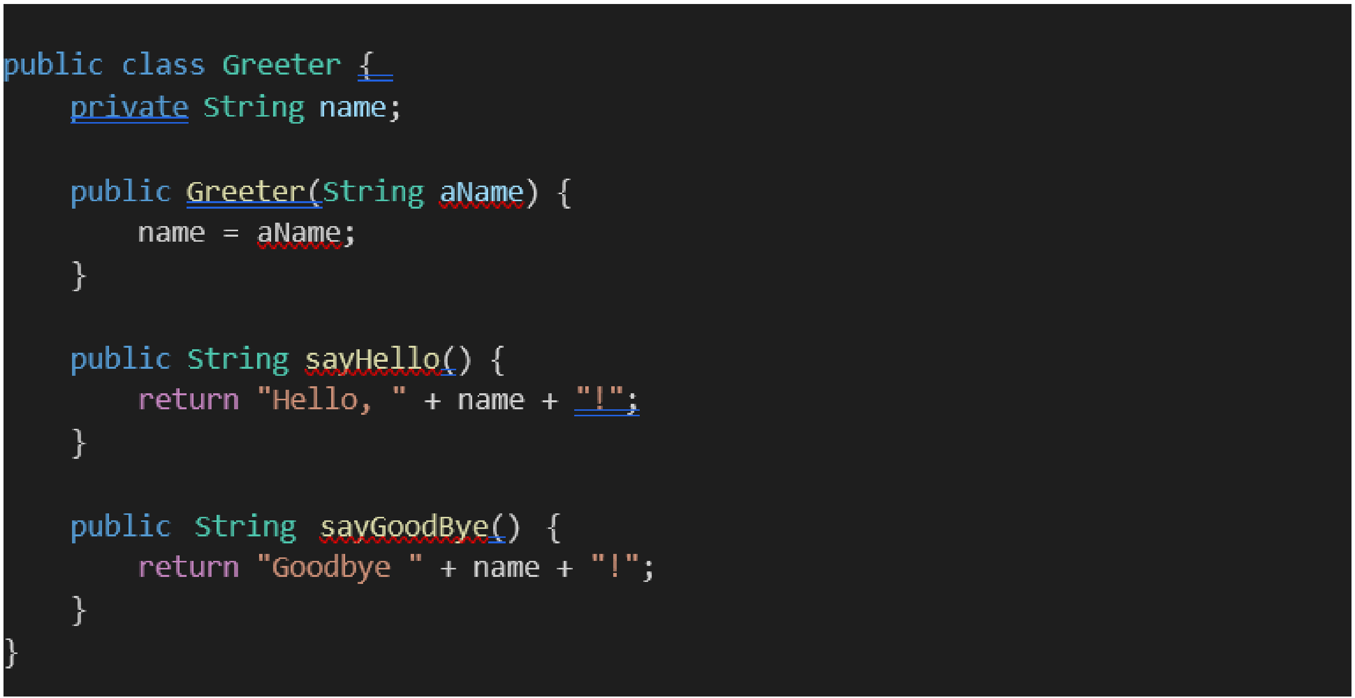
java GreeterTest Mars Venus

then the program should

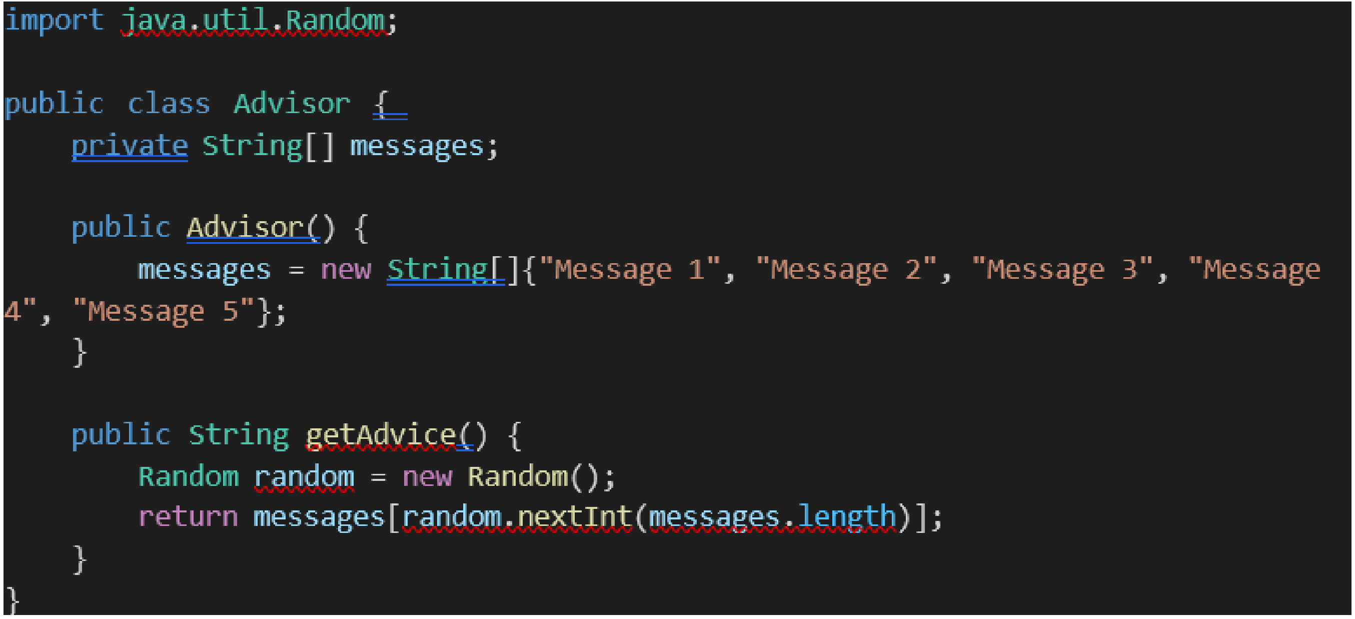
print.

```
Hello, Mars!
Hello,Venu!
Advice: Never say No
Goodbye Venus!
Goodbye Mars!
```

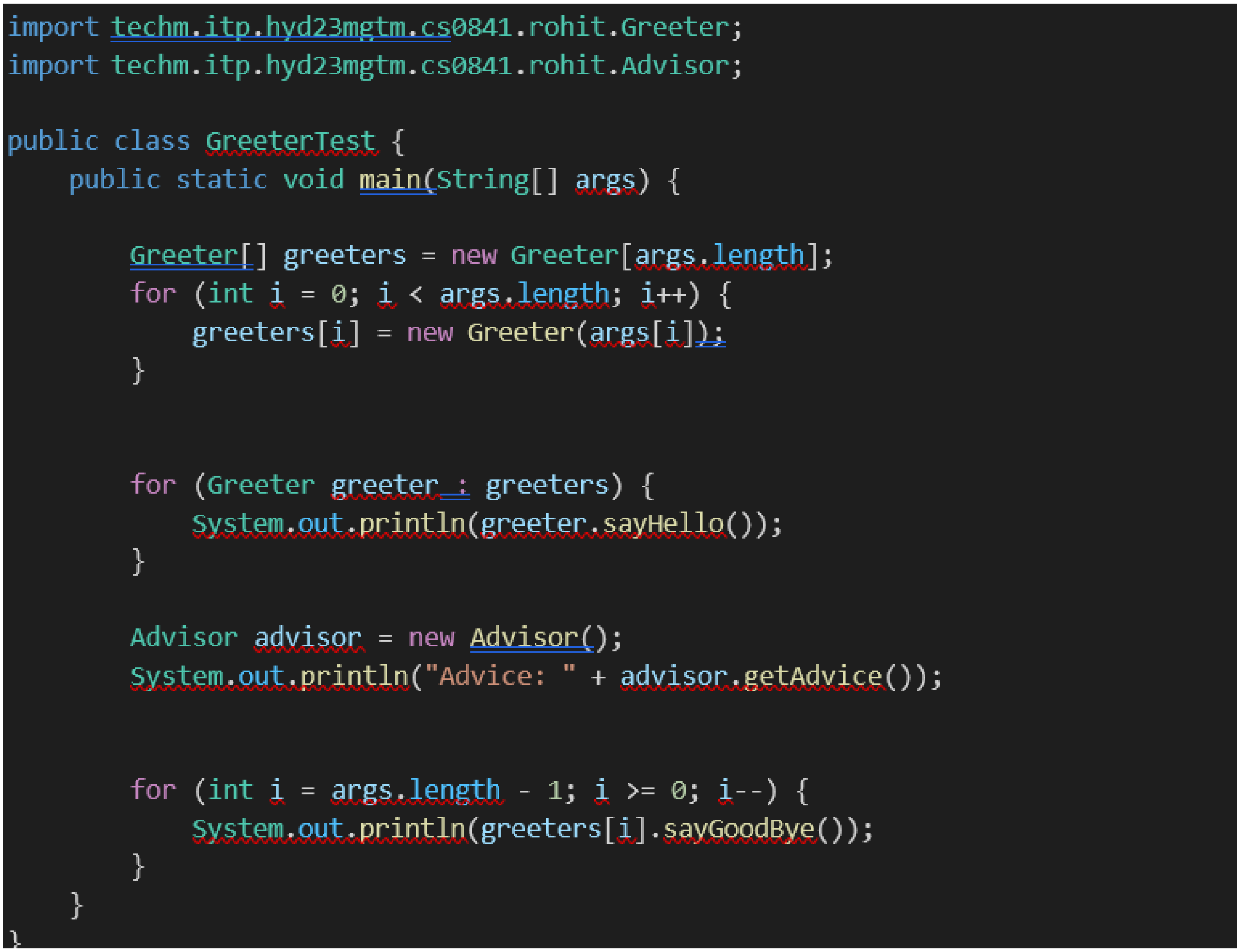
CODE:



```java
public class Greeter {
    private String name;

    public Greeter(String aName) {
        name = aName;
    }

    public String sayHello() {
        return "Hello, " + name + "!";
    }

    public String sayGoodBye() {
        return "Goodbye " + name + "!";
    }
}
```

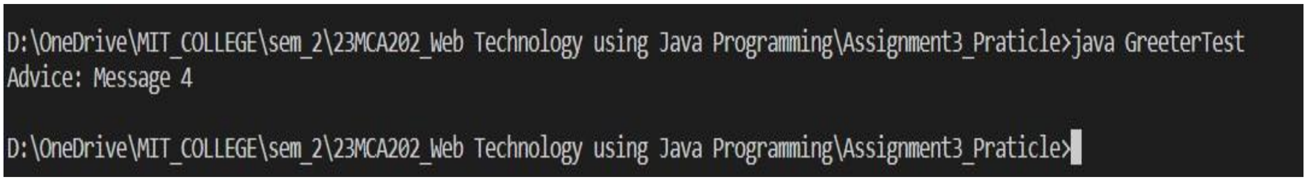


```java
import java.util.Random;

public class Advisor {
    private String[] messages;

    public Advisor() {
        messages = new String[]{"Message 1", "Message 2", "Message 3", "Message 4", "Message 5"};
    }

    public String getAdvice() {
        Random random = new Random();
        return messages[random.nextInt(messages.length)];
    }
}
```



```
import techm.itp.hyd23mgtm.cs0841.rohit.Greeter;
import techm.itp.hyd23mgtm.cs0841.rohit.Advisor;

public class GreeterTest {
    public static void main(String[] args) {

        Greeter[] greeters = new Greeter[args.length];
        for (int i = 0; i < args.length; i++) {
            greeters[i] = new Greeter(args[i]);
        }


        for (Greeter greeter : greeters) {
            System.out.println(greeter.sayHello());
        }

        Advisor advisor = new Advisor();
        System.out.println("Advice: " + advisor.getAdvice());


        for (int i = args.length - 1; i >= 0; i--) {
            System.out.println(greeters[i].sayGoodBye());
        }
    }
}
```

OUTPUT:



```
D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment3_Praticle>java GreeterTest
Advice: Message 4

D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment3_Praticle>
```

Q.2 Create a package esg.itp.shape containing the following classes and interfaces.

An interface Polygon containing the members

as given below: area float perimeter float

void calcArea( ); abstract method to calculate area of a particular polygon given its dimensions

void calcPeri( ); abstract method to calculate perimeter of a particular polygon given its dimensions

void display( ); method to display the area and perimeter of the given polygon

Create a class Square that implements Polygon and

has the following member: side float

Square(float s); constructor to initialize side of square

Create another class Rectangle that implements Polygon and has

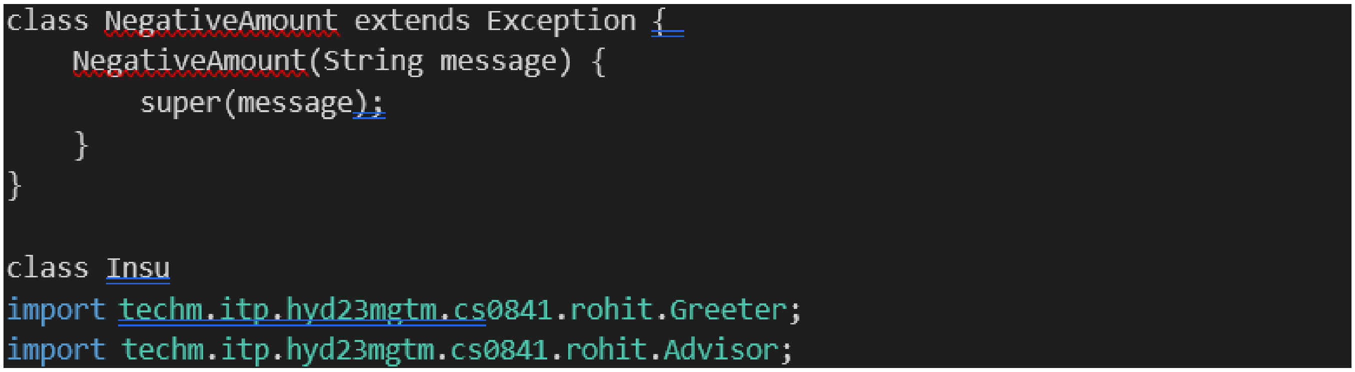
the following member: length float
breadth float

Rectangle(int len, int bre); constructor to initialize length and breadth of a rectangle

Outside the package, create a class that imports the above package an instantiates an object of the Square class and an object of the Rectangle class.

Call the above methods on each of the classes to calculate the area and perimeter given the side and the length/breadth of the Square class and the Rectangle class respectively.

CODE:



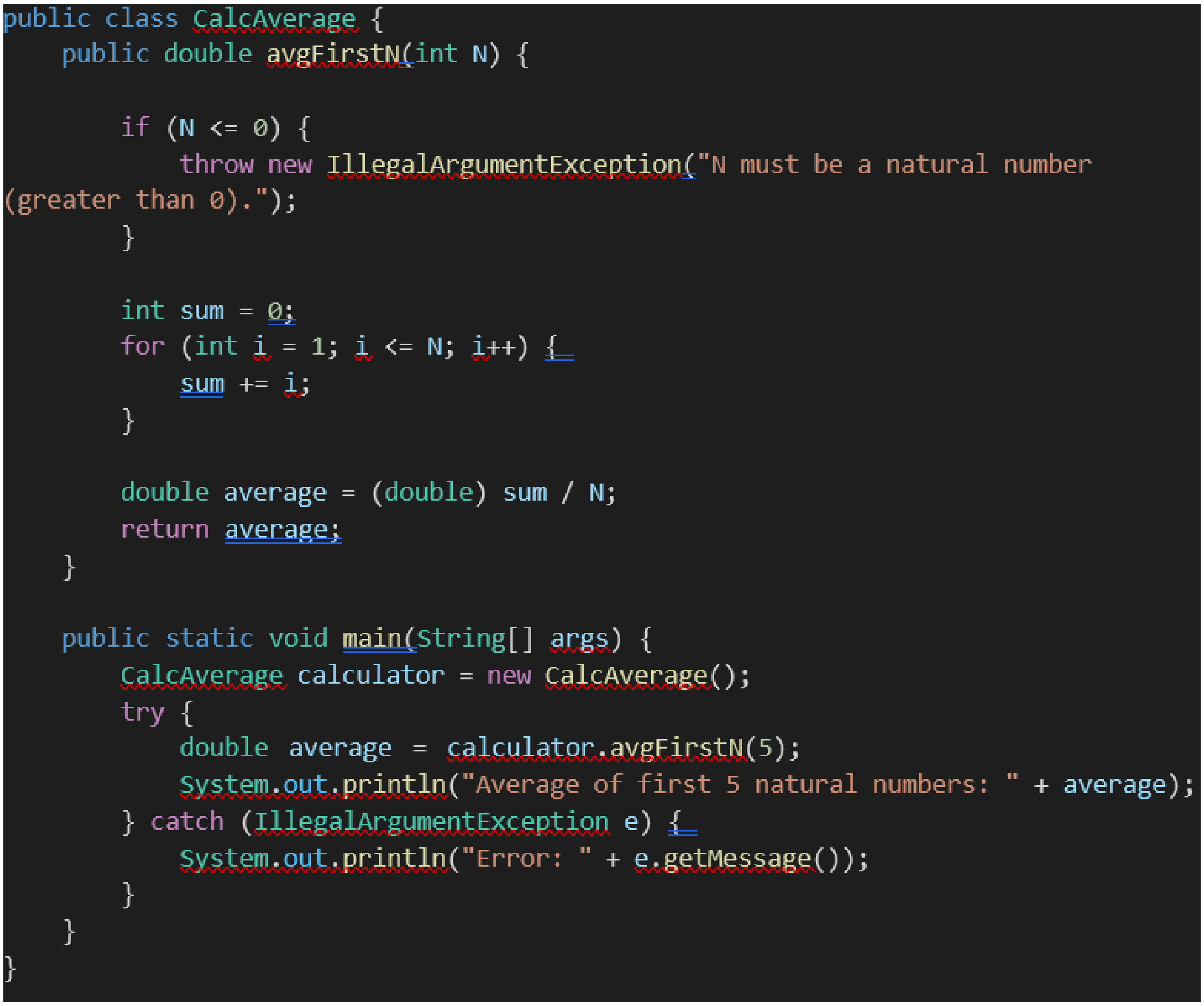
```
class NegativeAmount extends Exception {
    NegativeAmount(String message) {
        super(message);
    }
}

class Insu
import techm.itp.hyd23mgtm.cs0841.rohit.Greeter;
import techm.itp.hyd23mgtm.cs0841.rohit.Advisor;
```

Q.3 Create a class called CalcAverage that has the following

method: public double avgFirstN(int N)

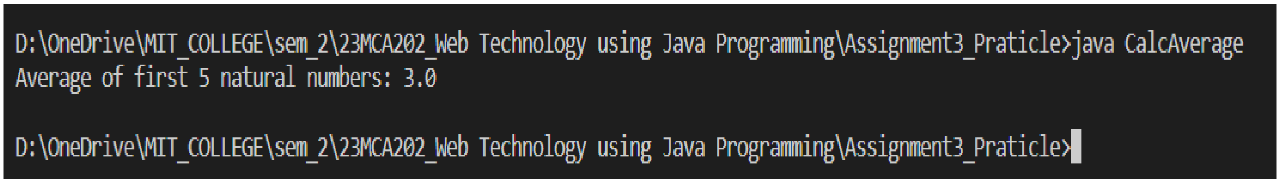
This method receives an integer as a parameter and calculates the average of first N natural numbers. If N is not a natural number, throw an exception IllegalArgumentException with an appropriate message.

<u>CODE:</u>



```java
public class CalcAverage {
    public double avgFirstN(int N) {

        if (N <= 0) {
            throw new IllegalArgumentException("N must be a natural number
(greater than 0).");
        }

        int sum = 0;
        for (int i = 1; i <= N; i++) {
            sum += i;
        }

        double average = (double) sum / N;
        return average;
    }

    public static void main(String[] args) {
        CalcAverage calculator = new CalcAverage();
        try {
            double  average  =  calculator.avgFirstN(5);
            System.out.println("Average of first 5 natural numbers: " + average);
        } catch (IllegalArgumentException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

OUTPUT:



```
D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment3_Praticle>java CalcAverage
Average of first 5 natural numbers: 3.0

D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment3_Praticle>
```

Q.4 Create a class Number having the following features:

Attributes

int first number

int second number

result double stores the result of arithmetic operations performed on a and b

Member functions

Number(x, y) constructor to initialize the values of a
and b add() stores the sum of a and b in result

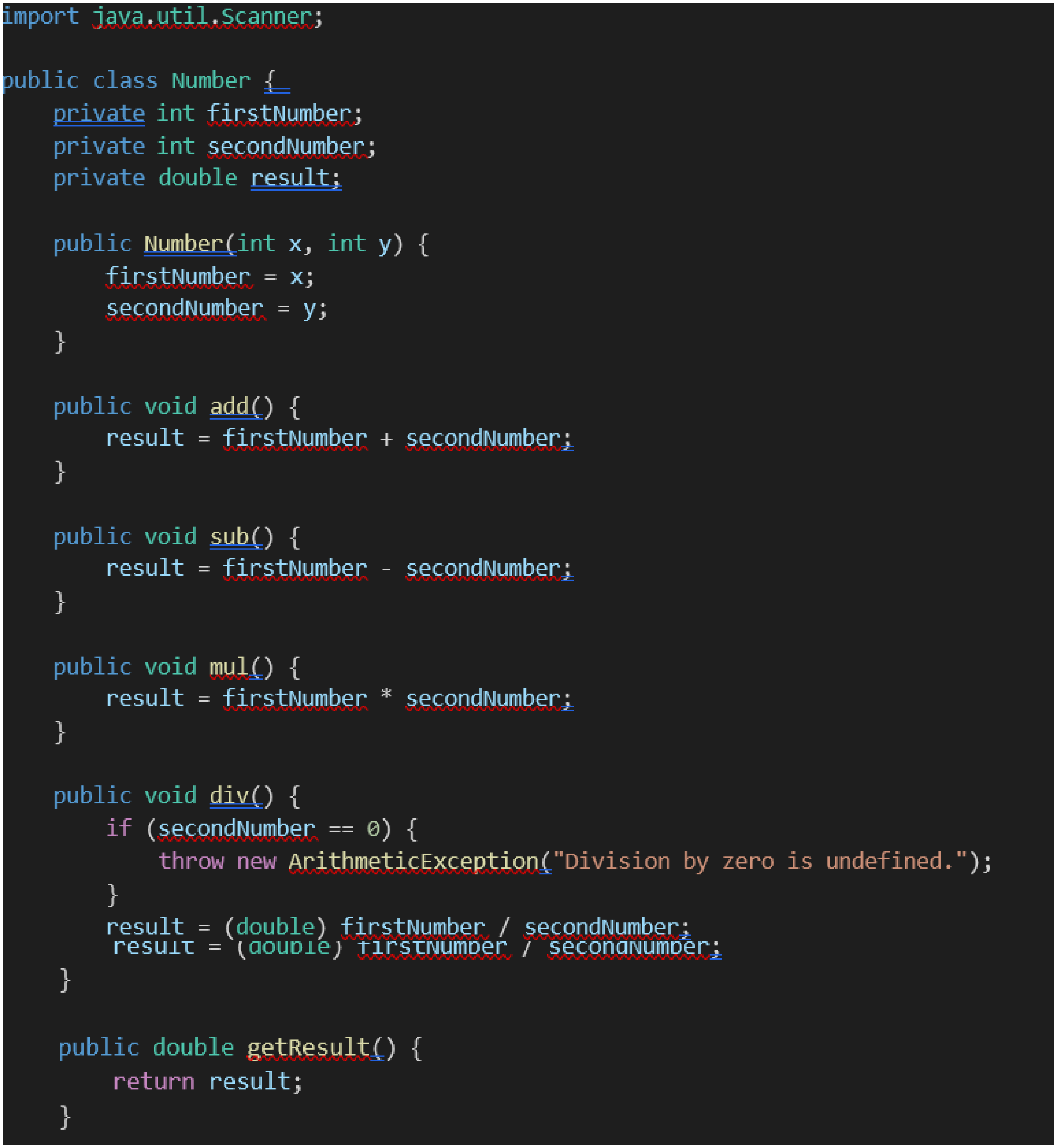
sub() stores difference of a and b in result

mul() stores product in result

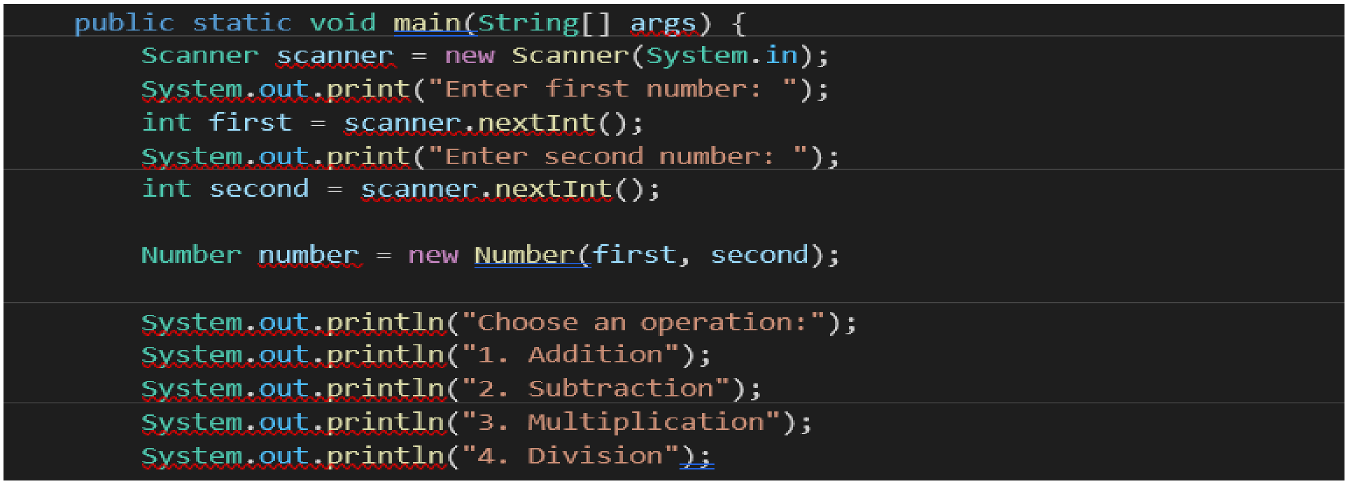
div() stores a divided by b in result

Test to see if b is 0 and throw an appropriate exception since division by zero is undefined. Display a menu to the user to perform the above four arithmetic operations.

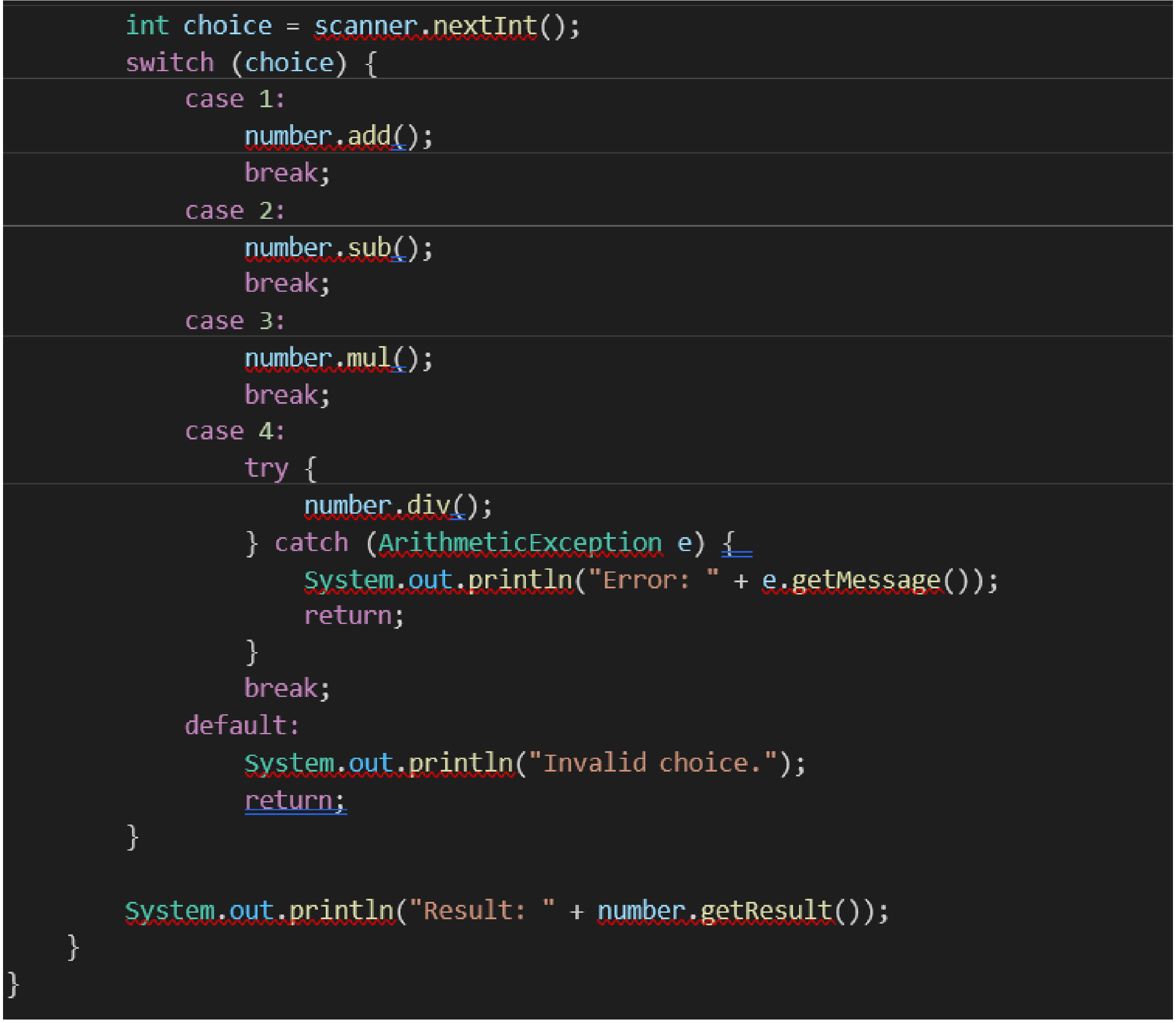
CODE:



```
import java.util.Scanner;

public class Number {
    private int firstNumber;
    private int secondNumber;
    private double result;

    public Number(int x, int y) {
        firstNumber = x;
        secondNumber = y;
    }

    public void add() {
        result = firstNumber + secondNumber;
    }

    public void sub() {
        result = firstNumber - secondNumber;
    }

    public void mul() {
        result = firstNumber * secondNumber;
    }

    public void div() {
        if (secondNumber == 0) {
            throw new ArithmeticException("Division by zero is undefined.");
        }
        result = (double) firstNumber / secondNumber;
        result = (double) firstNumber / secondNumber;
    }

    public double getResult() {
        return result;
    }
```

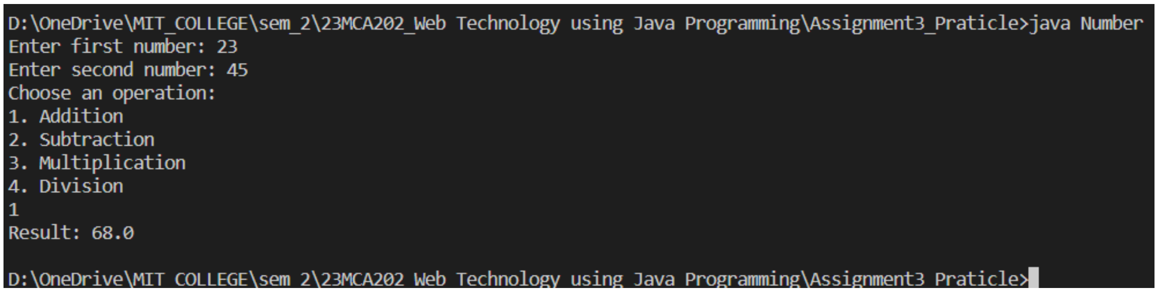


```
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter first number: ");
        int first = scanner.nextInt();
        System.out.print("Enter second number: ");
        int second = scanner.nextInt();

        Number number = new Number(first, second);

        System.out.println("Choose an operation:");
        System.out.println("1. Addition");
        System.out.println("2. Subtraction");
        System.out.println("3. Multiplication");
        System.out.println("4. Division");
```



```
        int choice = scanner.nextInt();
        switch (choice) {
            case 1:
                number.add();
                break;
            case 2:
                number.sub();
                break;
            case 3:
                number.mul();
                break;
            case 4:
                try {
                    number.div();
                } catch (ArithmeticException e) {
                    System.out.println("Error: " + e.getMessage());
                    return;
                }
                break;
            default:
                System.out.println("Invalid choice.");
                return;
        }

        System.out.println("Result: " + number.getResult());
    }
}
```

OUTPUT:



```
D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment3_Praticle>java Number
Enter first number: 23
Enter second number: 45
Choose an operation:
1. Addition
2. Subtraction
3. Multiplication
4. Division
1
Result: 68.0

D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment3_Praticle>
```

Q.6 Create a class BankAccount having the members as given below: accNo integer custName string

accType string (indicates 'Savings' or 'Current') balance float

Include the following methods in the BankAccount class:

Void
deposit(floatamt);
void
withdraw(float
amt); float
getBalance();

deposit(float amt) method allows you to credit an amount into the current balance. If amount is negative, throw an exception NegativeAmount to block the operation from being performed.

withdraw(float amt) method allows you to debit an amount from the current balance. Please ensure a minimum balance of Rs. 1000/- in the account for savings account and Rs. 5000/- for current account, else throw an exception InsufficientFunds and block the withdrawal operation. Also throw an exception NegativeAmount to block the operation from being performed if the amt parameter passed to this function is negative.

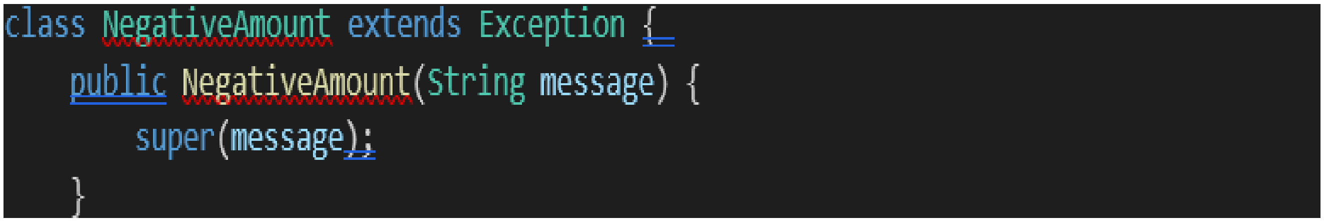
getBalance() method returns the current balance. If the current balance is below the minimum required balance, then throw an exception LowBalanceException accordingly.

Have constructor to which you will pass, accno, cust_name, acctype and initial balance.

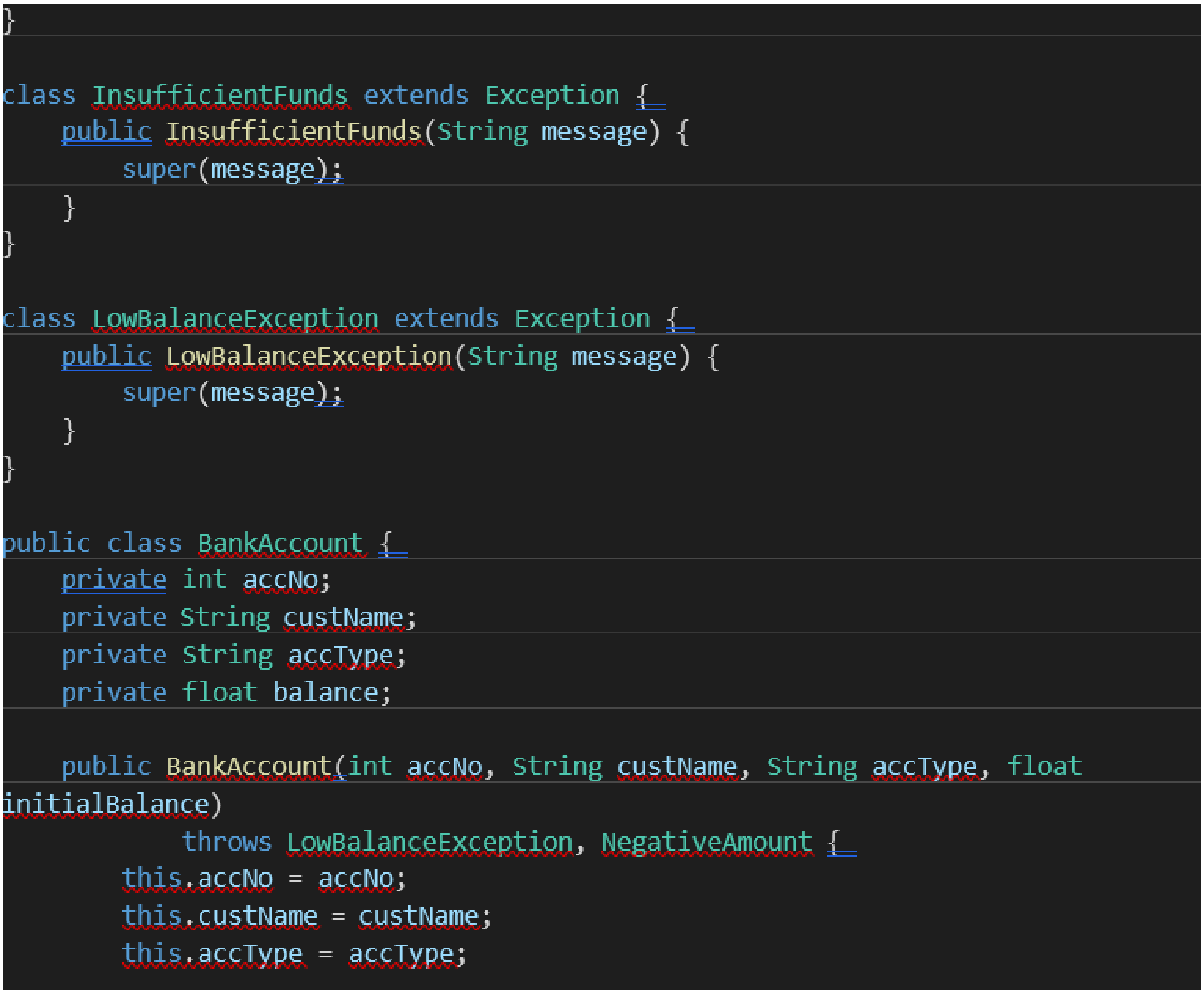
And check whether the balance is less than 1000 or not in case of savings account and less than 5000 in case of a current account. If so, then raise a LowBalanceException.

In either case if the balance is negative then raise the NegativeAmount exception accordingly.

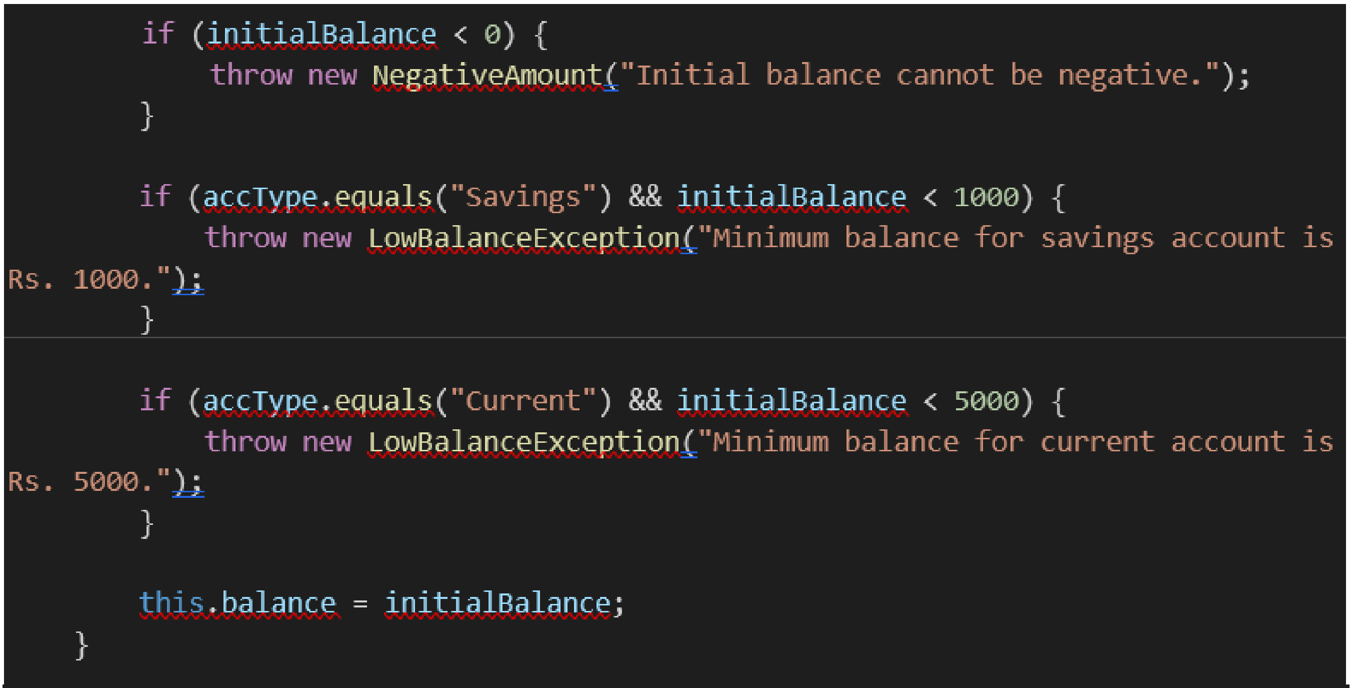
CODE:



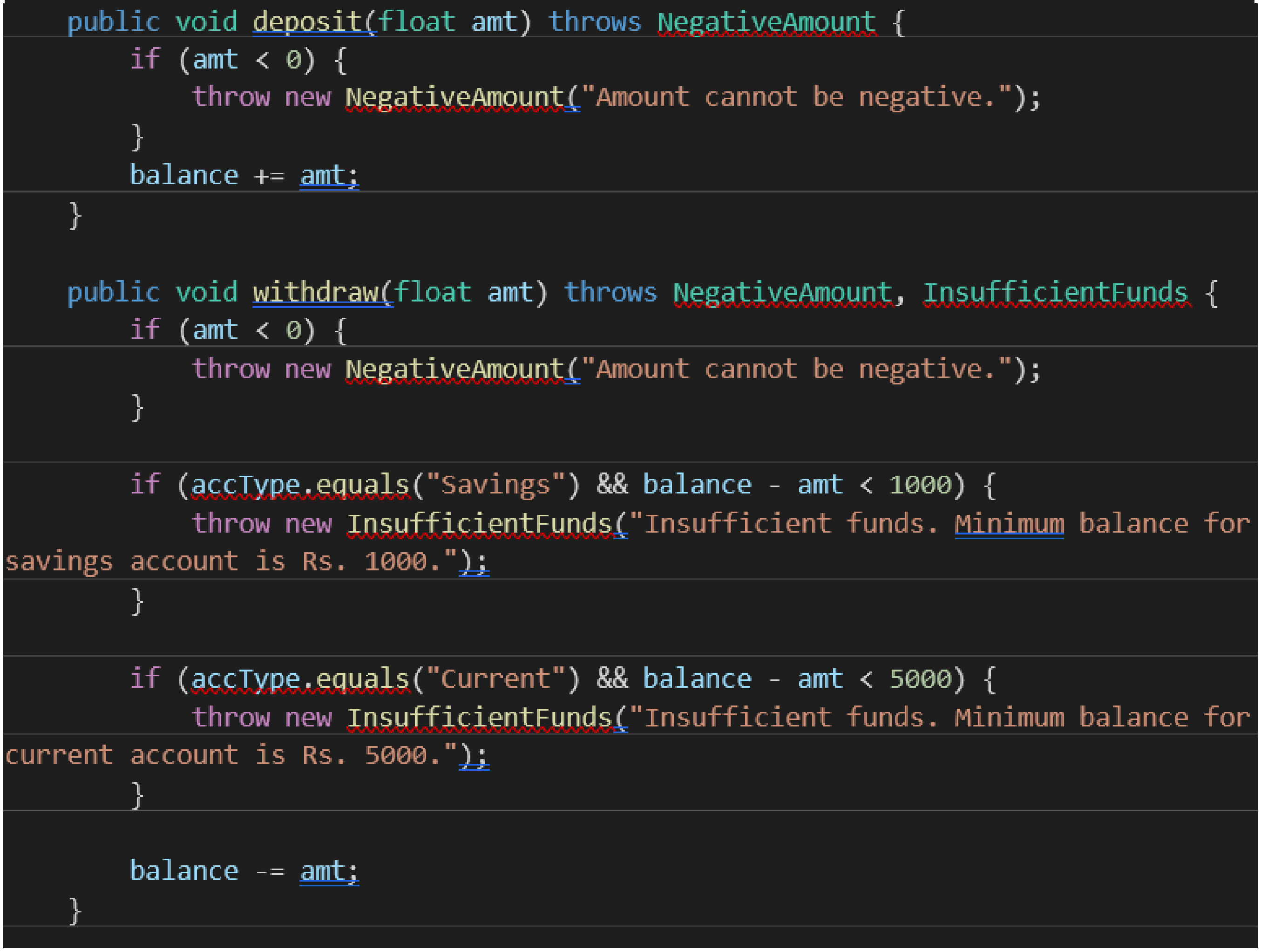
```
class NegativeAmount extends Exception {
    public NegativeAmount(String message) {
        super(message);
    }
```



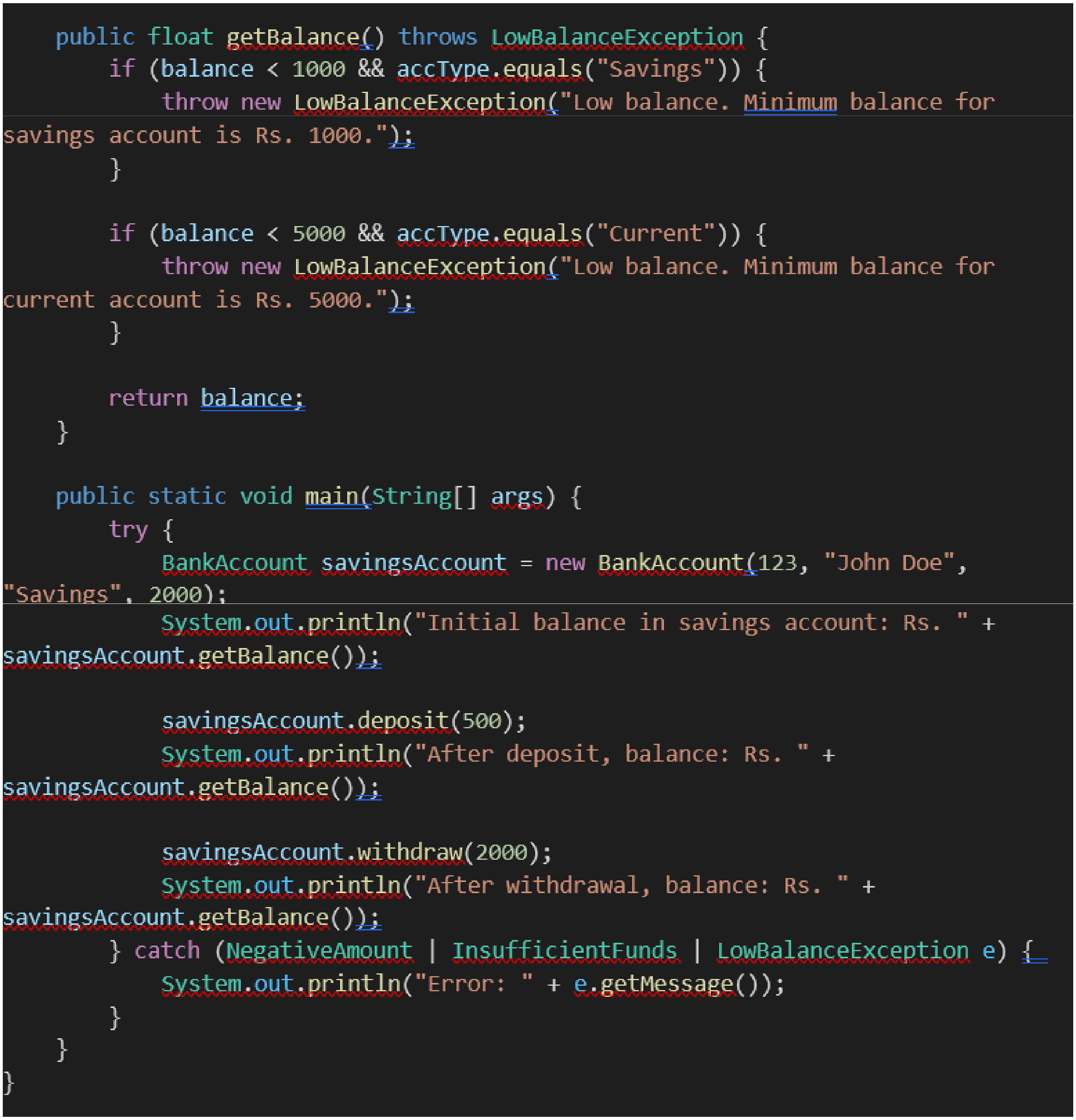
```
}

class InsufficientFunds extends Exception {
    public InsufficientFunds(String message) {
        super(message);
    }
}

class LowBalanceException extends Exception {
    public LowBalanceException(String message) {
        super(message);
    }
}

public class BankAccount {
    private int accNo;
    private String custName;
    private String accType;
    private float balance;

    public BankAccount(int accNo, String custName, String accType, float
initialBalance)
            throws LowBalanceException, NegativeAmount {
        this.accNo = accNo;
        this.custName = custName;
        this.accType = accType;
```



```
        if (initialBalance < 0) {
            throw new NegativeAmount("Initial balance cannot be negative.");
        }

        if (accType.equals("Savings") && initialBalance < 1000) {
            throw new LowBalanceException("Minimum balance for savings account is
Rs. 1000.");
        }

        if (accType.equals("Current") && initialBalance < 5000) {
            throw new LowBalanceException("Minimum balance for current account is
Rs. 5000.");
        }

        this.balance = initialBalance;
    }
```

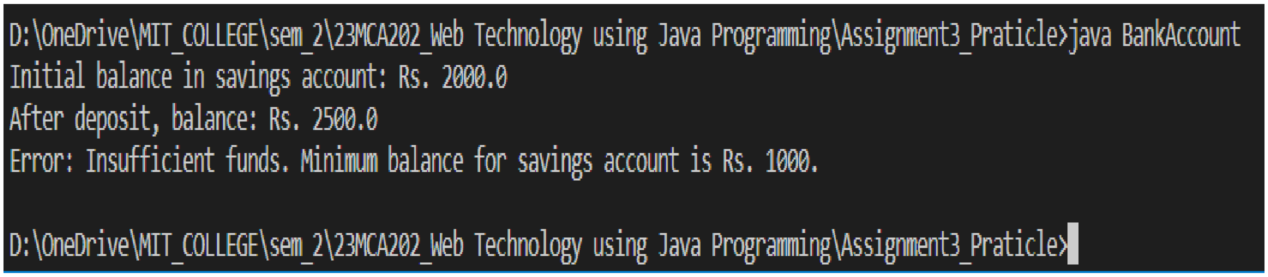


```
    public void deposit(float amt) throws NegativeAmount {
        if (amt < 0) {
            throw new NegativeAmount("Amount cannot be negative.");
        }
        balance += amt;
    }

    public void withdraw(float amt) throws NegativeAmount, InsufficientFunds {
        if (amt < 0) {
            throw new NegativeAmount("Amount cannot be negative.");
        }

        if (accType.equals("Savings") && balance - amt < 1000) {
            throw new InsufficientFunds("Insufficient funds. Minimum balance for
savings account is Rs. 1000.");
        }

        if (accType.equals("Current") && balance - amt < 5000) {
            throw new InsufficientFunds("Insufficient funds. Minimum balance for
current account is Rs. 5000.");
        }

        balance -= amt;
    }
```



```
    public float getBalance() throws LowBalanceException {
        if (balance < 1000 && accType.equals("Savings")) {
            throw new LowBalanceException("Low balance. Minimum balance for 
savings account is Rs. 1000.");
        }

        if (balance < 5000 && accType.equals("Current")) {
            throw new LowBalanceException("Low balance. Minimum balance for 
current account is Rs. 5000.");
        }

        return balance;
    }

    public static void main(String[] args) {
        try {
            BankAccount savingsAccount = new BankAccount(123, "John Doe", 
"Savings", 2000);
            System.out.println("Initial balance in savings account: Rs. " + 
savingsAccount.getBalance());

            savingsAccount.deposit(500);
            System.out.println("After deposit, balance: Rs. " + 
savingsAccount.getBalance());

            savingsAccount.withdraw(2000);
            System.out.println("After withdrawal, balance: Rs. " + 
savingsAccount.getBalance());
        } catch (NegativeAmount | InsufficientFunds | LowBalanceException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

OUTPUT:



```
D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment3_Praticle>java BankAccount
Initial balance in savings account: Rs. 2000.0
After deposit, balance: Rs. 2500.0
Error: Insufficient funds. Minimum balance for savings account is Rs. 1000.

D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment3_Praticle>
```

Q.7 Create a class with following specifications.

Class Emp

empId	**int**
empName	**string**
designation	**string**
basic	**double**
hra	**double** **readOnly**

Methods

printDET()

calculateHRA()

printDET() methods will show details of the EMP.

calculateHRA() method will calculate HRA based

on basic. There will 3 designations supported by

the application.

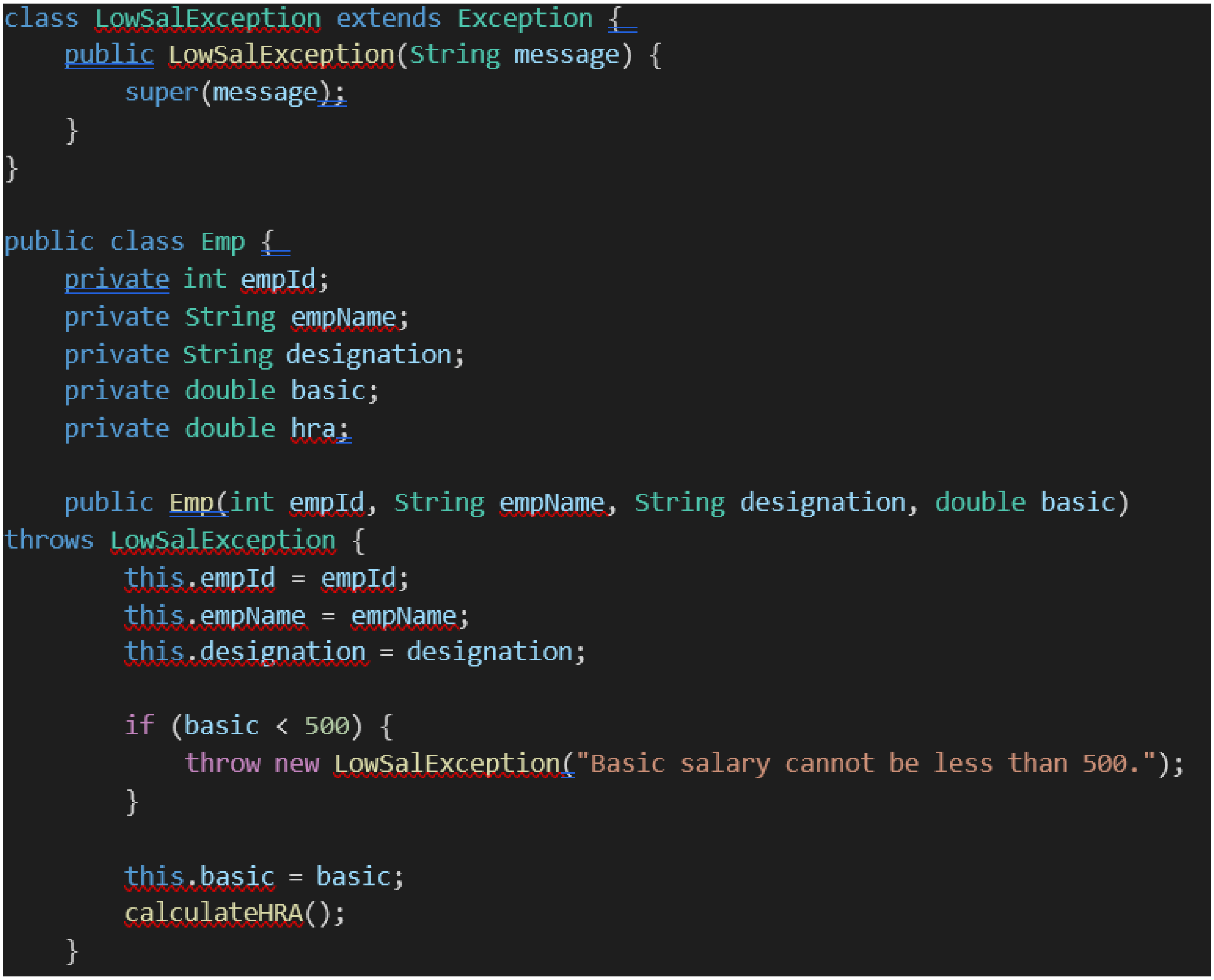
If designation is “Manager” - HRA will be 10% of BASIC
If designation is “Officer” - HRA will be 12% of BASIC
if category is “CLERK” - HRA will be 5% of BASIC

Have constructor to which you will pass, empId, designation, basic and price.

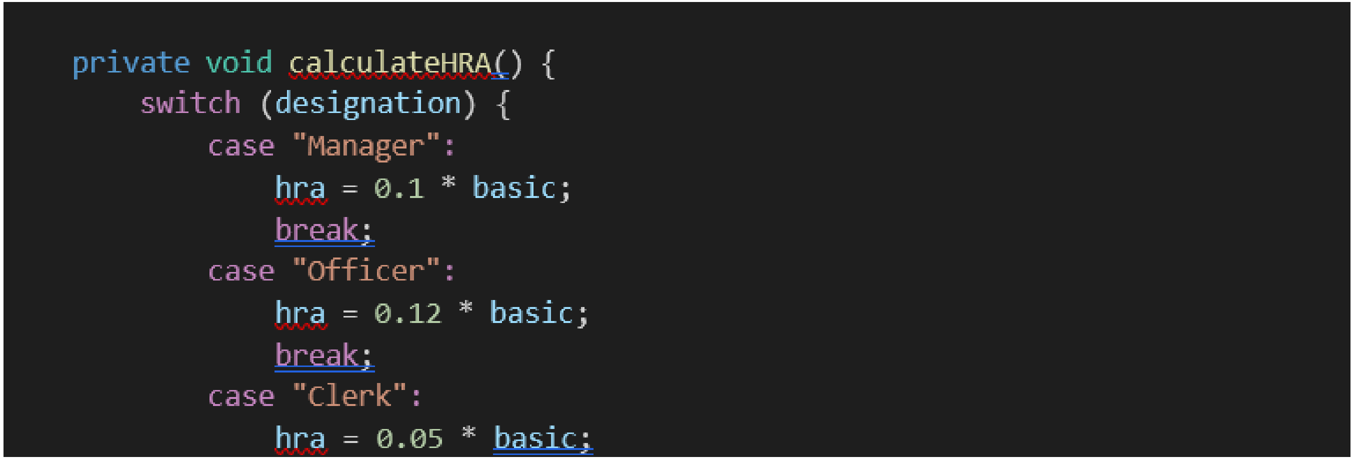
And checks whether the BASIC is less than 500 or not. If it is less than 500 raise a custom Exception as given below

Create LowSalException class with proper user message to handle BASIC less than 500.

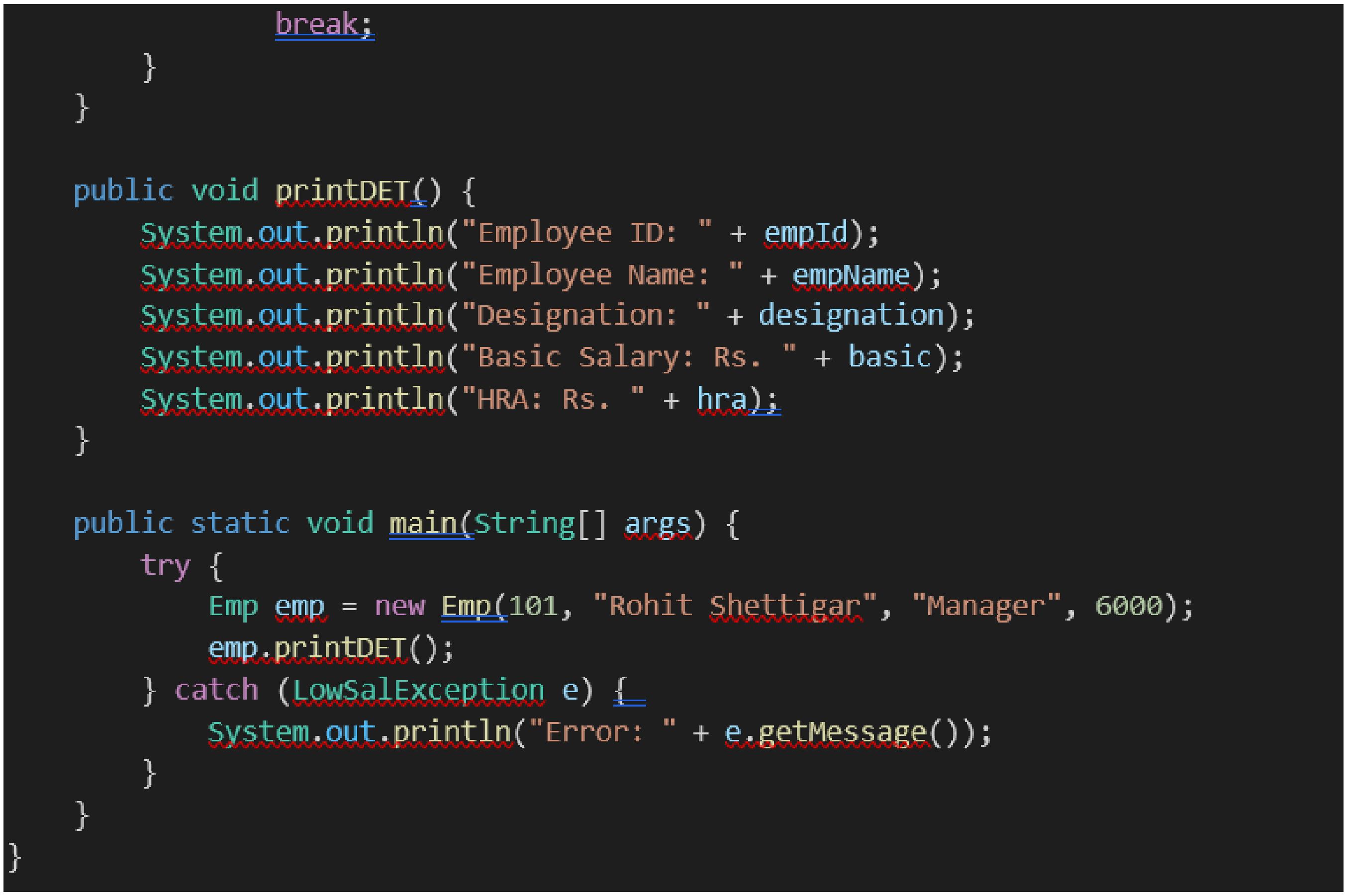
CODE:



```
class LowSalException extends Exception {
    public LowSalException(String message) {
        super(message);
    }
}

public class Emp {
    private int empId;
    private String empName;
    private String designation;
    private double basic;
    private double hra;

    public Emp(int empId, String empName, String designation, double basic)
throws LowSalException {
        this.empId = empId;
        this.empName = empName;
        this.designation = designation;

        if (basic < 500) {
            throw new LowSalException("Basic salary cannot be less than 500.");
        }

        this.basic = basic;
        calculateHRA();
    }
```

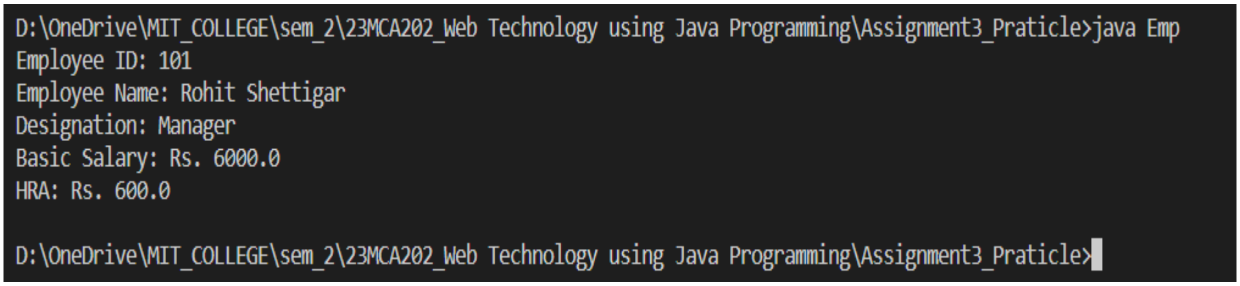


```

    private void calculateHRA() {
        switch (designation) {
            case "Manager":
                hra = 0.1 * basic;
                break;
            case "Officer":
                hra = 0.12 * basic;
                break;
            case "Clerk":
                hra = 0.05 * basic;
```



```
                break;
            }
        }

    public void printDET() {
        System.out.println("Employee ID: " + empId);
        System.out.println("Employee Name: " + empName);
        System.out.println("Designation: " + designation);
        System.out.println("Basic Salary: Rs. " + basic);
        System.out.println("HRA: Rs. " + hra);
    }

    public static void main(String[] args) {
        try {
            Emp emp = new Emp(101, "Rohit Shettigar", "Manager", 6000);
            emp.printDET();
        } catch (LowSalException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

OUTPUT:



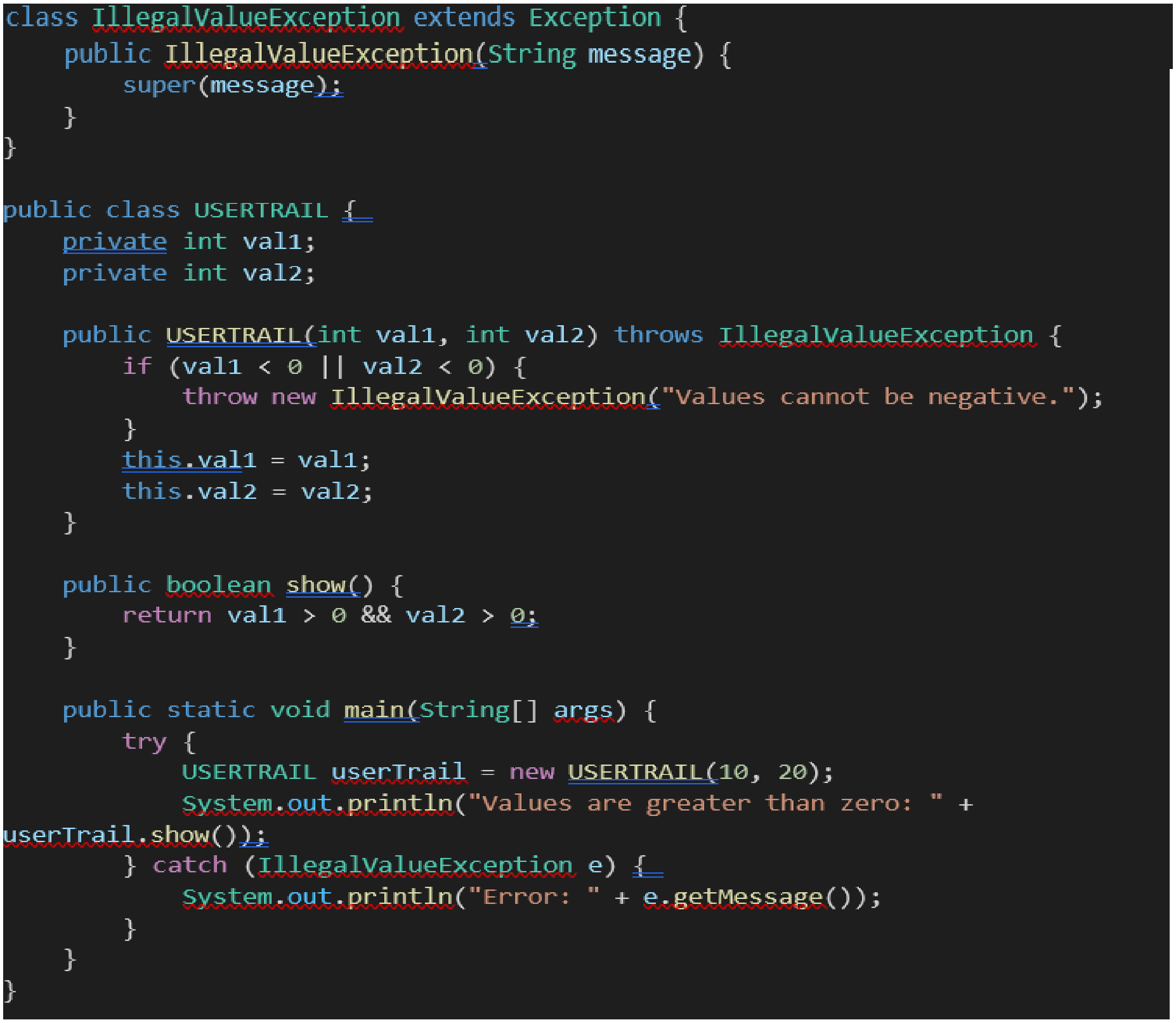
```
D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment3_Praticle>java Emp
Employee ID: 101
Employee Name: Rohit Shettigar
Designation: Manager
Basic Salary: Rs. 6000.0
HRA: Rs. 600.0

D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment3_Praticle>
```

Q.8 Create a class USERTRAIL with following specifications. val1, val2 type int Methods

boolean show () will check if val1 and val2 are greater or less than Zero

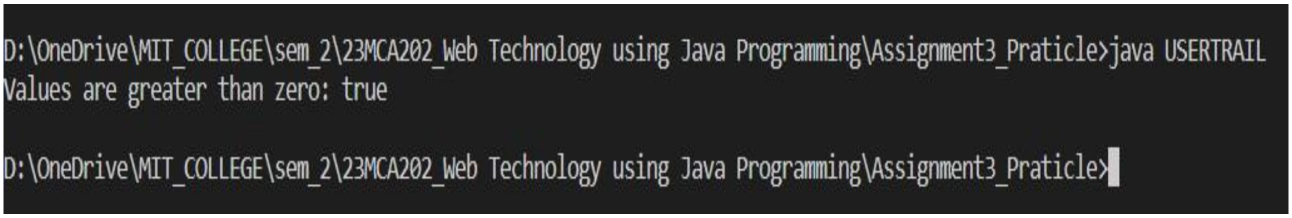
Have constructor which will val1, val2 and check whether if it is less than 0 then raise a custom Exception (name: Illegal value exception.)

CODE:



```
class IllegalValueException extends Exception {
    public IllegalValueException(String message) {
        super(message);
    }
}

public class USERTRAIL {
    private int val1;
    private int val2;

    public USERTRAIL(int val1, int val2) throws IllegalValueException {
        if (val1 < 0 || val2 < 0) {
            throw new IllegalValueException("Values cannot be negative.");
        }
        this.val1 = val1;
        this.val2 = val2;
    }

    public boolean show() {
        return val1 > 0 && val2 > 0;
    }

    public static void main(String[] args) {
        try {
            USERTRAIL userTrail = new USERTRAIL(10, 20);
            System.out.println("Values are greater than zero: " +
userTrail.show());
        } catch (IllegalValueException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

OUTPUT:



```
D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment3_Praticle>java USERTRAIL
Values are greater than zero: true

D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment3_Praticle>
```

ASSIGNMENTS NO. 4

Objective: At the end of the assignments, participants will be able to understand and write code for IO operations.

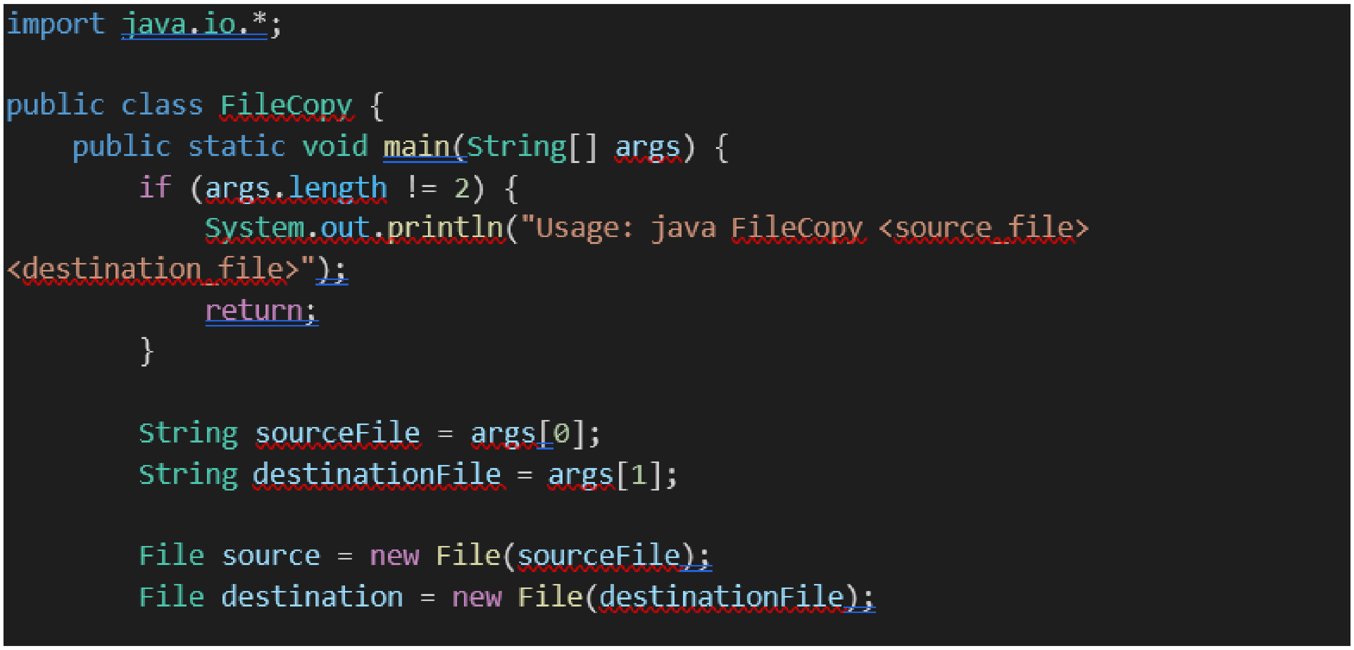
Concept: I/O Strems

Q.1 Write a program which take source file and destination file as input as command line arguments. It copies the source file contents to destination file. If source file does not exist, it should give appriate message to use. If destination file does not exist, it should be created. If it exists, program should ask that, "whether you want to overwrite? (Yes/No".

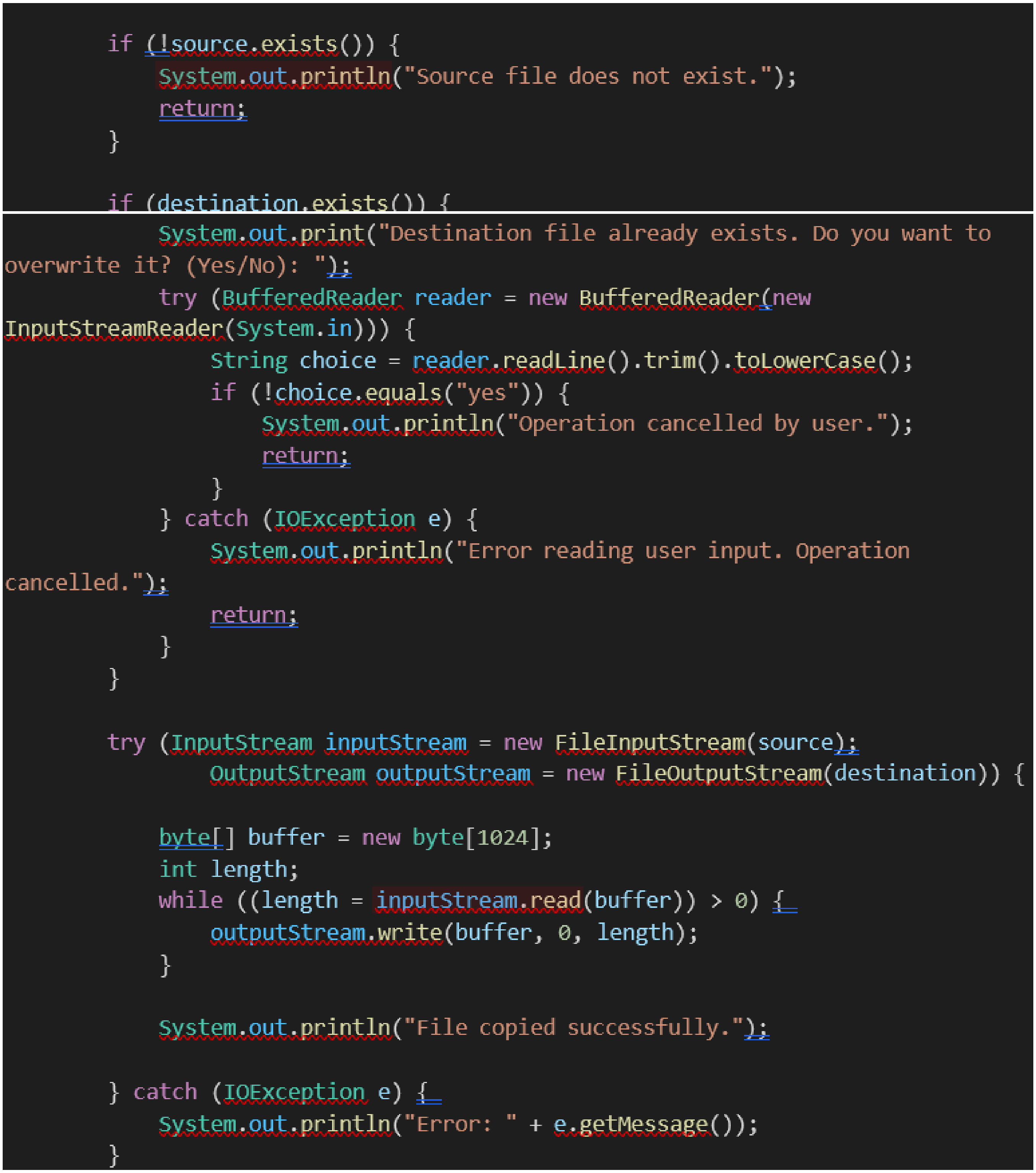
On the basis of user choice, appropriate action should be taken.

Note: Files may be any type of files like bitmap files, exe files, text files etc.

CODE:

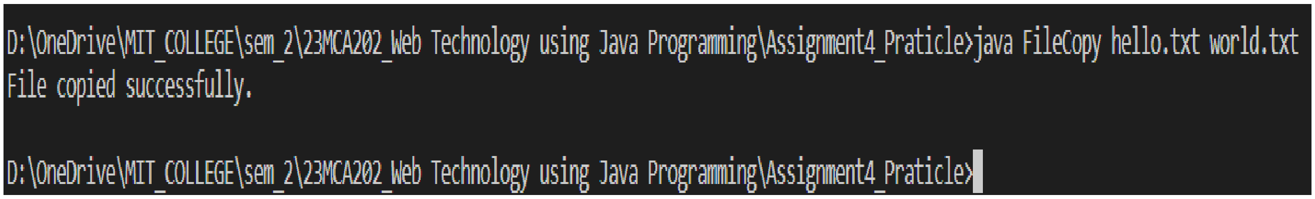


```
import java.io.*;

public class FileCopy {
    public static void main(String[] args) {
        if (args.length != 2) {
            System.out.println("Usage: java FileCopy <source_file>
<destination_file>");
            return;
        }

        String sourceFile = args[0];
        String destinationFile = args[1];

        File source = new File(sourceFile);
        File destination = new File(destinationFile);
```



```
        if (!source.exists()) {
            System.out.println("Source file does not exist.");
            return;
        }

        if (destination.exists()) {
            System.out.print("Destination file already exists. Do you want to
overwrite it? (Yes/No): ");
            try (BufferedReader reader = new BufferedReader(new
InputStreamReader(System.in))) {
                String choice = reader.readLine().trim().toLowerCase();
                if (!choice.equals("yes")) {
                    System.out.println("Operation cancelled by user.");
                    return;
                }
            } catch (IOException e) {
                System.out.println("Error reading user input. Operation
cancelled.");
                return;
            }
        }

        try (InputStream inputStream = new FileInputStream(source);
                OutputStream outputStream = new FileOutputStream(destination)) {

            byte[] buffer = new byte[1024];
            int length;
            while ((length = inputStream.read(buffer)) > 0) {
                outputStream.write(buffer, 0, length);
            }

            System.out.println("File copied successfully.");

        } catch (IOException e) {
            System.out.println("Error: " + e.getMessage());
        }
```

OUTPUT:



```
D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment4_Praticle>java FileCopy hello.txt world.txt
File copied successfully.

D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment4_Praticle>
```

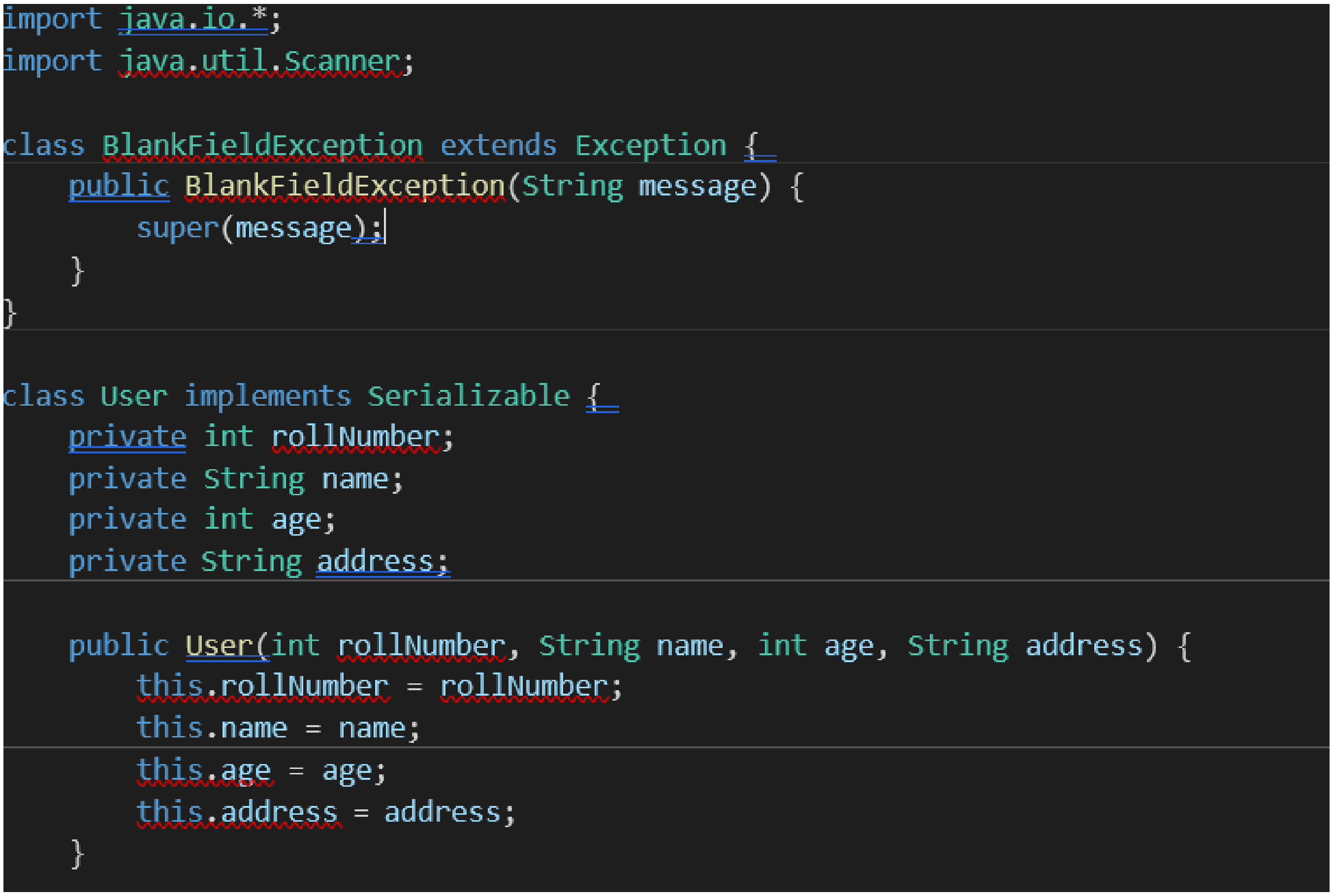
Q.2 **Write a stream based program which will accept Roll Number, Name, Age and Address from user Age and Roll-no should be numeric. Handle with built-in exception.**
None of the field should be blank. Handle with custom

exception, Ask user ,whether to write the data in the file

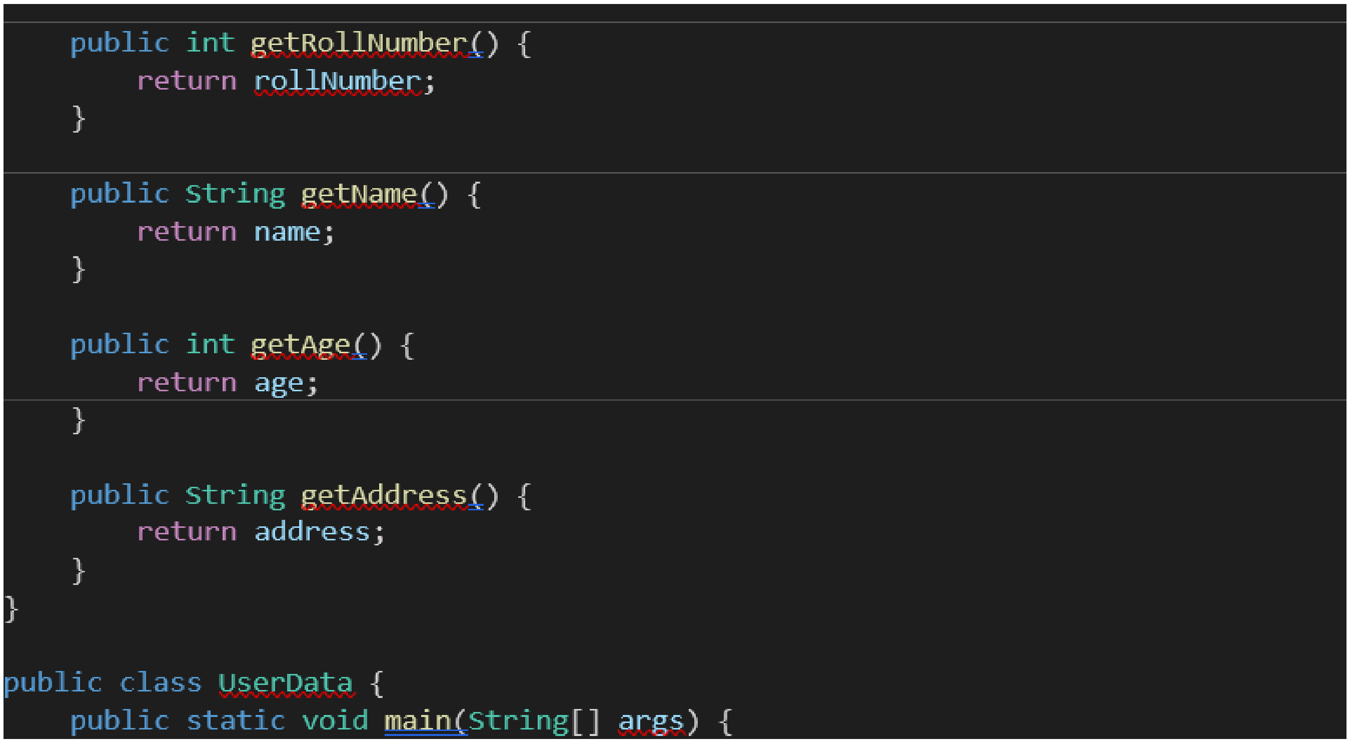
If answer is yes then data is saved into a file as an object(User can write many records in the file), otherwise terminate the current program

Write another program to display all the records saved into the file.

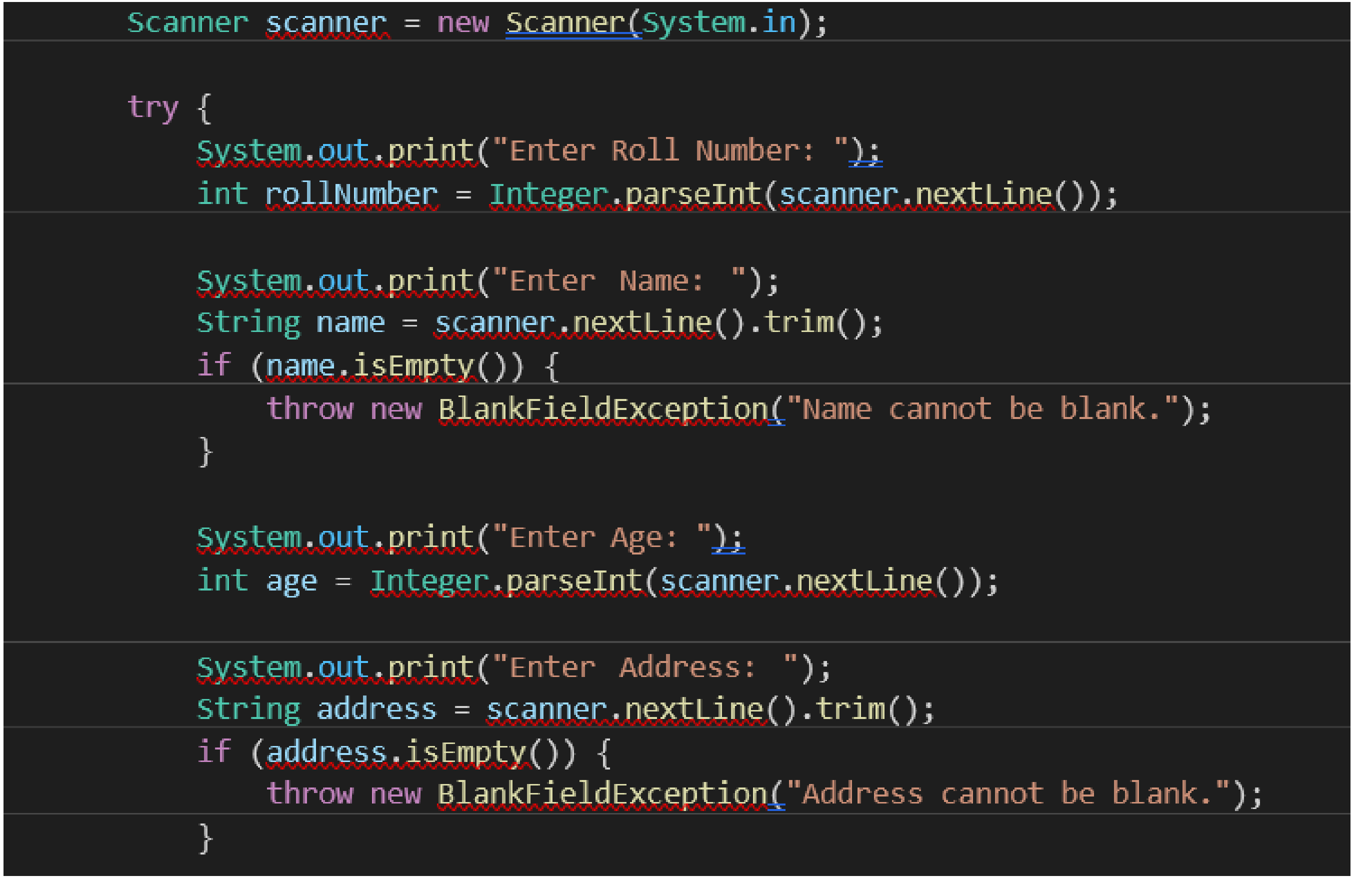
CODE:



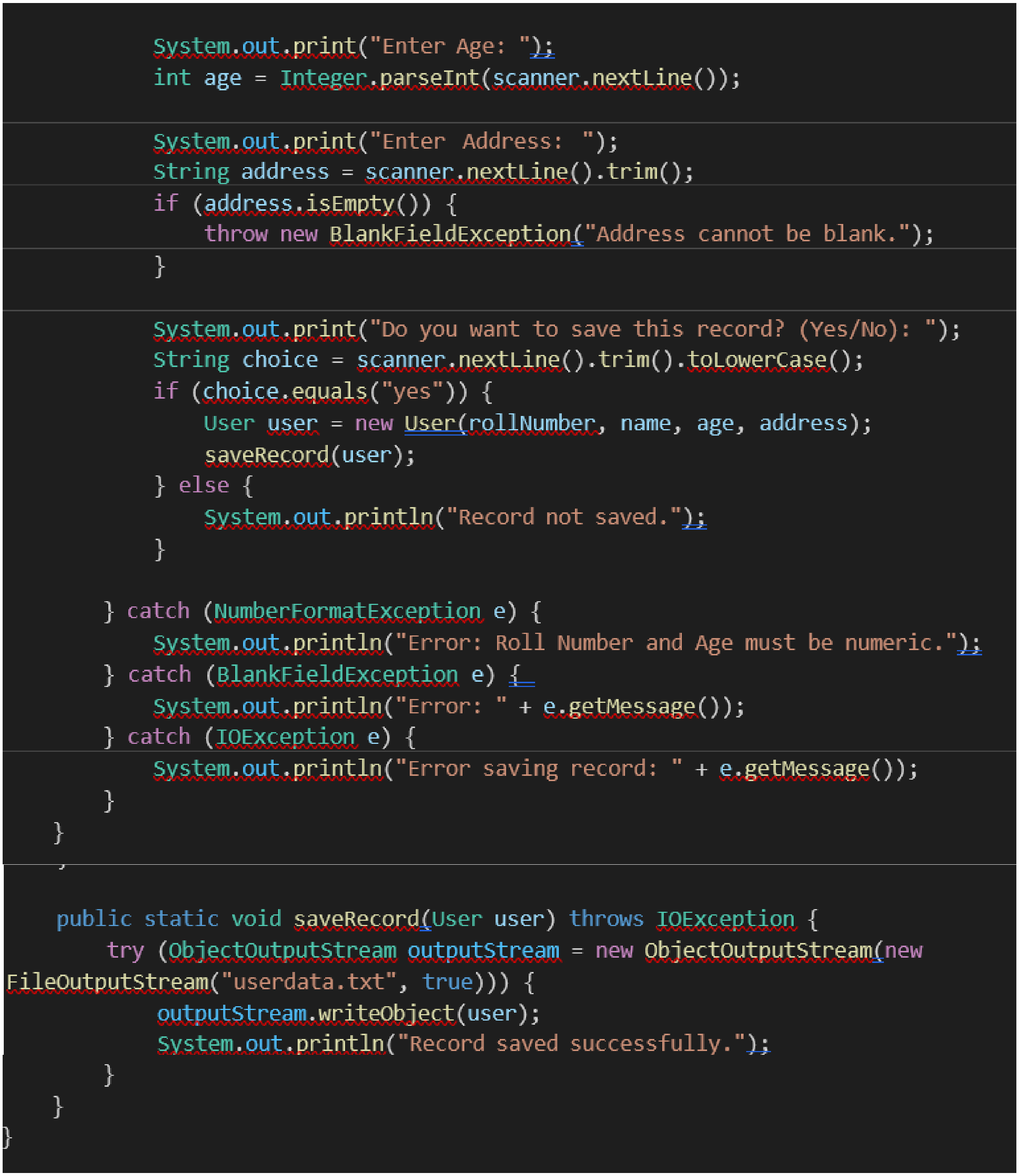
```
import java.io.*;
import java.util.Scanner;

class BlankFieldException extends Exception {
    public BlankFieldException(String message) {
        super(message);
    }
}

class User implements Serializable {
    private int rollNumber;
    private String name;
    private int age;
    private String address;

    public User(int rollNumber, String name, int age, String address) {
        this.rollNumber = rollNumber;
        this.name = name;
        this.age = age;
        this.address = address;
    }
```



```java
    public int getRollNumber() {
        return rollNumber;
    }

    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
    }

    public String getAddress() {
        return address;
    }
}

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

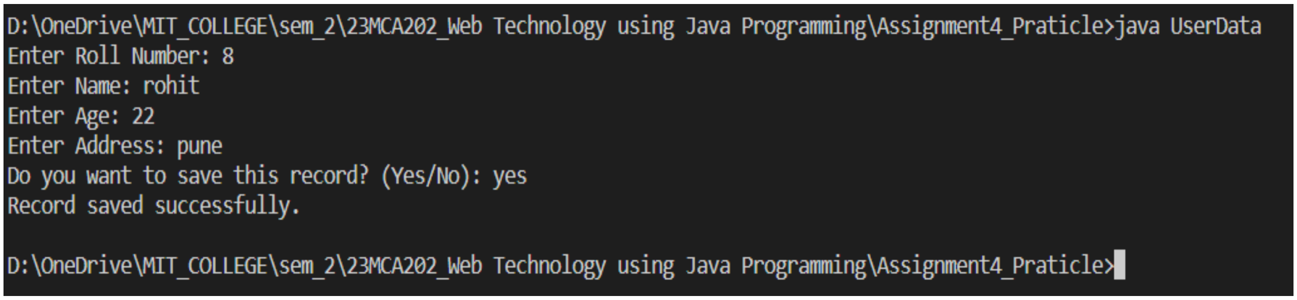


```java
        Scanner scanner = new Scanner(System.in);

        try {
            System.out.print("Enter Roll Number: ");
            int rollNumber = Integer.parseInt(scanner.nextLine());

            System.out.print("Enter Name: ");
            String name = scanner.nextLine().trim();
            if (name.isEmpty()) {
                throw new BlankFieldException("Name cannot be blank.");
            }

            System.out.print("Enter Age: ");
            int age = Integer.parseInt(scanner.nextLine());

            System.out.print("Enter Address: ");
            String address = scanner.nextLine().trim();
            if (address.isEmpty()) {
                throw new BlankFieldException("Address cannot be blank.");
            }
```



```
            System.out.print("Enter Age: ");
            int age = Integer.parseInt(scanner.nextLine());

            System.out.print("Enter  Address:  ");
            String address = scanner.nextLine().trim();
            if (address.isEmpty()) {
                throw new BlankFieldException("Address cannot be blank.");
            }

            System.out.print("Do you want to save this record? (Yes/No): ");
            String choice = scanner.nextLine().trim().toLowerCase();
            if (choice.equals("yes")) {
                User user = new User(rollNumber, name, age, address);
                saveRecord(user);
            } else {
                System.out.println("Record not saved.");
            }

        } catch (NumberFormatException e) {
            System.out.println("Error: Roll Number and Age must be numeric.");
        } catch (BlankFieldException e) {
            System.out.println("Error: " + e.getMessage());
        } catch (IOException e) {
            System.out.println("Error saving record: " + e.getMessage());
        }
    }

    public static void saveRecord(User user) throws IOException {
        try (ObjectOutputStream outputStream = new ObjectOutputStream(new
FileOutputStream("userdata.txt", true))) {
            outputStream.writeObject(user);
            System.out.println("Record saved successfully.");
        }
    }
}
```

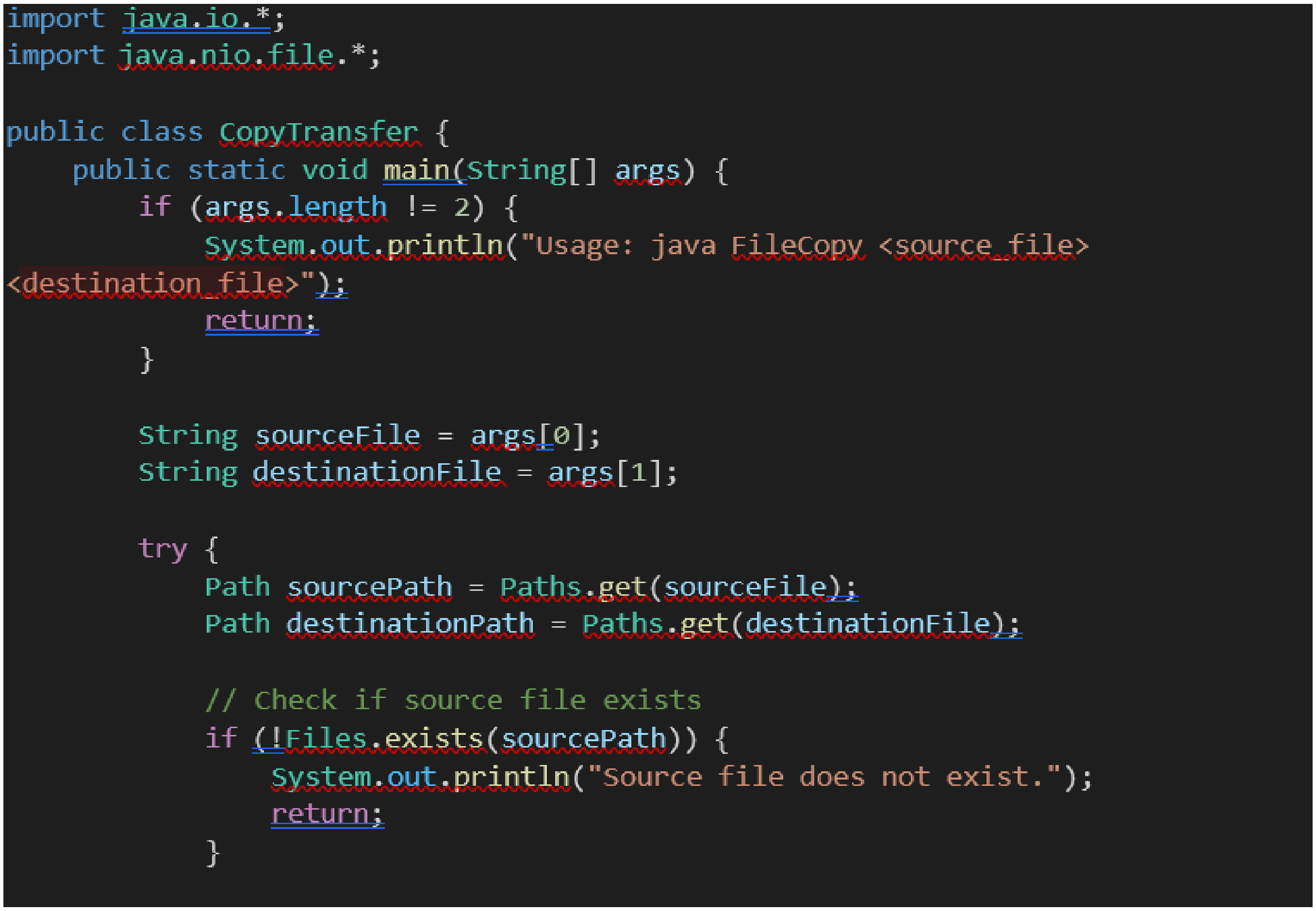
OUTPUT:



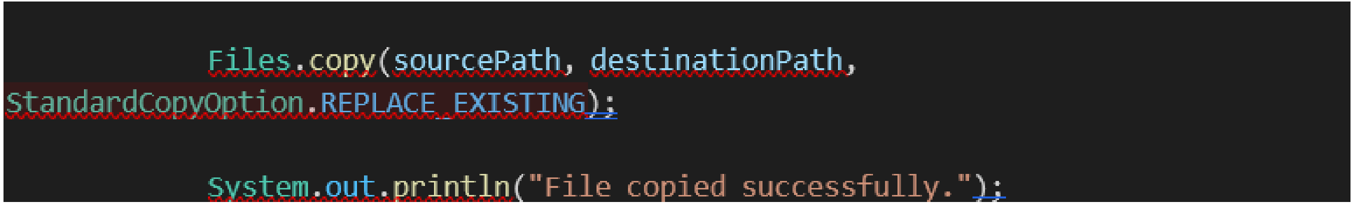
```
D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment4_Praticle>java UserData
Enter Roll Number: 8
Enter Name: rohit
Enter Age: 22
Enter Address: pune
Do you want to save this record? (Yes/No): yes
Record saved successfully.

D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment4_Praticle>
```

Q.3 Write a program using java file system to copy the contents of one file into another.

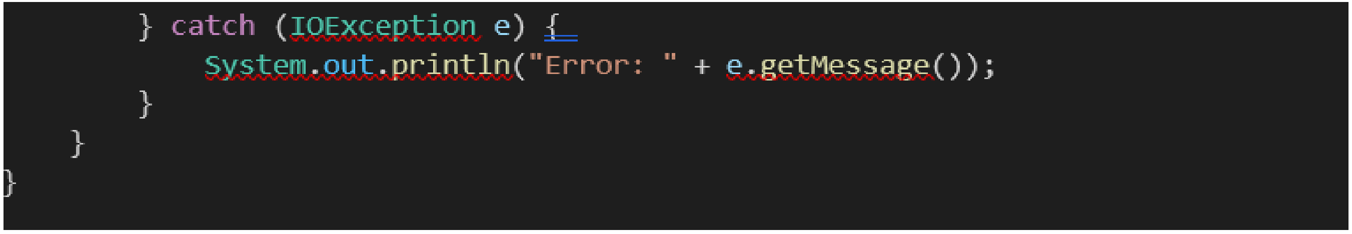
CODE:



```
import java.io.*;
import java.nio.file.*;

public class CopyTransfer {
    public static void main(String[] args) {
        if (args.length != 2) {
            System.out.println("Usage: java FileCopy <source_file>
<destination_file>");
            return;
        }

        String sourceFile = args[0];
        String destinationFile = args[1];

        try {
            Path sourcePath = Paths.get(sourceFile);
            Path destinationPath = Paths.get(destinationFile);

            // Check if source file exists
            if (!Files.exists(sourcePath)) {
                System.out.println("Source file does not exist.");
                return;
            }
```

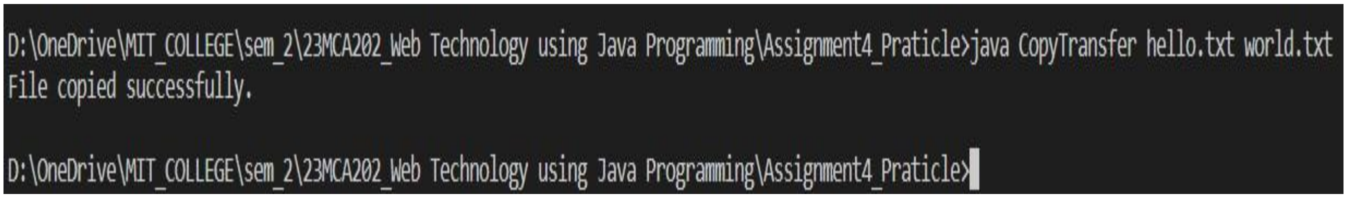


```
            Files.copy(sourcePath, destinationPath,
StandardCopyOption.REPLACE_EXISTING);

            System.out.println("File copied successfully.");
```



```
        } catch (IOException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

OUTPUT:



```
D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment4_Praticle>java CopyTransfer hello.txt world.txt
File copied successfully.

D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment4_Praticle>
```

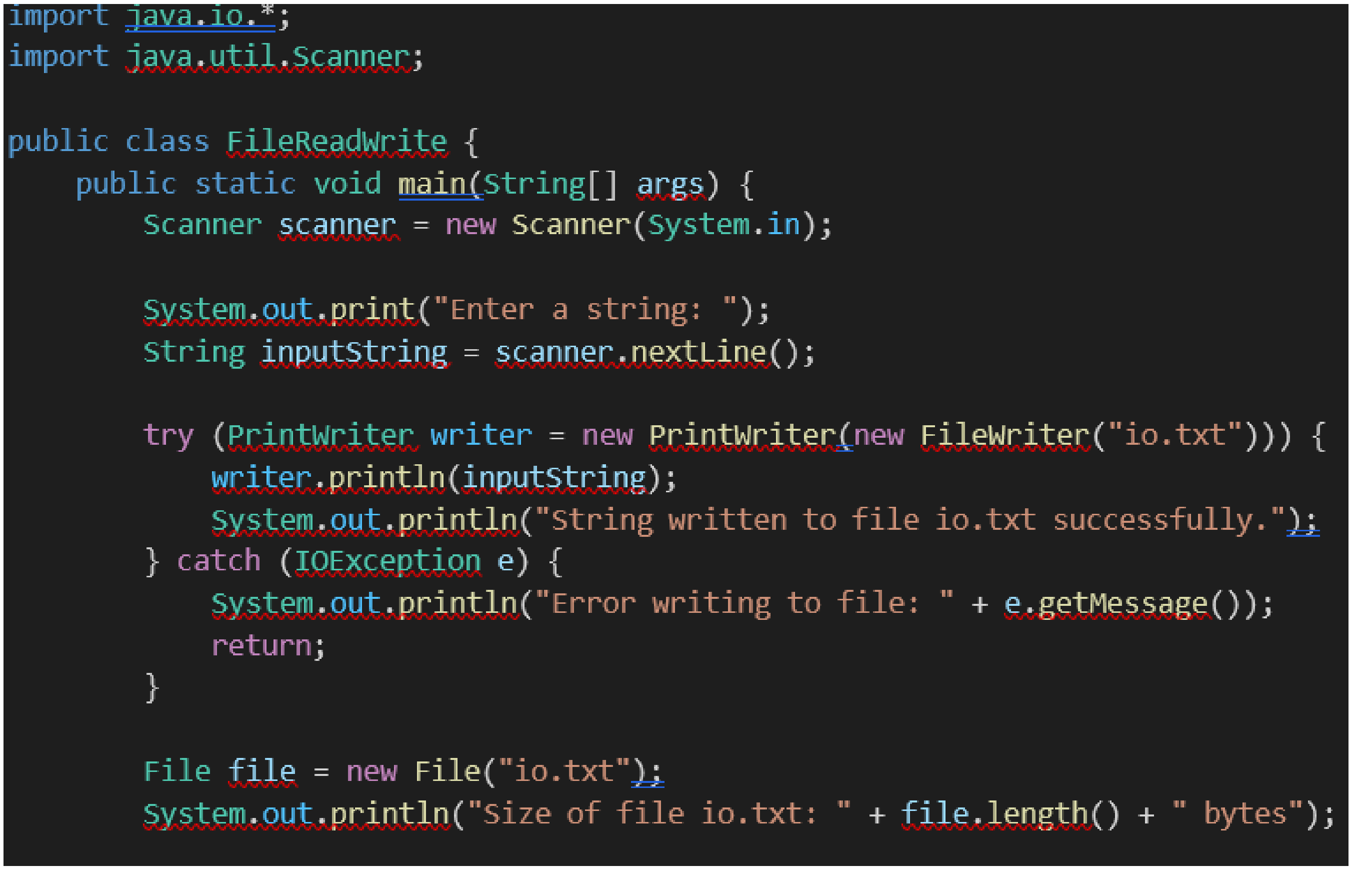
Q.4 Write a program which will accept an input String from user

Write the input in the file io.txt

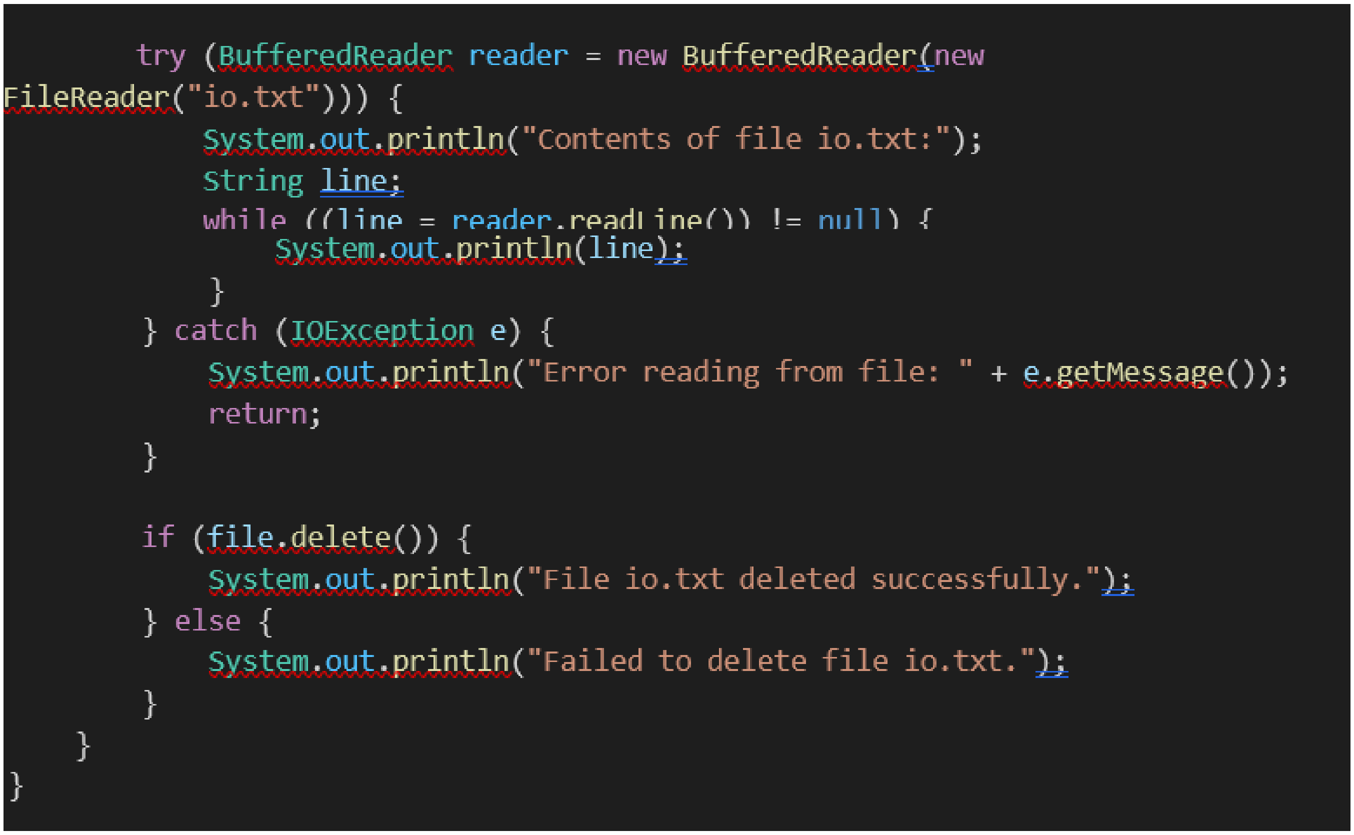
Show size of the file

Read contets from the file and display them on console Delete io.txt file using File class.

CODE:

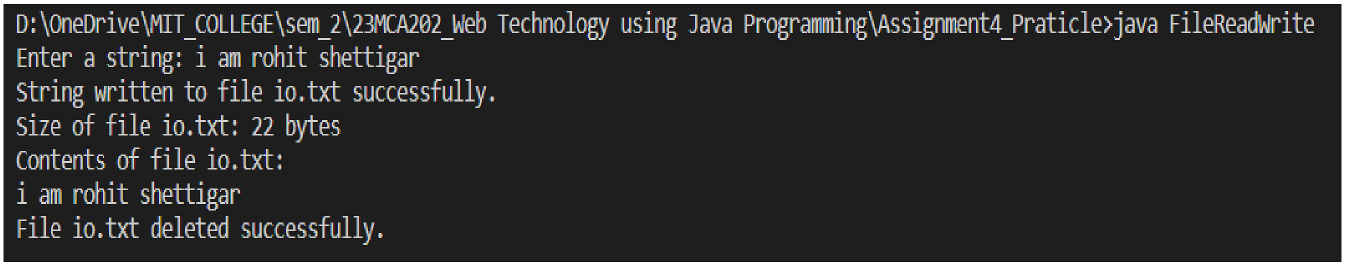


```
import java.io.*;
import java.util.Scanner;

public class FileReadWrite {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a string: ");
        String inputString = scanner.nextLine();

        try (PrintWriter writer = new PrintWriter(new FileWriter("io.txt"))) {
            writer.println(inputString);
            System.out.println("String written to file io.txt successfully.");
        } catch (IOException e) {
            System.out.println("Error writing to file: " + e.getMessage());
            return;
        }

        File file = new File("io.txt");
        System.out.println("Size of file io.txt: " + file.length() + " bytes");
```



```
        try (BufferedReader reader = new BufferedReader(new 
FileReader("io.txt"))) {
            System.out.println("Contents of file io.txt:");
            String line;
            while ((line = reader.readLine()) != null) {
                System.out.println(line);
            }
        } catch (IOException e) {
            System.out.println("Error reading from file: " + e.getMessage());
            return;
        }

        if (file.delete()) {
            System.out.println("File io.txt deleted successfully.");
        } else {
            System.out.println("Failed to delete file io.txt.");
        }
    }
}
```

OUTPUT:



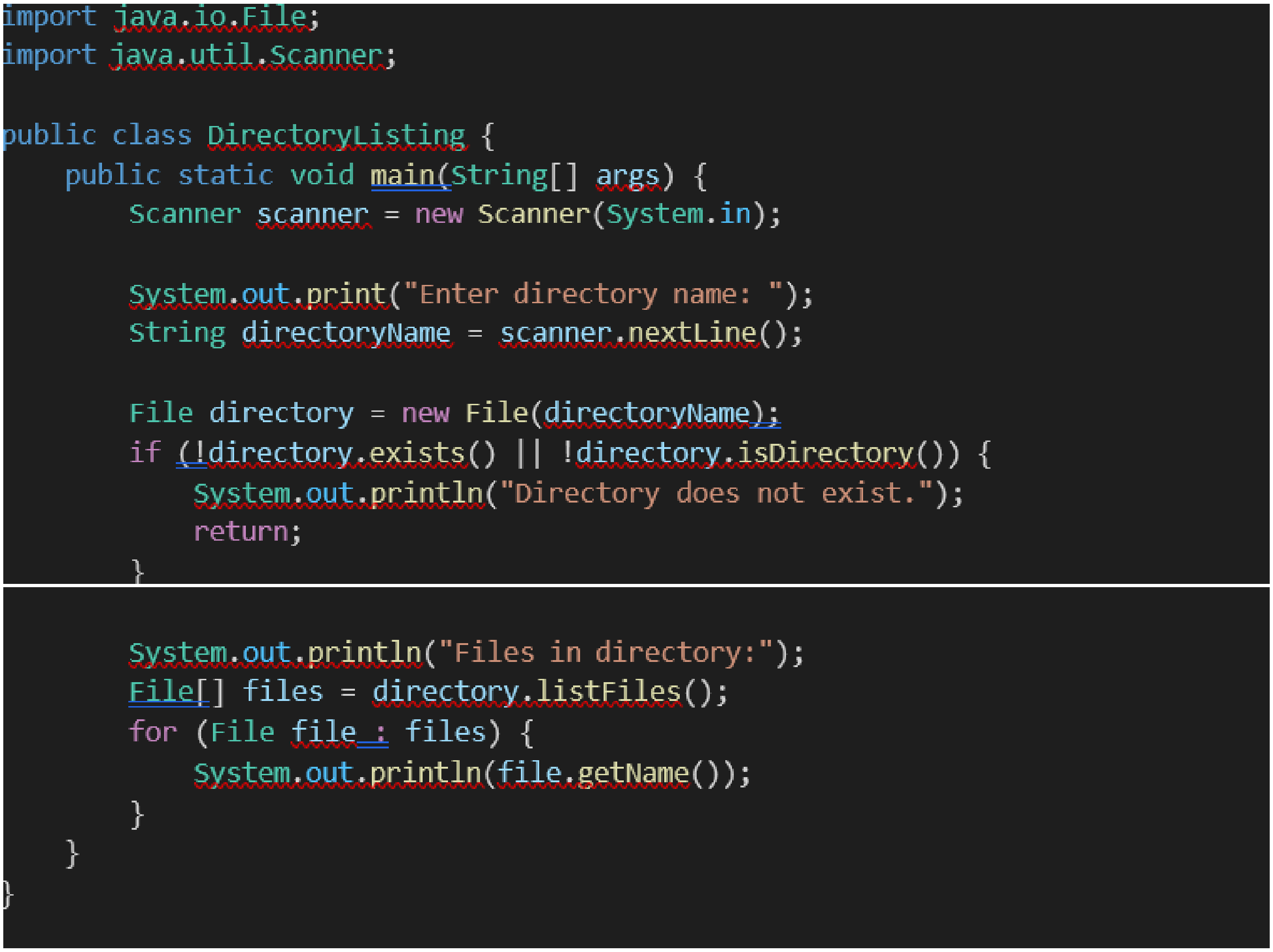
```
D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment4_Praticle>java FileReadWrite
Enter a string: i am rohit shettigar
String written to file io.txt successfully.
Size of file io.txt: 22 bytes
Contents of file io.txt:
i am rohit shettigar
File io.txt deleted successfully.
```

Q.5 **Accept a directory name in form of String from user using proper IO stream. Store it in a variable. (This is Self Study Assignment. Refer Java API documentation.)**

Search if it exists in your system.

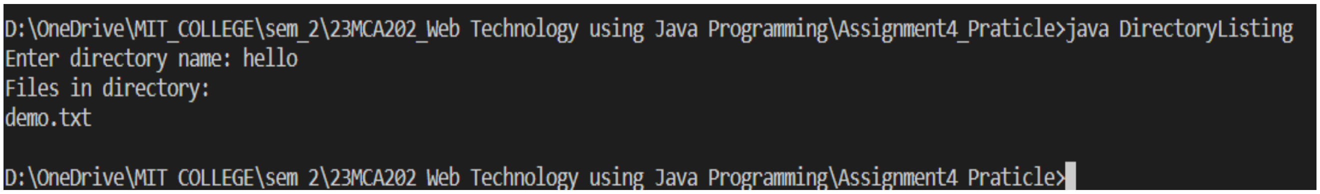
If it exists then show all the files present in the directory otherwise print the message that directory Does not Exists.

CODE:



```
import java.io.File;
import java.util.Scanner;

public class DirectoryListing {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter directory name: ");
        String directoryName = scanner.nextLine();

        File directory = new File(directoryName);
        if (!directory.exists() || !directory.isDirectory()) {
            System.out.println("Directory does not exist.");
            return;
        }

        System.out.println("Files in directory:");
        File[] files = directory.listFiles();
        for (File file : files) {
            System.out.println(file.getName());
        }
    }
}
```

OUTPUT:



```
D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment4_Praticle>java DirectoryListing
Enter directory name: hello
Files in directory:
demo.txt

D:\OneDrive\MIT COLLEGE\sem 2\23MCA202 Web Technology using Java Programming\Assignment4 Praticle>
```

Q.6 Create a class “CDR” with the following members:

A_Number
B_Number
duration
calculated
Charge

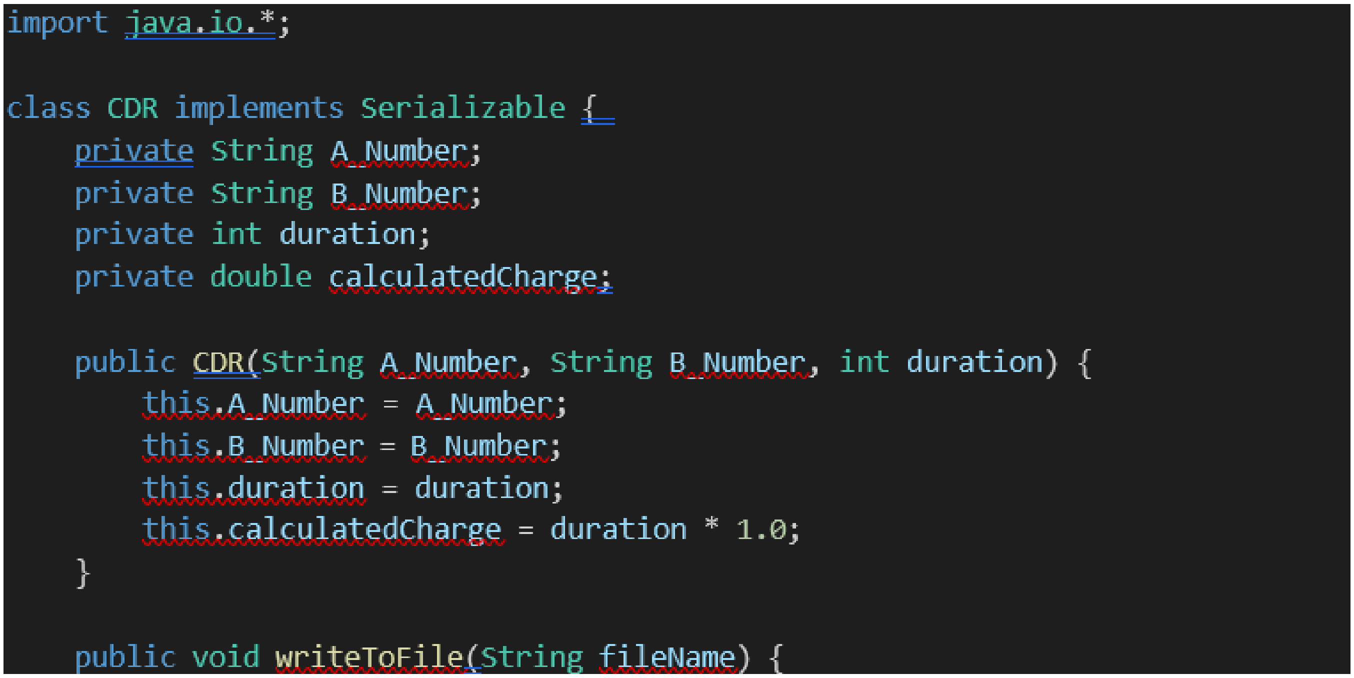
Write a program which will accept the A_Number, B_Number and duration of Call from user(Duration is in minutes). Rate that call using 1 Rupee per min rate and store the calculated charge in

“calculatedCharge.txt” file. Write this

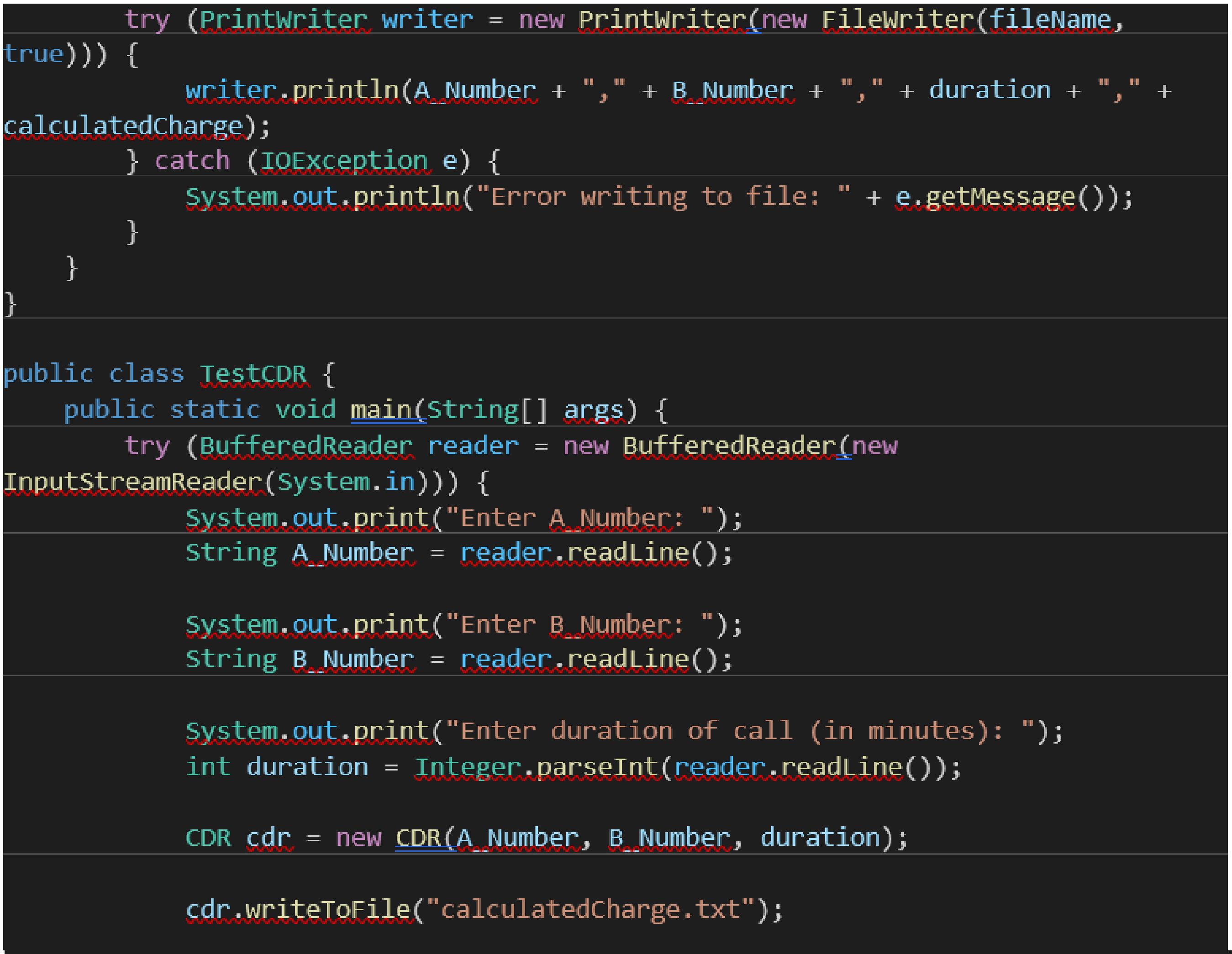
object to “rated_cdr.txt”. Note : CDR means

Call Details Record.

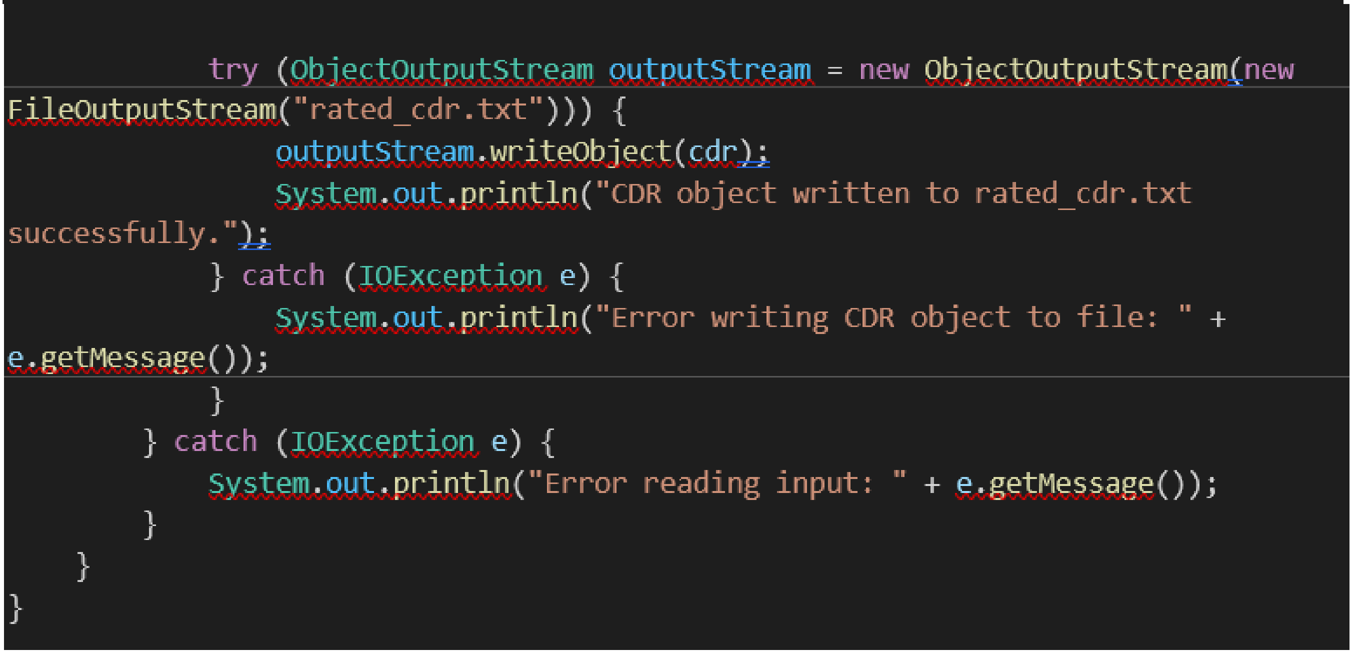
CODE:



```
import java.io.*;

class CDR implements Serializable {
    private String A_Number;
    private String B_Number;
    private int duration;
    private double calculatedCharge;

    public CDR(String A_Number, String B_Number, int duration) {
        this.A_Number = A_Number;
        this.B_Number = B_Number;
        this.duration = duration;
        this.calculatedCharge = duration * 1.0;
    }

    public void writeToFile(String fileName) {
```

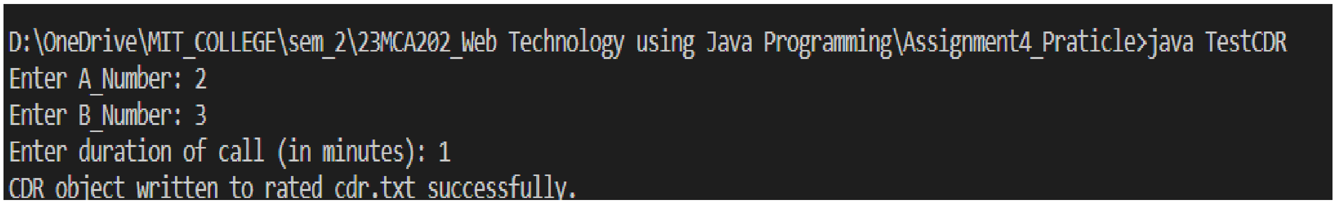


```
        try (PrintWriter writer = new PrintWriter(new FileWriter(fileName,
true))) {
            writer.println(A_Number + "," + B_Number + "," + duration + "," +
calculatedCharge);
        } catch (IOException e) {
            System.out.println("Error writing to file: " + e.getMessage());
        }
    }
}

public class TestCDR {
    public static void main(String[] args) {
        try (BufferedReader reader = new BufferedReader(new
InputStreamReader(System.in))) {
            System.out.print("Enter A_Number: ");
            String A_Number = reader.readLine();

            System.out.print("Enter B_Number: ");
            String B_Number = reader.readLine();

            System.out.print("Enter duration of call (in minutes): ");
            int duration = Integer.parseInt(reader.readLine());

            CDR cdr = new CDR(A_Number, B_Number, duration);

            cdr.writeToFile("calculatedCharge.txt");
```



```
            try (ObjectOutputStream outputStream = new ObjectOutputStream(new
FileOutputStream("rated_cdr.txt"))) {
                outputStream.writeObject(cdr);
                System.out.println("CDR object written to rated_cdr.txt
successfully.");
            } catch (IOException e) {
                System.out.println("Error writing CDR object to file: " +
e.getMessage());
            }
        } catch (IOException e) {
            System.out.println("Error reading input: " + e.getMessage());
        }
    }
}
```

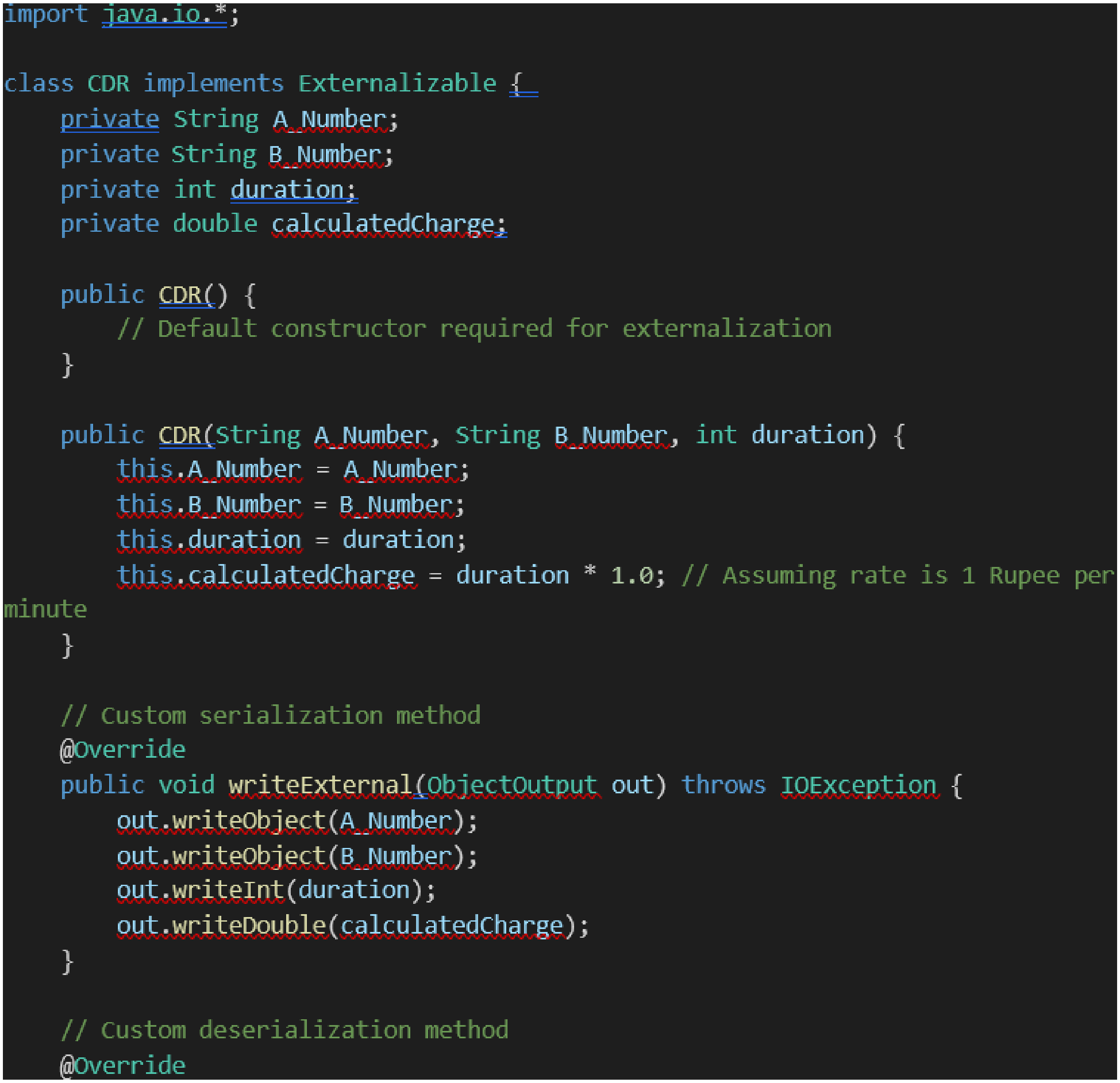
OUTPUT:



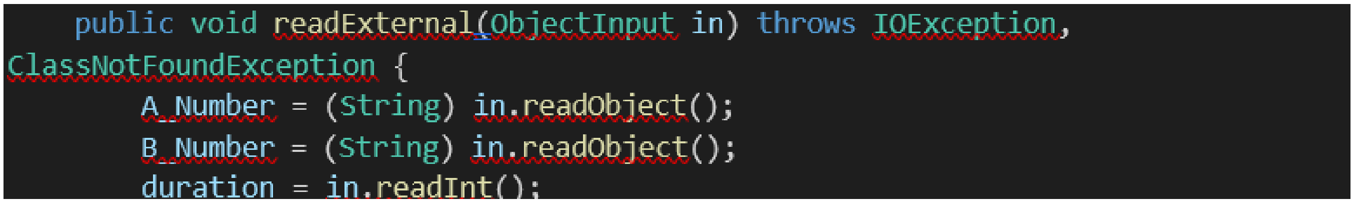
```
D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment4_Praticle>java TestCDR
Enter A_Number: 2
Enter B_Number: 3
Enter duration of call (in minutes): 1
CDR object written to rated_cdr.txt successfully.
```

Q.7 **Identify that, what functionality need to be added in above assignment , so that it will require object externalization. Take approval for this functionality from the faculty. Then implement this functionality using Object Externalization.**

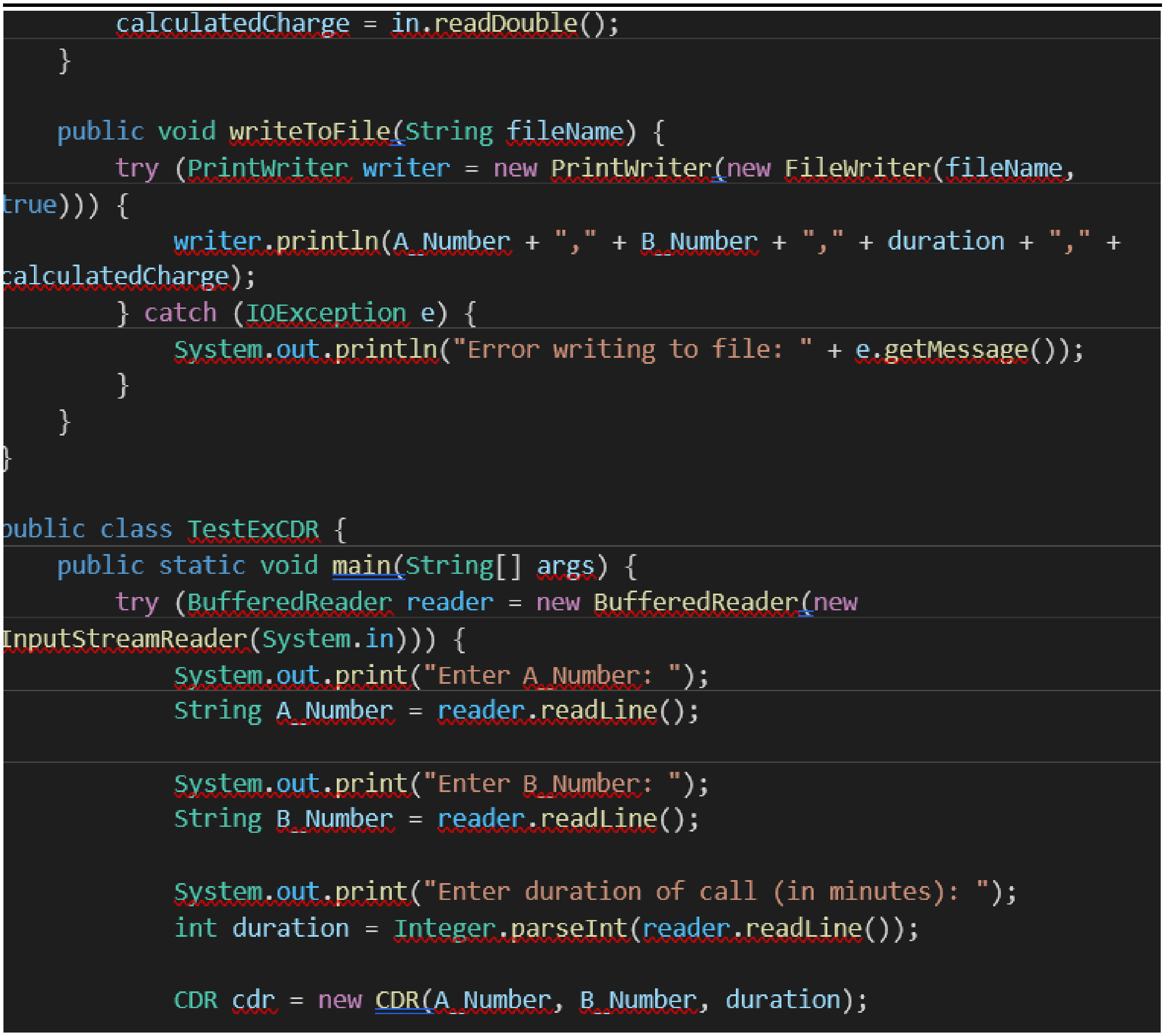
CODE:



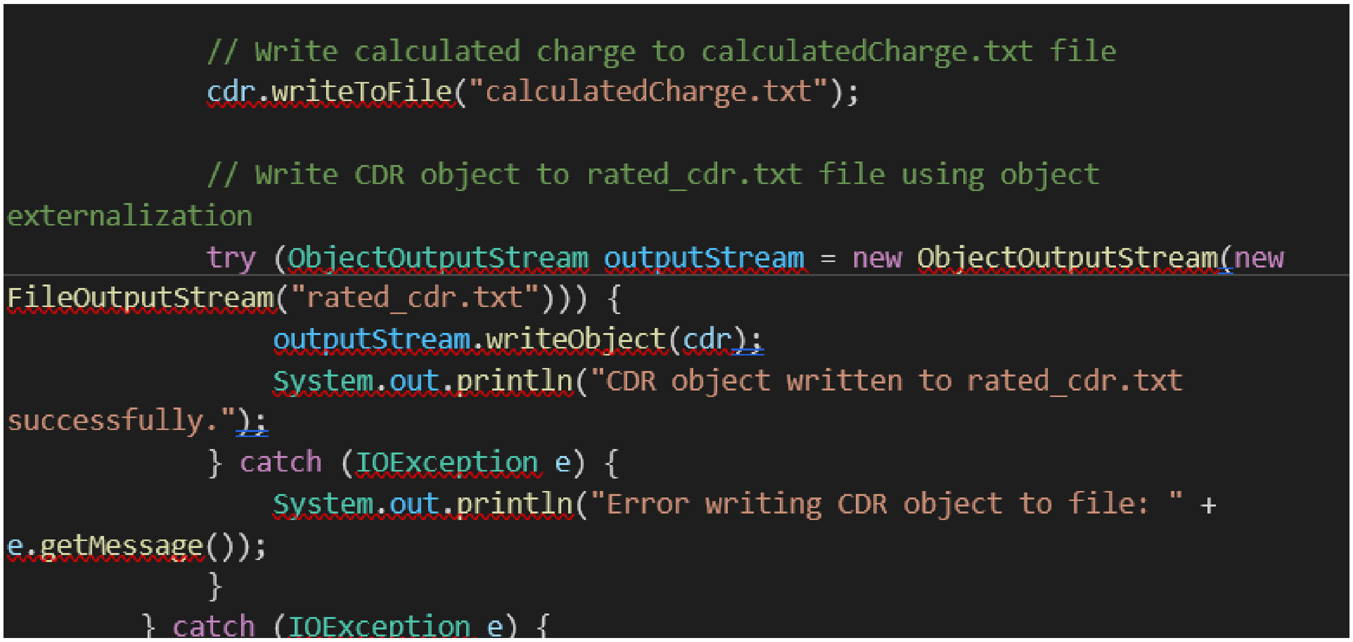
```
import java.io.*;

class CDR implements Externalizable {
    private String A_Number;
    private String B_Number;
    private int duration;
    private double calculatedCharge;

    public CDR() {
        // Default constructor required for externalization
    }

    public CDR(String A_Number, String B_Number, int duration) {
        this.A_Number = A_Number;
        this.B_Number = B_Number;
        this.duration = duration;
        this.calculatedCharge = duration * 1.0; // Assuming rate is 1 Rupee per
minute
    }

    // Custom serialization method
    @Override
    public void writeExternal(ObjectOutput out) throws IOException {
        out.writeObject(A_Number);
        out.writeObject(B_Number);
        out.writeInt(duration);
        out.writeDouble(calculatedCharge);
    }

    // Custom deserialization method
    @Override
```



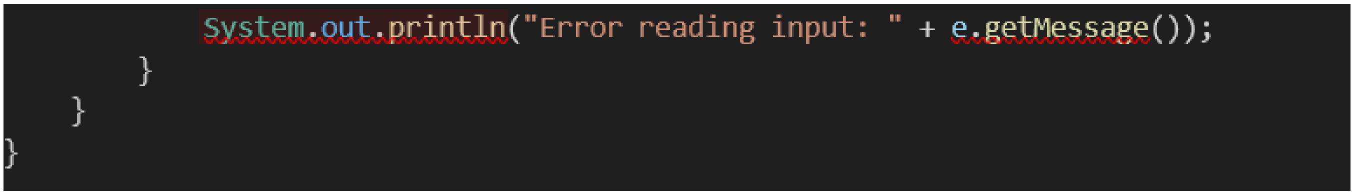
```
    public void readExternal(ObjectInput in) throws IOException,
ClassNotFoundException {
        A_Number = (String) in.readObject();
        B_Number = (String) in.readObject();
        duration = in.readInt();
```



```
        calculatedCharge = in.readDouble();
    }

    public void writeToFile(String fileName) {
        try (PrintWriter writer = new PrintWriter(new FileWriter(fileName,
true))) {
            writer.println(A_Number + "," + B_Number + "," + duration + "," +
calculatedCharge);
        } catch (IOException e) {
            System.out.println("Error writing to file: " + e.getMessage());
        }
    }
}

public class TestExCDR {
    public static void main(String[] args) {
        try (BufferedReader reader = new BufferedReader(new
InputStreamReader(System.in))) {
            System.out.print("Enter A_Number: ");
            String A_Number = reader.readLine();

            System.out.print("Enter B_Number: ");
            String B_Number = reader.readLine();

            System.out.print("Enter duration of call (in minutes): ");
            int duration = Integer.parseInt(reader.readLine());

            CDR cdr = new CDR(A_Number, B_Number, duration);
```

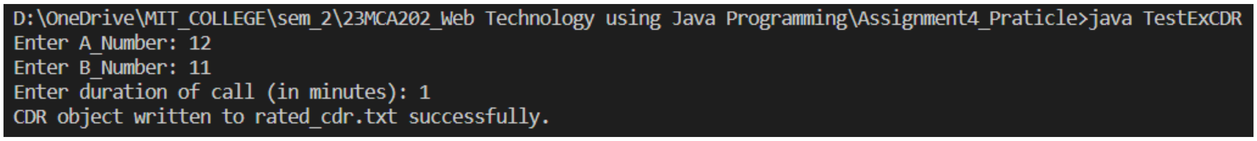


```
            // Write calculated charge to calculatedCharge.txt file
            cdr.writeToFile("calculatedCharge.txt");

            // Write CDR object to rated_cdr.txt file using object
externalization
            try (ObjectOutputStream outputStream = new ObjectOutputStream(new
FileOutputStream("rated_cdr.txt"))) {
                outputStream.writeObject(cdr);
                System.out.println("CDR object written to rated_cdr.txt
successfully.");
            } catch (IOException e) {
                System.out.println("Error writing CDR object to file: " +
e.getMessage());
            }
        } catch (IOException e) {
```



```
            System.out.println("Error reading input: " + e.getMessage());
        }
    }
}
```

OUTPUT:



```
D:\OneDrive\MIT_COLLEGE\sem_2\23MCA202_Web Technology using Java Programming\Assignment4_Praticle>java TestExCDR
Enter A_Number: 12
Enter B_Number: 11
Enter duration of call (in minutes): 1
CDR object written to rated_cdr.txt successfully.
```

Prepared By

Prof. Sheela Hundekari

Sign of Practical Subject Teacher: