VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming (23CS3PCOOJ) *Submitted by*

DHARUNYA BALAVELAVAN (1BM23CS090)

in partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
September-2024 to Jan-2025
B. M. S. College of Engineering,
Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled “Object oriented java programming(23CS3PCOOJ)” carried out by **DHARUNYA BALAVELAVAN (1BM23CS090),** who is a bonafide student of **B. M. S. College of Engineering.** It is in partial fulfilment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2024. The Lab report has been approved as it satisfies the academic requirements in respect of an Object oriented java programming(23CS3PCOOJ) work prescribed for the said degree.

Mrs.Swathi Sridharan
Associate Professor
Department of CSE, BMSCE

Dr. Joythi S Nayak
Professor HOD
Department of CSE, BMSCE

`

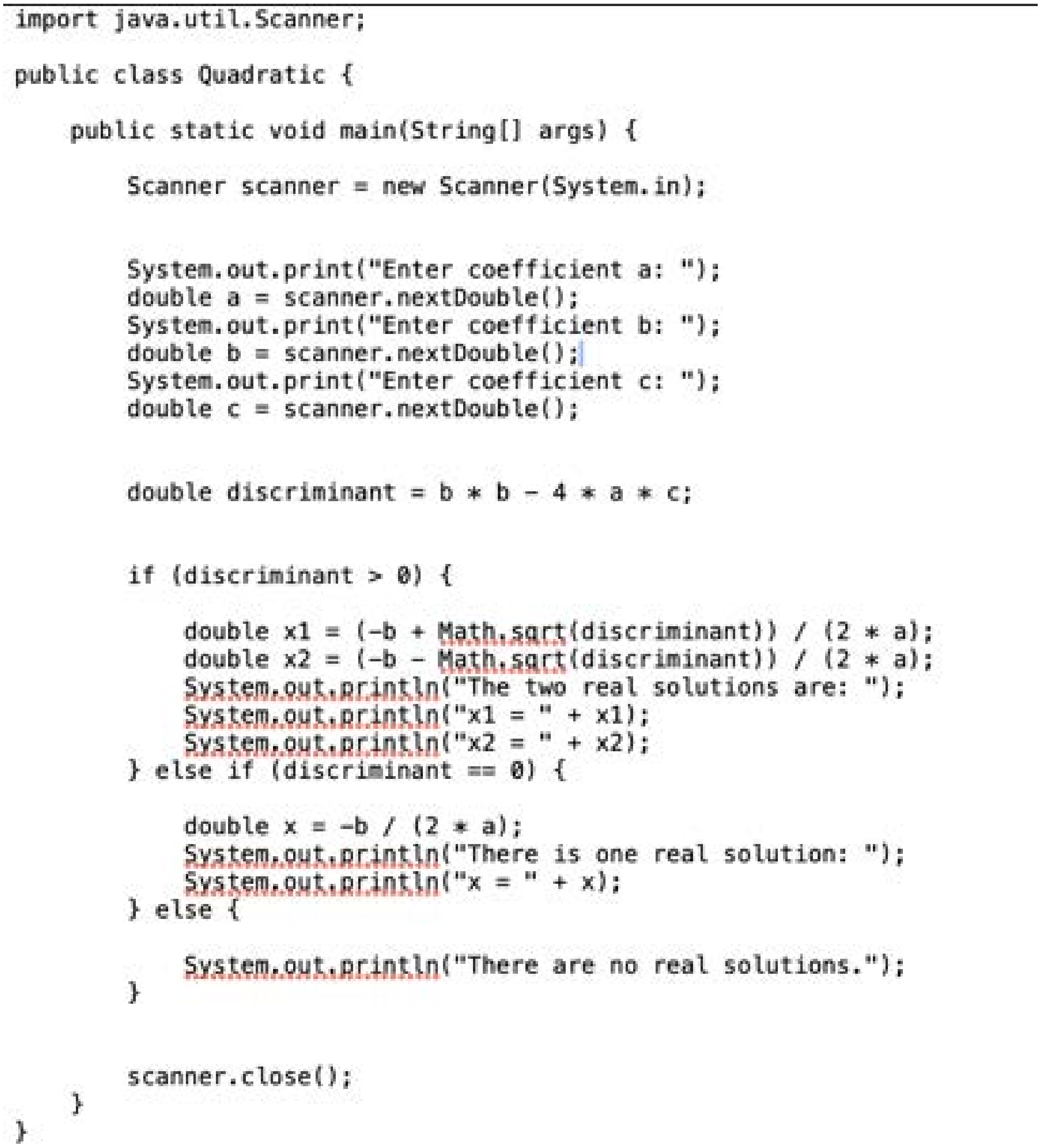
INDEX

Github Link

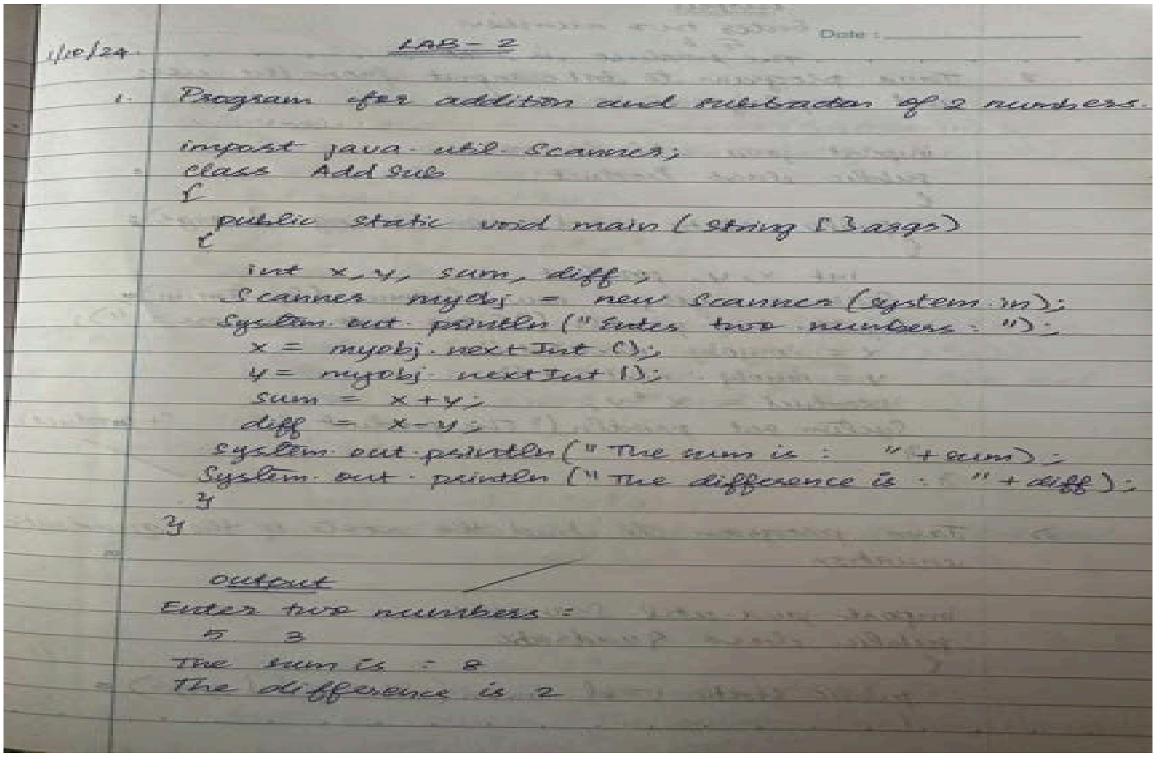
https://github.com/Dharunya21/OOJ

LAB PROGRAMS

1.Develop a Java program that prints all real solutions to the quadratic equation ax2 +bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminant b2 -4ac is negative, display a message stating that there are no real solutions.



```
import java.util.Scanner;

public class Quadratic {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);


        System.out.print("Enter coefficient a: ");
        double a = scanner.nextDouble();
        System.out.print("Enter coefficient b: ");
        double b = scanner.nextDouble();
        System.out.print("Enter coefficient c: ");
        double c = scanner.nextDouble();


        double discriminant = b * b - 4 * a * c;


        if (discriminant > 0) {

            double x1 = (-b + Math.sqrt(discriminant)) / (2 * a);
            double x2 = (-b - Math.sqrt(discriminant)) / (2 * a);
            System.out.println("The two real solutions are: ");
            System.out.println("x1 = " + x1);
            System.out.println("x2 = " + x2);
        } else if (discriminant == 0) {

            double x = -b / (2 * a);
            System.out.println("There is one real solution: ");
            System.out.println("x = " + x);
        } else {

            System.out.println("There are no real solutions.");
        }


        scanner.close();
    }
}
```



1/10/24

LAB - 2

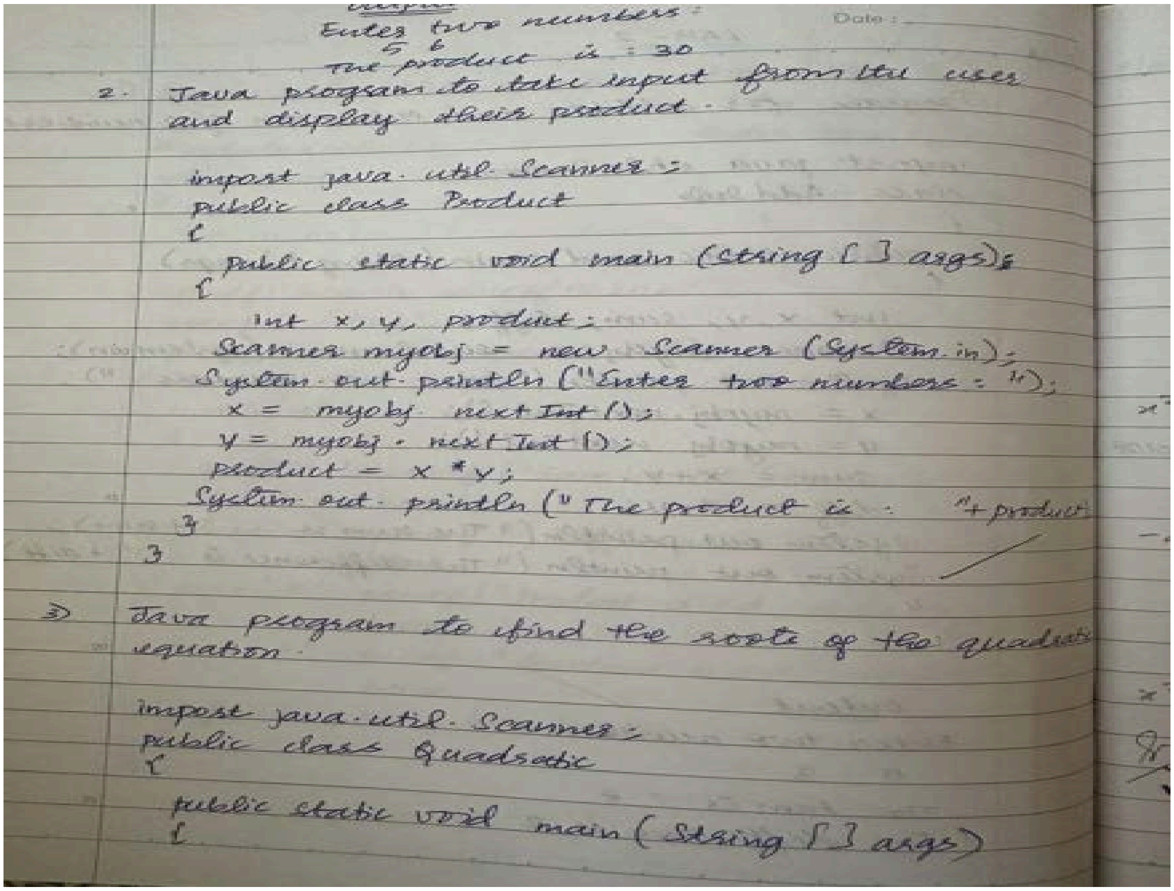
Date :

1. Program for addition and subtraction of 2 numbers.

```
import java.util.Scanner;
class AddSub
{
 public static void main (String [] args)
 {
   int x, y, sum, diff;
  Scanner myobj = new Scanner(system.in);
  System.out.println("Enter two numbers : ");
   x = myobj.nextInt ();
   y = myobj.nextInt ();
   sum = x+y;
   diff = x-y;
  system.out.println(" The sum is :   " + sum);
  System.out.println (" The difference is :   " + diff);
 }
}
```

output

```
Enter two numbers :
 5  3
The sum is : 8
The difference is 2
```



Date :

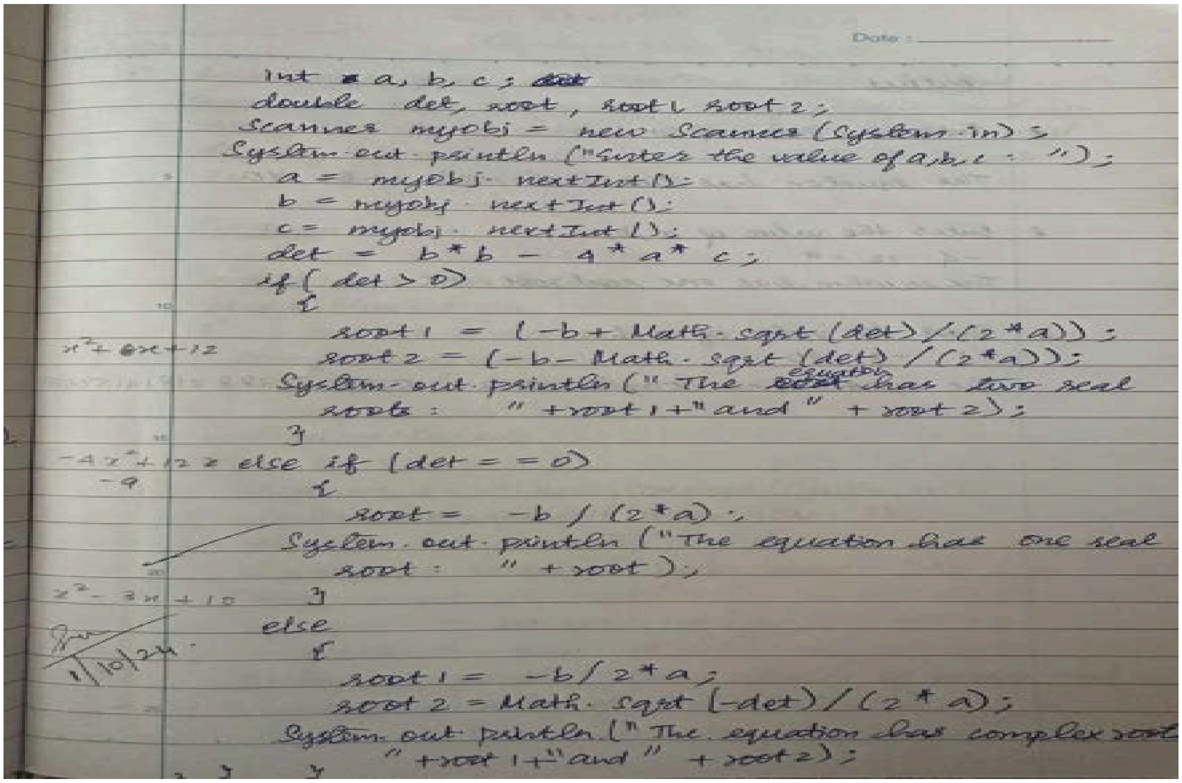
```
Enter two numbers :
 5 6
The product is : 30
```

2. Java program to take input from the user and display their product.

```
import java.util.Scanner;
public class Product
{
  public static void main (String [ ] args)
 {
    int x, y, product;
  Scanner myobj = new Scanner (System.in);
  System.out.println ("Enter two numbers : ");
   x = myobj.nextInt ();
   y = myobj.nextInt ();
  product = x * y;
 System.out.println (" The product is :   " + product
 }
}
```

3) Java program to find the roots of the quadr equation.

```
import java.util.Scanner;
public class Quadratic
{
 public static void main ( String [] args)
 {
```



Date :

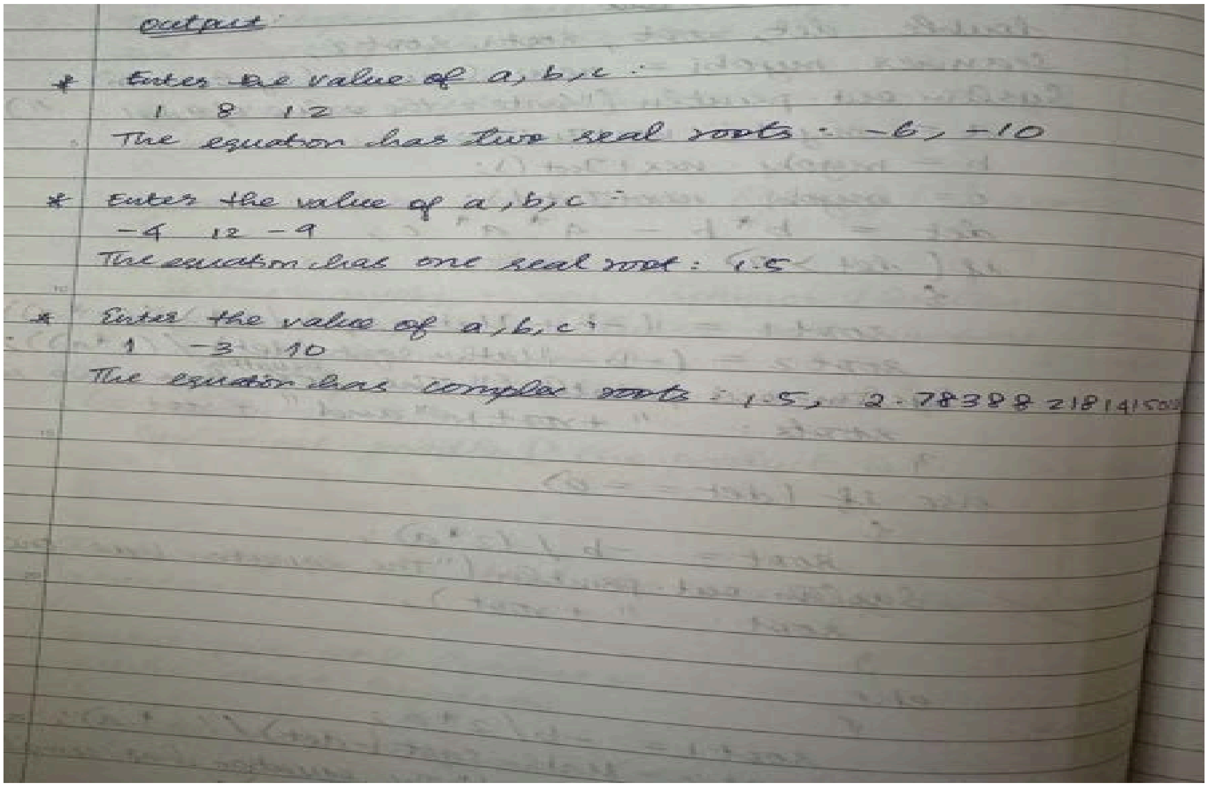
```
int ~~*~~ a, b, c; ~~det~~
double det, root, root1 root2;
Scanner myobj = new Scanner(System.in);
System.out.println("Enter the value of a,b,c : ");
 a = myobj.nextInt();
 b = myobj.nextInt();
 c= myobj.nextInt();
 det = b*b - 4*a*c;
 if( det > 0)
   {
     root1 = (-b+ Math.sqrt (det)/(2*a));
     root2 = (-b- Math.sqrt (det) /(2*a));
  System.out.println(" The ~~root~~ equation has two real
     roots :   " +root1+"and" + root2);
   }
 else if (det = = 0)
   {
     root =  -b / (2*a);
  System.out.println("The equation has one real
     root :   " + root);
   }
 else
   {
     root1 =  -b/ 2*a;
     root2 = Math.sqrt (-det)/(2*a);
  System.out.println(" The equation has complex root
        " +root1+"and" + root2);
 }  }
```

$x^2+6x+12$

$-4x^2+12x$
-9

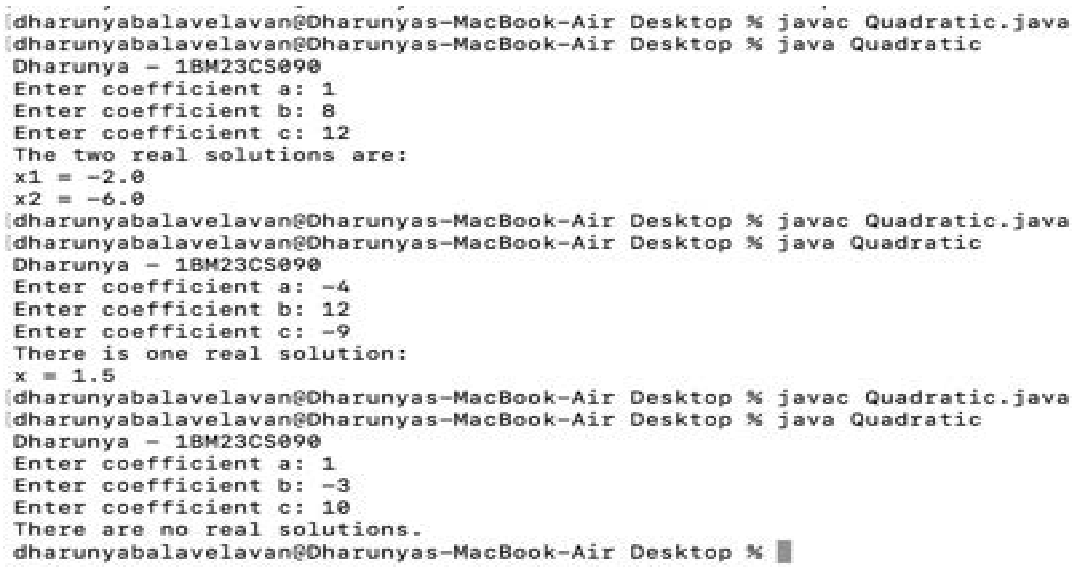
$x^2-3x+10$

1/10/24



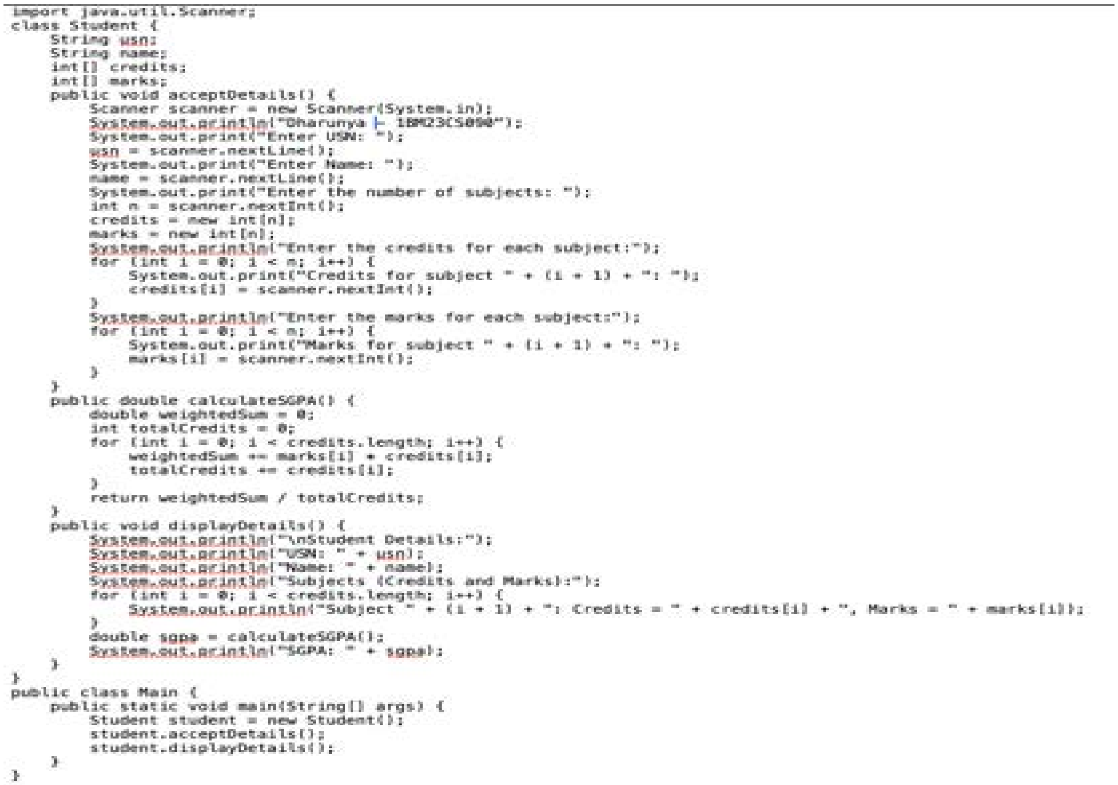
output

```
*  Enter the value of a,b,c :
    1   8   12
   The equation has two real roots : -6, -10

*  Enter the value of a,b,c :
   -4   12  -9
   The equation has one real root : 1.5

*  Enter the value of a,b,c :
    1   -3   10
   The equation has complex roots : 1.5,   2.7838821814150
```

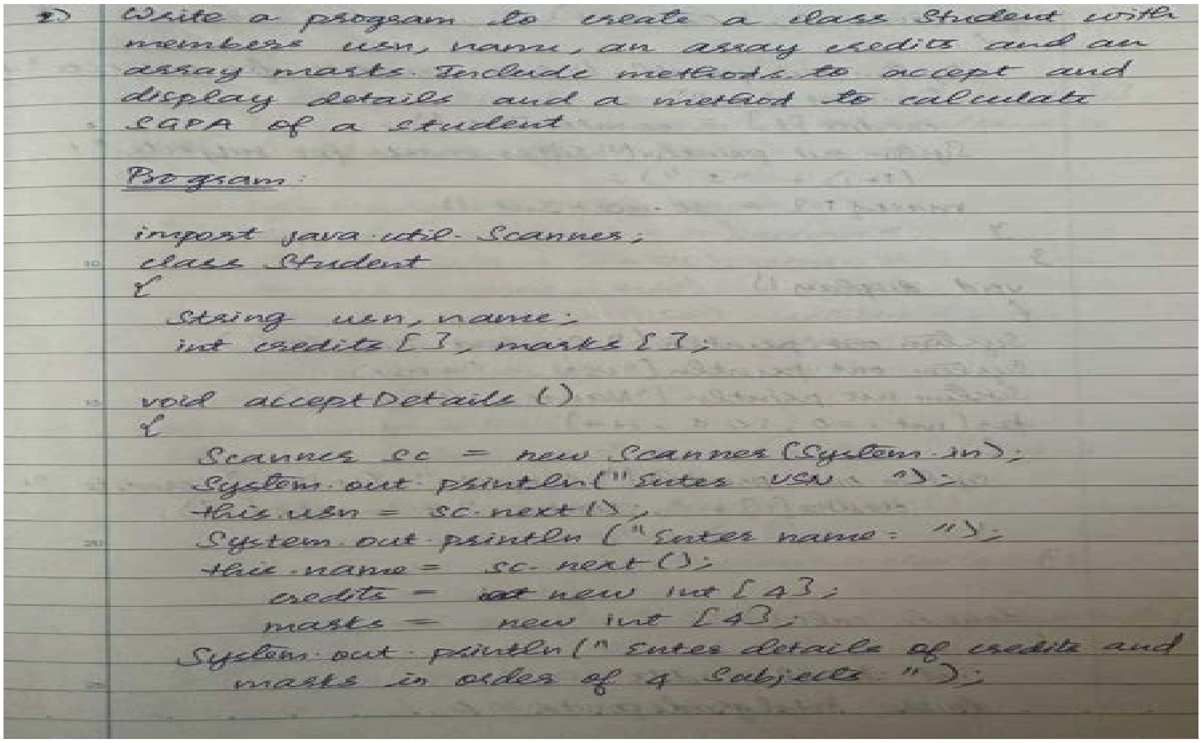


```
dharunyabalavelavan@Dharunyas-MacBook-Air Desktop % javac Quadratic.java
dharunyabalavelavan@Dharunyas-MacBook-Air Desktop % java Quadratic
Dharunya - 1BM23CS090
Enter coefficient a: 1
Enter coefficient b: 8
Enter coefficient c: 12
The two real solutions are:
x1 = -2.0
x2 = -6.0
dharunyabalavelavan@Dharunyas-MacBook-Air Desktop % javac Quadratic.java
dharunyabalavelavan@Dharunyas-MacBook-Air Desktop % java Quadratic
Dharunya - 1BM23CS090
Enter coefficient a: -4
Enter coefficient b: 12
Enter coefficient c: -9
There is one real solution:
x = 1.5
dharunyabalavelavan@Dharunyas-MacBook-Air Desktop % javac Quadratic.java
dharunyabalavelavan@Dharunyas-MacBook-Air Desktop % java Quadratic
Dharunya - 1BM23CS090
Enter coefficient a: 1
Enter coefficient b: -3
Enter coefficient c: 10
There are no real solutions.
dharunyabalavelavan@Dharunyas-MacBook-Air Desktop % 
```

2.Develop a Java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.



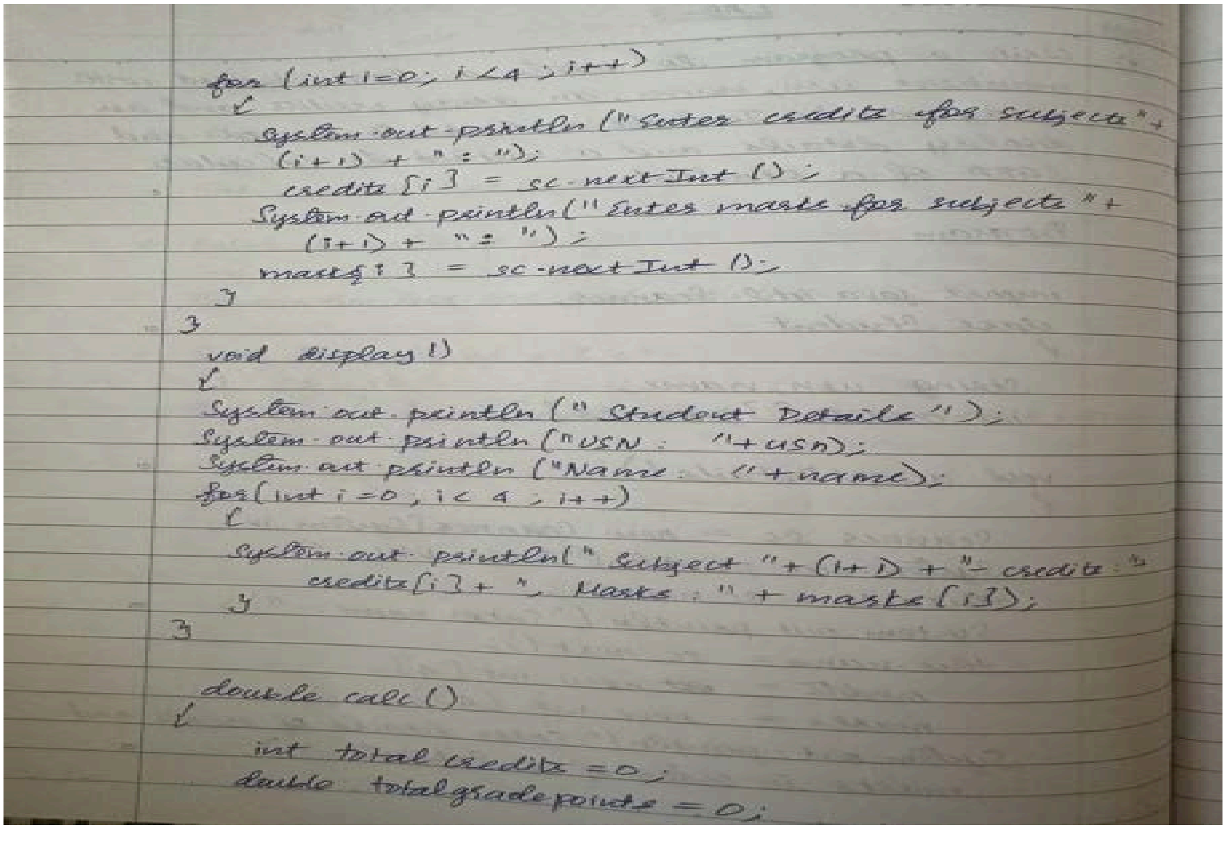
```
import java.util.Scanner;
class Student {
    String usn;
    String name;
    int[] credits;
    int[] marks;
    public void acceptDetails() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Dharunya - 1BM23CS090");
        System.out.print("Enter USN: ");
        usn = scanner.nextLine();
        System.out.print("Enter Name: ");
        name = scanner.nextLine();
        System.out.print("Enter the number of subjects: ");
        int n = scanner.nextInt();
        credits = new int[n];
        marks = new int[n];
        System.out.println("Enter the credits for each subject:");
        for (int i = 0; i < n; i++) {
            System.out.print("Credits for subject " + (i + 1) + ": ");
            credits[i] = scanner.nextInt();
        }
        System.out.println("Enter the marks for each subject:");
        for (int i = 0; i < n; i++) {
            System.out.print("Marks for subject " + (i + 1) + ": ");
            marks[i] = scanner.nextInt();
        }
    }
    public double calculateSGPA() {
        double weightedSum = 0;
        int totalCredits = 0;
        for (int i = 0; i < credits.length; i++) {
            weightedSum += marks[i] * credits[i];
            totalCredits += credits[i];
        }
        return weightedSum / totalCredits;
    }
    public void displayDetails() {
        System.out.println("\nStudent Details:");
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Subjects (Credits and Marks):");
        for (int i = 0; i < credits.length; i++) {
            System.out.println("Subject " + (i + 1) + ": Credits = " + credits[i] + ", Marks = " + marks[i]);
        }
        double sgpa = calculateSGPA();
        System.out.println("SGPA: " + sgpa);
    }
}
public class Main {
    public static void main(String[] args) {
        Student student = new Student();
        student.acceptDetails();
        student.displayDetails();
    }
}
```



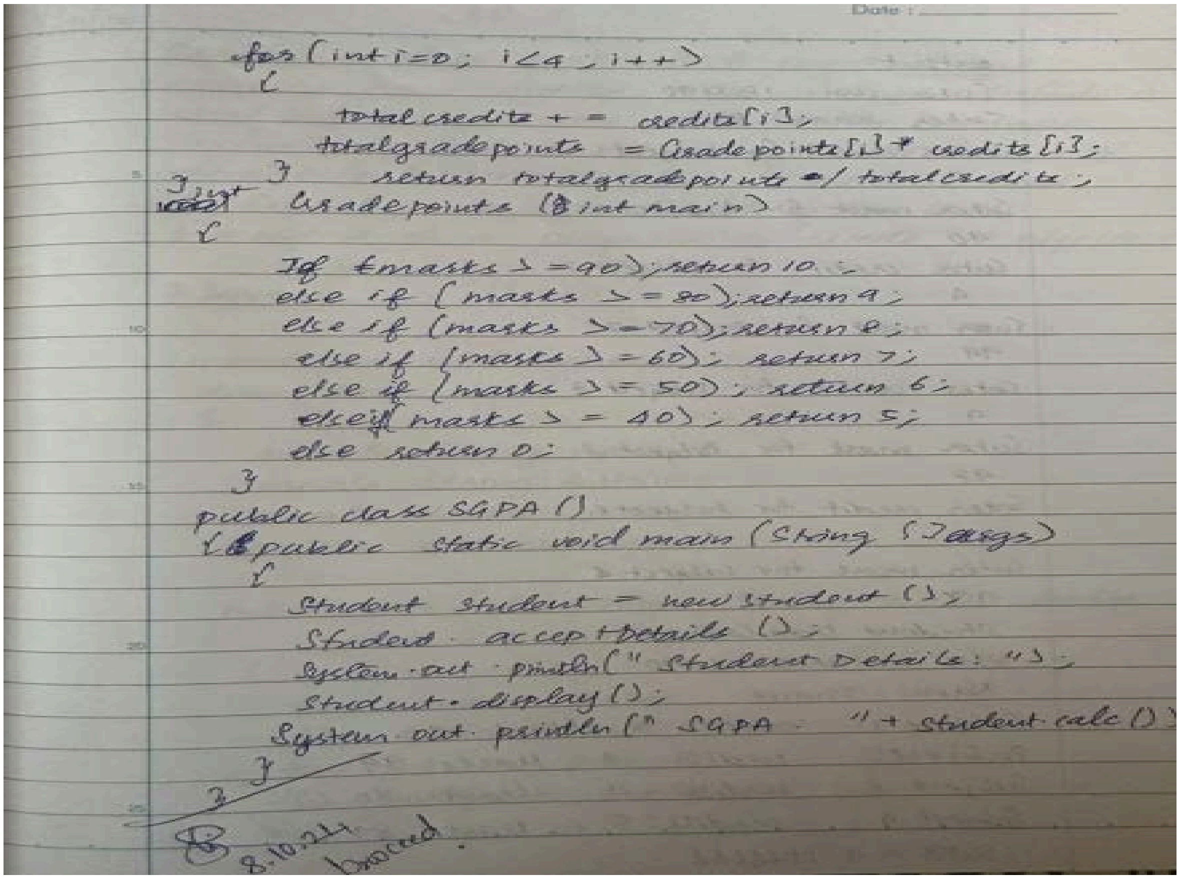
2) Write a program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student

Program:

```
import java.util.Scanner;
class Student
{
  String usn, name;
  int credits[], marks[];

void acceptDetails ()
{
   Scanner sc = new Scanner(System.in);
   System.out.println("Enter USN : ");
   this.usn = sc.next();
   System.out.println ("Enter name: ");
   this.name = sc.next();
      credits = ~~int~~ new int [4];
      marks =  new int [4];
   System.out.println(" Enter details of credits and
     marks in order of 4 Subjects ");
```



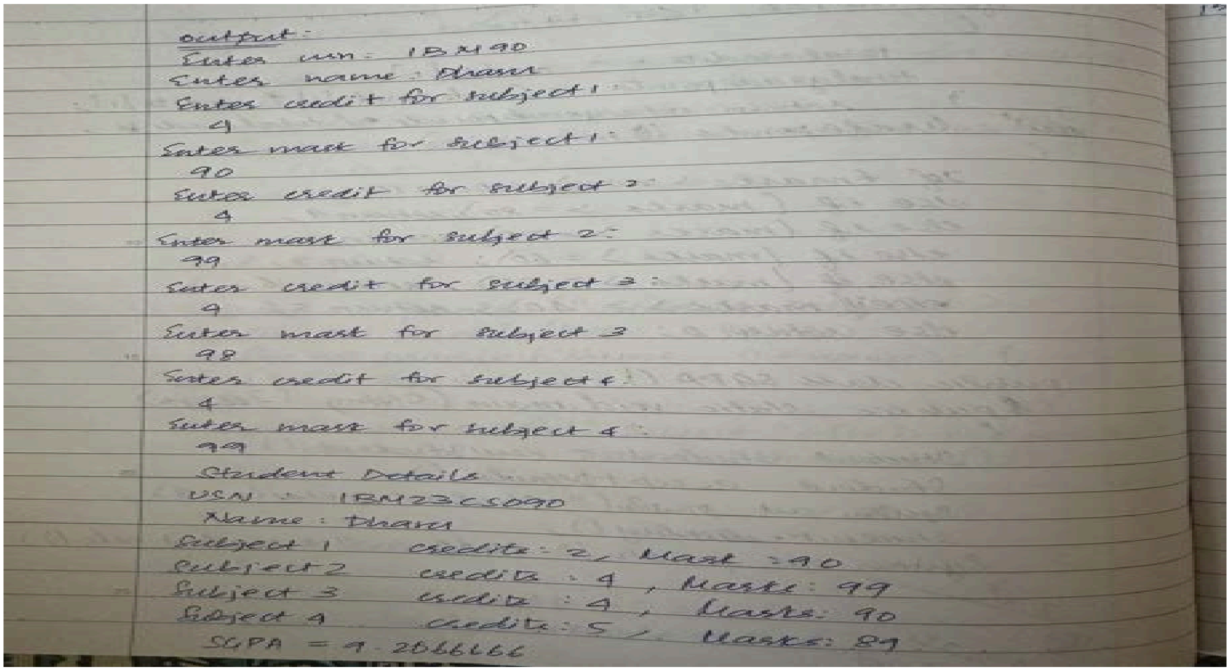
```
  for (int i=0; i<4 ;i++)
   {
     System.out.println("Enter credits for subjects"+
      (i+1) + ": ");
      credits [i] = sc.nextInt ();
    System.out.println("Enter marks for subjects "+
       (i+1) + ": ");
    marks[i] = sc.nextInt ();
   }
 }
  void display()
  {
  System.out.println (" Student Details ");
  System.out.println ("USN :  "+usn);
  System.out.println ("Name :  "+name);
  for(int i =0 ; i< 4 ; i++)
   {
    System.out.println(" Subject "+ (i+1) + "- credits: "+
        credits[i]+ ", Marks : " + marks[i]);
   }
 }

  double calc()
 {
    int total credits =0;
   double  totalgradepoints =0;
```



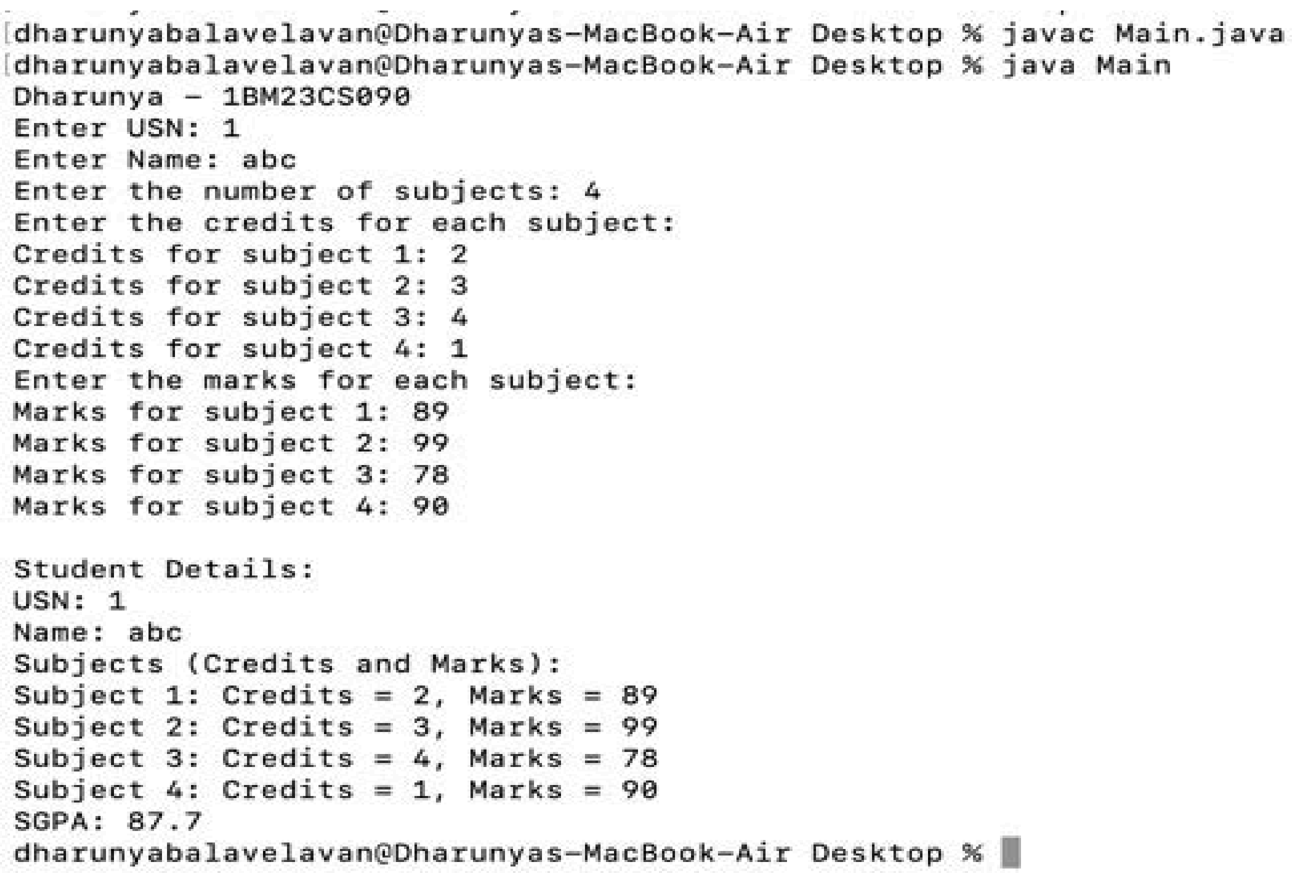
Date :

```
for (int i=0; i<4 ; i++)
 {
     totalcredits + = credits[i];
    totalgradepoints  = Gradepoints[i] * credits[i];
 }     return totalgradepoints / totalcredits ;
}int
    Gradepoints (int main)
 {
    If (marks > =90); return 10 ;
    else if (marks >= 80); return 9;
    else if (marks >=70); return 8;
     else if (marks > =60); return 7;
     else if (marks > = 50); return 6;
     else if marks > = 40); return 5;
     else return 0;
  }
public class SGPA ()
{ public static void main (String []args)
  {
     Student student = new student ();
     Student. acceptDetails ();
     System.out.println(" Student Details: ");
     Student . display ();
    System.out. println (" SGPA    " + student.calc ()
   }
 }
```

8.10.24

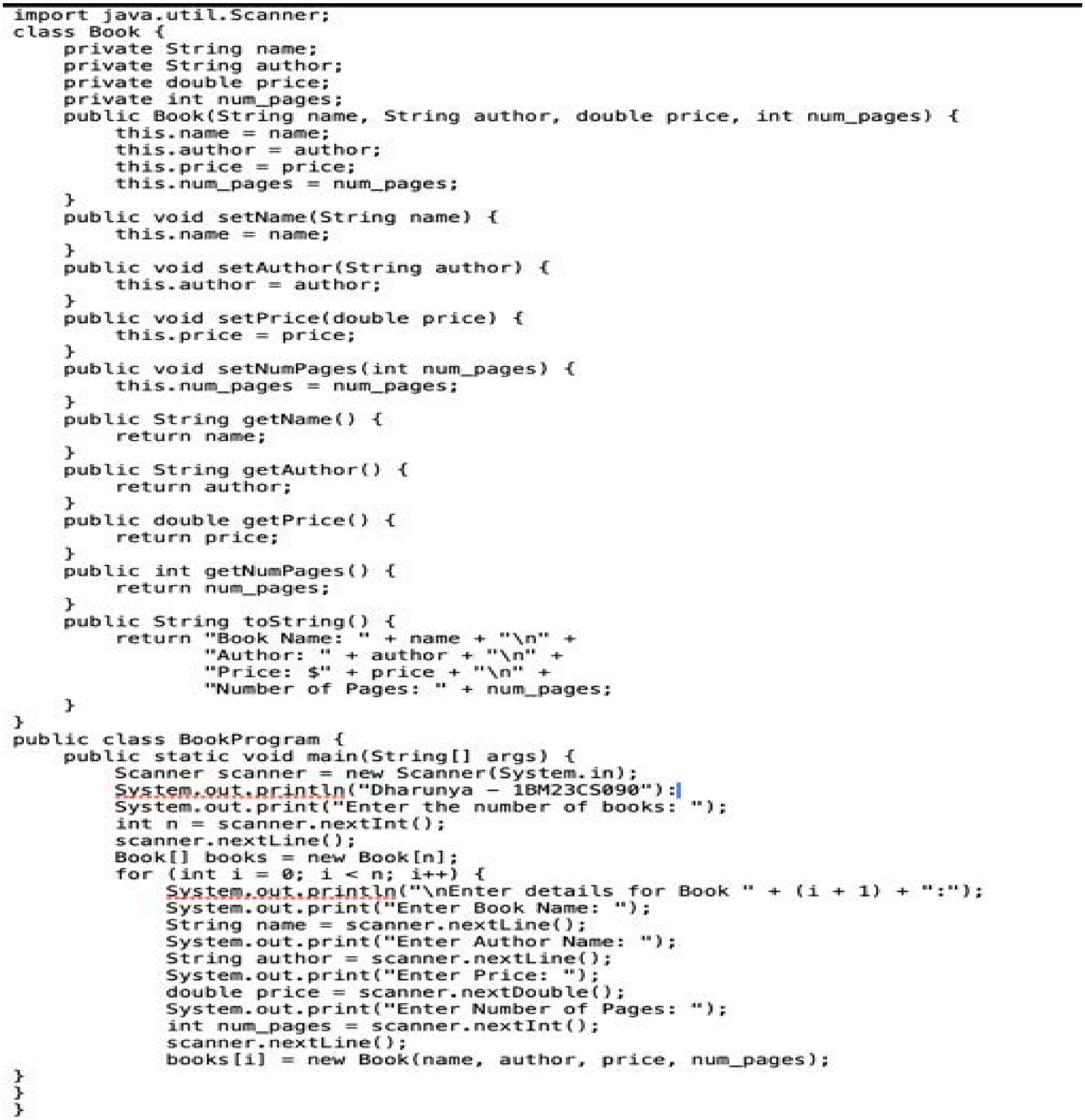


```
output:
Enter usn: 1BM90
Enter name: Dhani
Enter credit for subject1:
 4
Enter mark for subject1:
 90
Enter credit for subject 2:
  4
Enter mark for subject 2:
 99
Enter credit for subject 3:
 9
Enter mark for subject 3
 98
Enter credit for subject4:
 4
Enter mark for subject 4 :
 99
  Student Details
 USN  :  1BM23CS090
  Name : Dhani
Subject 1      credits: 2, Mark :90
Subject2       credits : 4 , Marks: 99
Subject 3      credits  : 4 , Marks: 90
Subject 4       credits: 5 ,  Marks: 89
  SGPA = 9.2066666
```

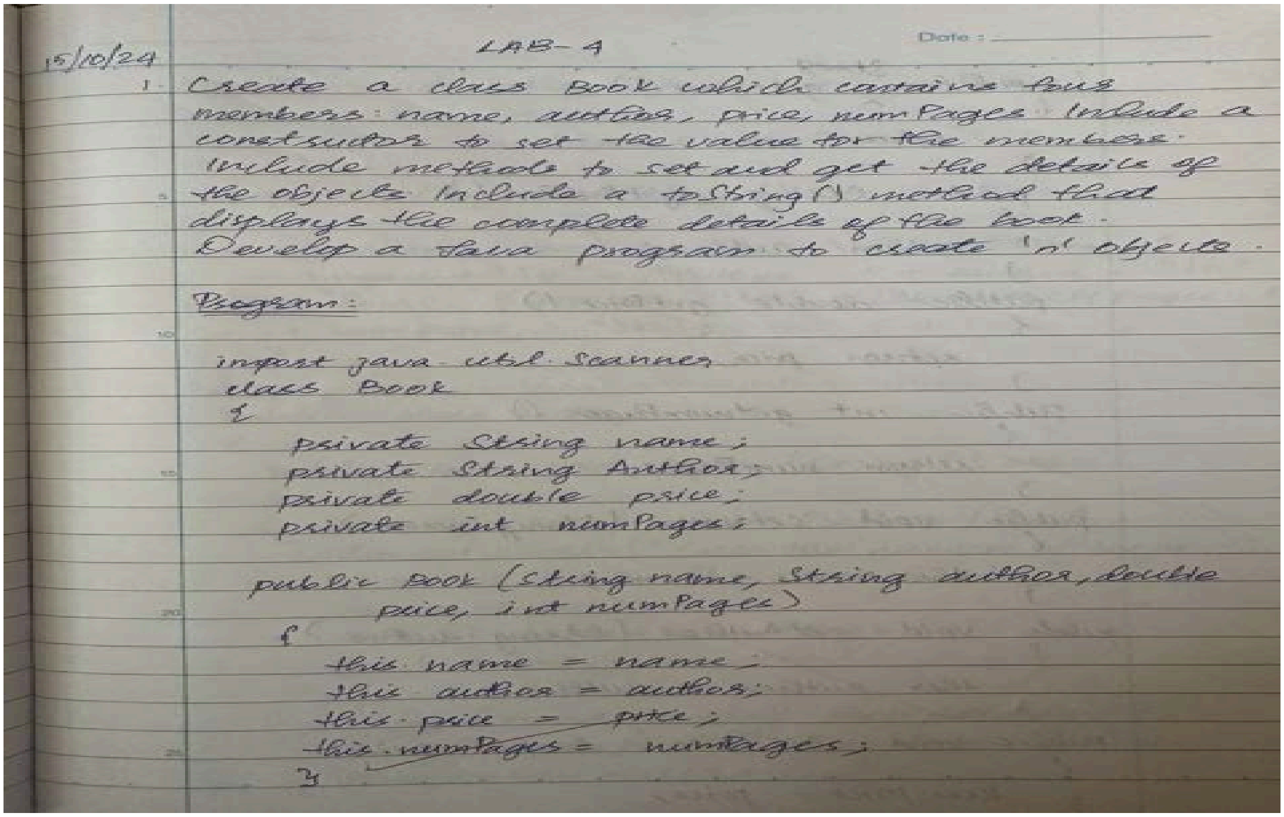


```
[dharunyabalavelavan@Dharunyas-MacBook-Air Desktop % javac Main.java
[dharunyabalavelavan@Dharunyas-MacBook-Air Desktop % java Main
Dharunya - 1BM23CS090
Enter USN: 1
Enter Name: abc
Enter the number of subjects: 4
Enter the credits for each subject:
Credits for subject 1: 2
Credits for subject 2: 3
Credits for subject 3: 4
Credits for subject 4: 1
Enter the marks for each subject:
Marks for subject 1: 89
Marks for subject 2: 99
Marks for subject 3: 78
Marks for subject 4: 90

Student Details:
USN: 1
Name: abc
Subjects (Credits and Marks):
Subject 1: Credits = 2, Marks = 89
Subject 2: Credits = 3, Marks = 99
Subject 3: Credits = 4, Marks = 78
Subject 4: Credits = 1, Marks = 90
SGPA: 87.7
dharunyabalavelavan@Dharunyas-MacBook-Air Desktop % 
```

3.Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString( ) method that could display the complete details of the book. Develop a Java program to create n book objects.



```
import java.util.Scanner;
class Book {
    private String name;
    private String author;
    private double price;
    private int num_pages;
    public Book(String name, String author, double price, int num_pages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.num_pages = num_pages;
    }
    public void setName(String name) {
        this.name = name;
    }
    public void setAuthor(String author) {
        this.author = author;
    }
    public void setPrice(double price) {
        this.price = price;
    }
    public void setNumPages(int num_pages) {
        this.num_pages = num_pages;
    }
    public String getName() {
        return name;
    }
    public String getAuthor() {
        return author;
    }
    public double getPrice() {
        return price;
    }
    public int getNumPages() {
        return num_pages;
    }
    public String toString() {
        return "Book Name: " + name + "\n" +
               "Author: " + author + "\n" +
                   "Price: $" + price + "\n" +
               "Number of Pages: " + num_pages;
    }
}
public class BookProgram {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Dharunya - 1BM23CS090");
        System.out.print("Enter the number of books: ");
        int n = scanner.nextInt();
        scanner.nextLine();
        Book[] books = new Book[n];
        for (int i = 0; i < n; i++) {
            System.out.println("\nEnter details for Book " + (i + 1) + ":");
            System.out.print("Enter Book Name: ");
            String name = scanner.nextLine();
            System.out.print("Enter Author Name: ");
            String author = scanner.nextLine();
            System.out.print("Enter Price: ");
            double price = scanner.nextDouble();
            System.out.print("Enter Number of Pages: ");
            int num_pages = scanner.nextInt();
            scanner.nextLine();
            books[i] = new Book(name, author, price, num_pages);
}
}
}
```

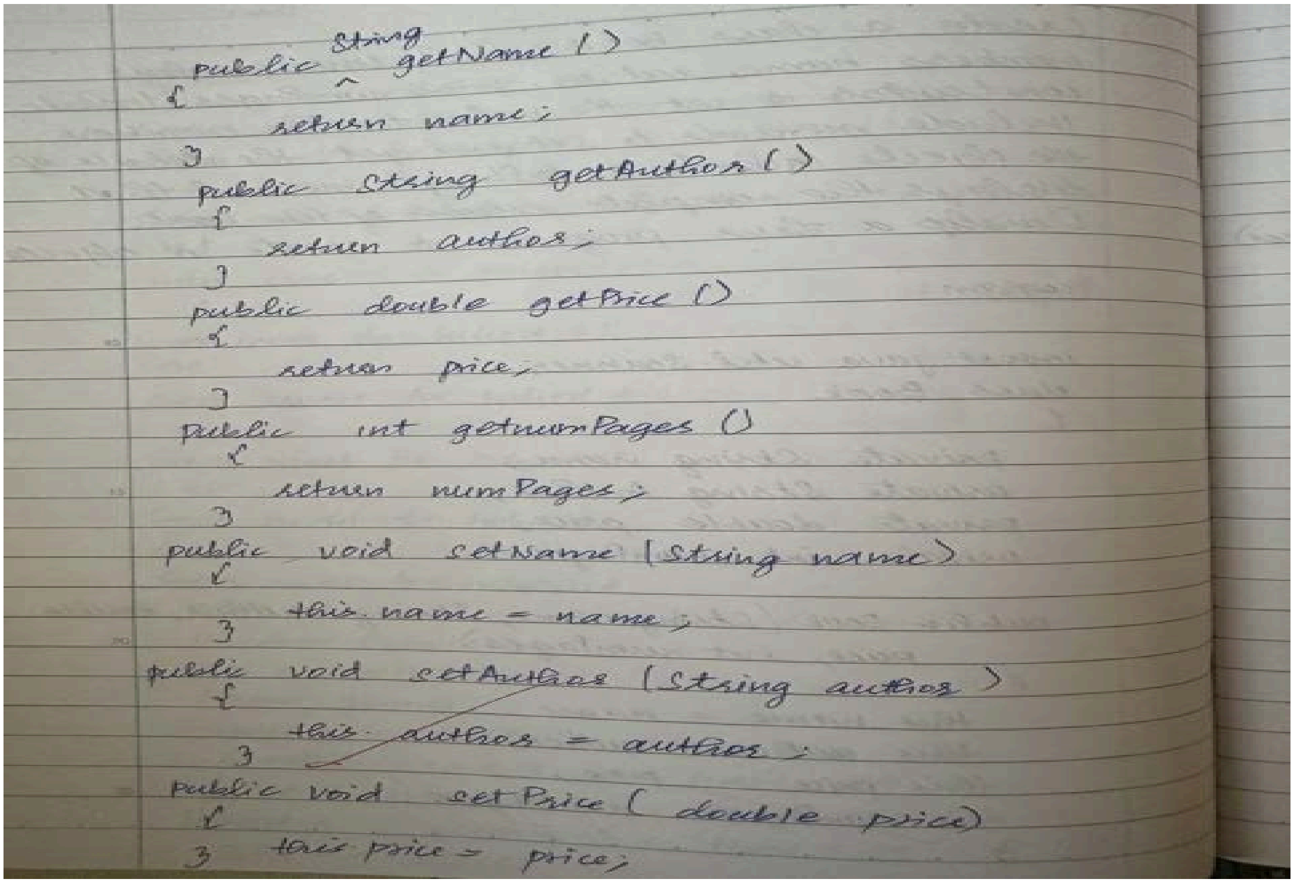


15/10/24 LAB-4 Date :

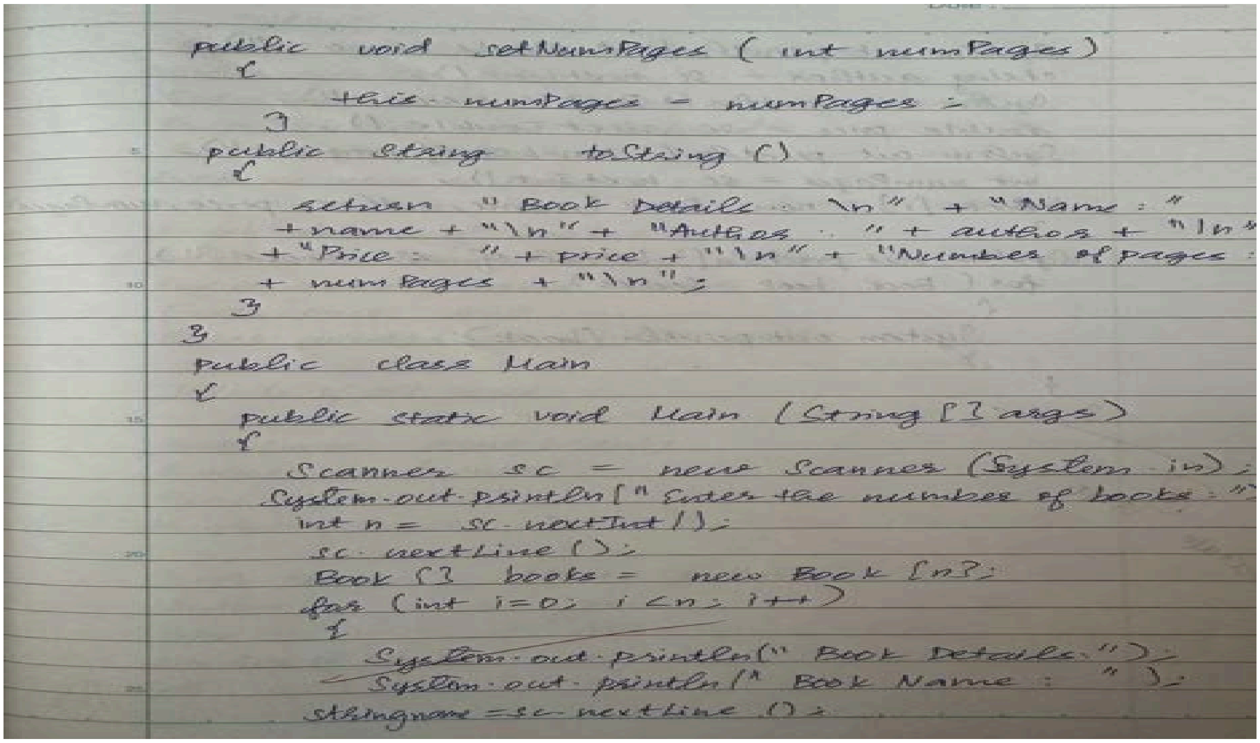
1. Create a class Book which contains four members: name, author, price, numPages Include a constructor to set the value for the members. Include methods to set and get the details of the objects Include a toString() method that displays the complete details of the book. Develop a Java program to create 'n' objects.

Program:

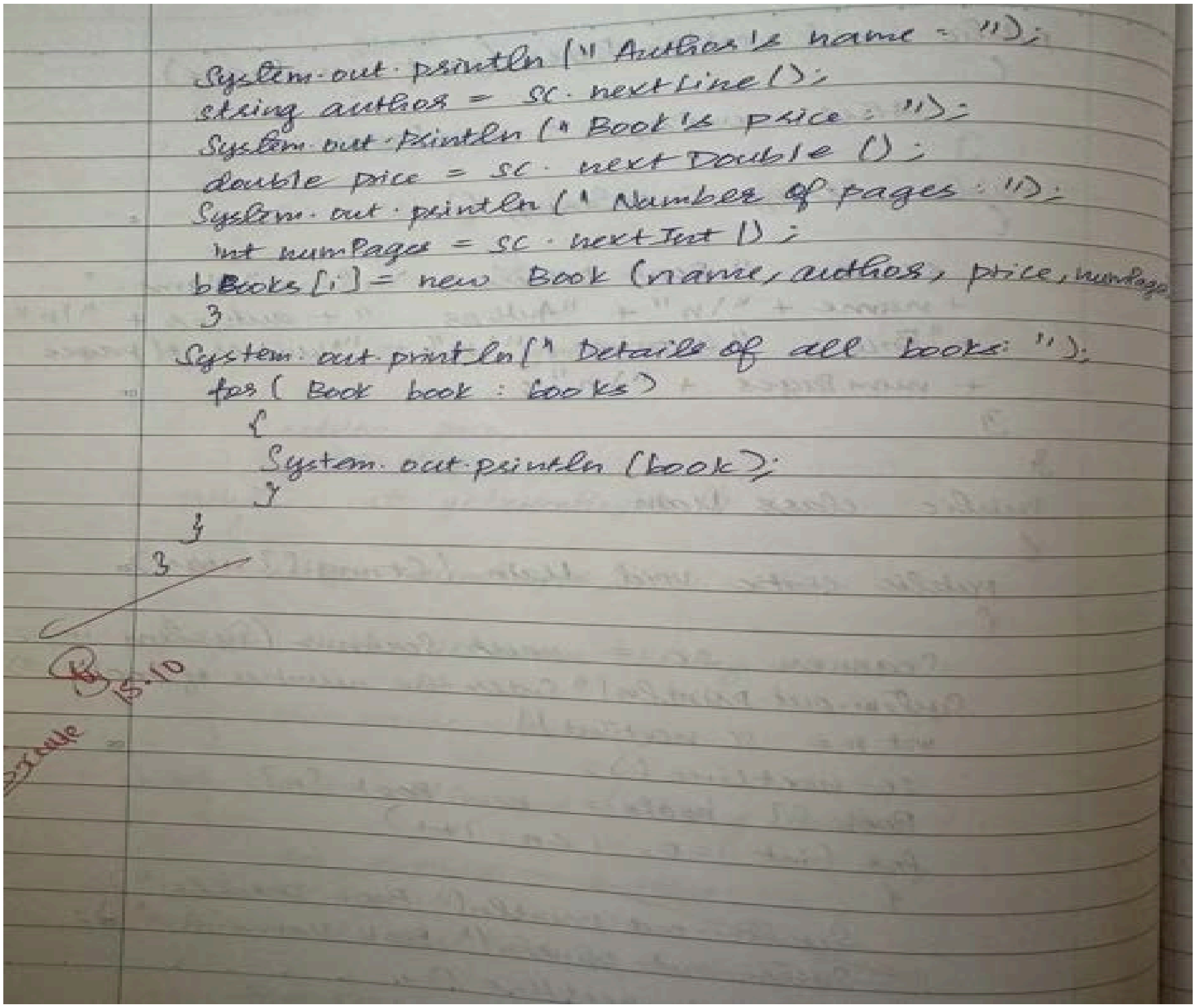
```
import java.util.Scanner
class Book
{
    private String name;
    private String Author;
    private double price;
    private int numPages;

  public Book (String name, String author, double
        price, int numPages)
  {
     this.name = name;
     this.author = author;
     this.price = price;
     this.numPages = numPages;
  }
```



```
public String getName ()
{
    return name;
}
public String getAuthor ()
 {
    return author;
 }
public double getPrice ()
 {
    return price;
 }
public int getnumPages ()
 {
    return numPages;
 }
public void setName (String name)
 {
    this.name = name;
 }
public void setAuthor (String author)
 {
    this.author = author;
 }
public void setPrice ( double price)
 {
 }  this.price = price;
```

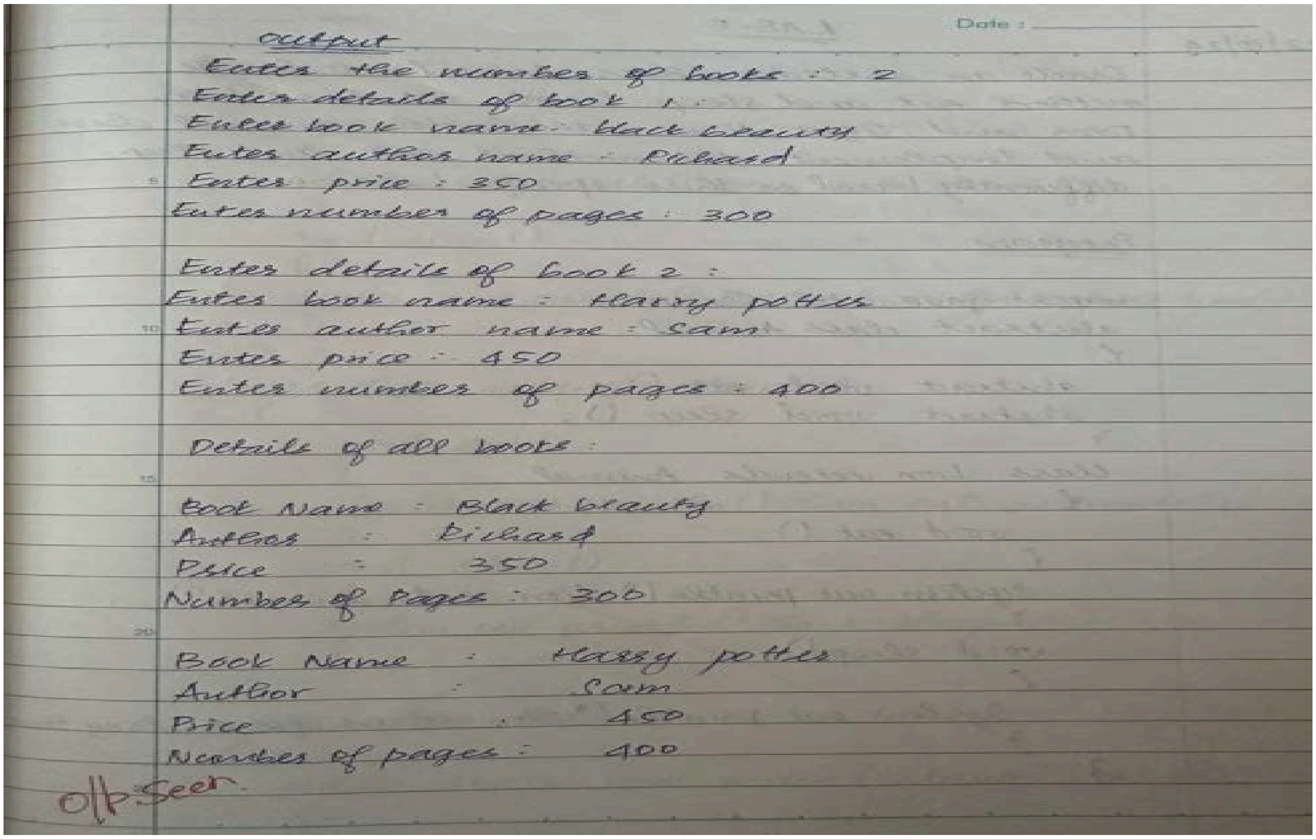


```
public void setNumPages ( int numPages)
  {
      this.numPages = numPages ;
   }
public String toString ()
  {
     return " Book Details : \n" + "Name : "
    +name + "\n" + "Author : " + author + "\n"
   + "Price : " + price + "\n" + "Number of pages :
   + numPages + "\n";
   }
}
public class Main
{
  public static void Main (String [] args)
  {
    Scanner sc = new Scanner (System.in);
   System.out.println (" Enter the number of books : "
    int n = sc.nextInt ();
     sc.nextLine ();
     Book [] books = new Book [n];
    for (int i=0; i<n; i++)
     {
       ~~System.out.println(" Book Details : ");~~
       System.out.println (" Book Name : " );
    stringname = sc.nextLine ();
```



```
System.out.println (" Author's name : ");
string author = sc.nextLine ();
System.out.Println (" Book's price : ");
double price = sc.nextDouble ();
System.out.println (" Number of pages : ");
 int numPages = sc.nextInt ();
bBooks [i] = new Book (name, author, price, numPages
 }
System.out.println (" Details of all books: ");
 for ( Book book : books)
   {
    System.out.println (book);
   }
 }
}
```

15.10



Date :

output

Enter the number of books : 2
Enter details of book 1 :
Enter book name : black beauty
Enter author name : Richard
Enter price : 350
Enter number of pages : 300

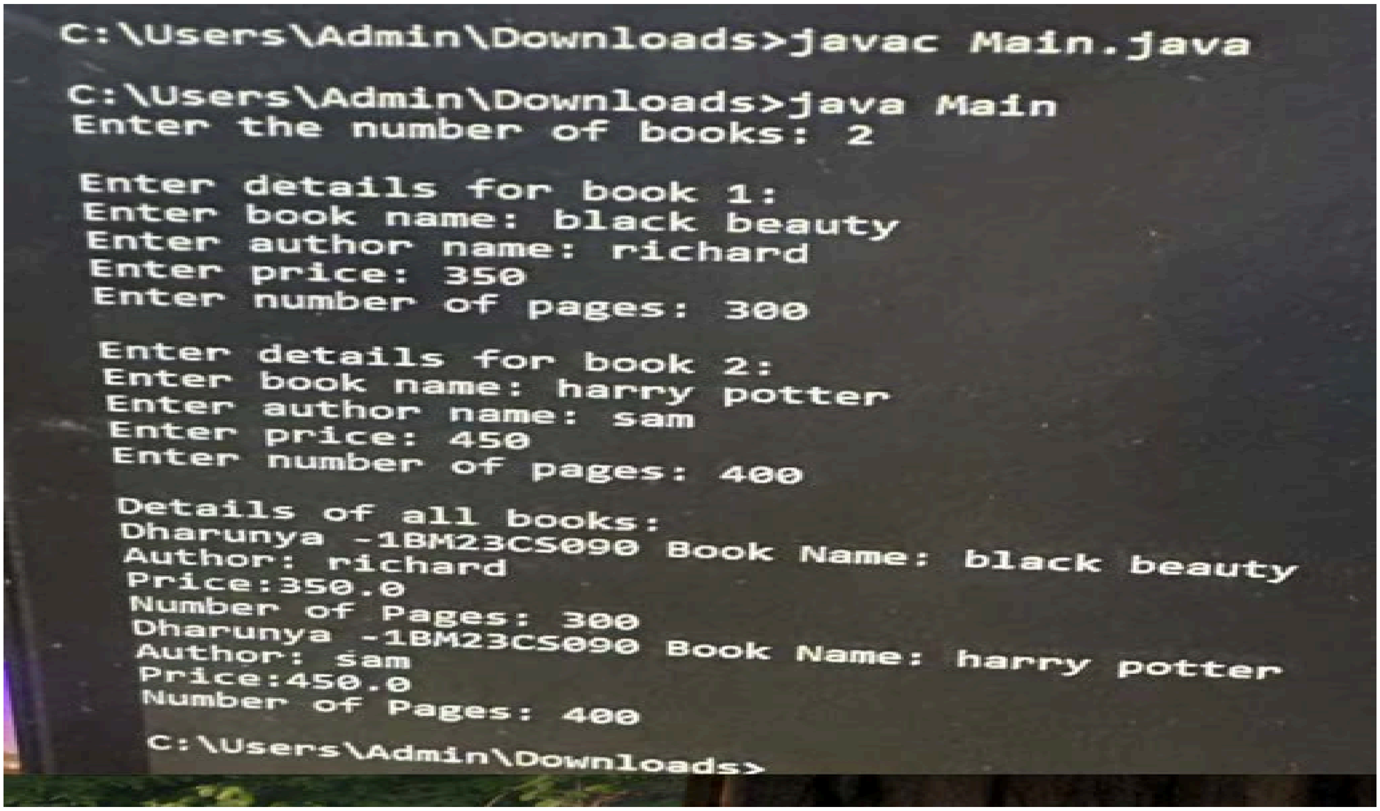
Enter details of book 2 :
Enter book name : Harry potter
Enter author name : sam
Enter price : 450
Enter number of pages : 400

Details of all books :

Book Name : Black beauty
Author : Richard
Price : 350
Number of pages : 300

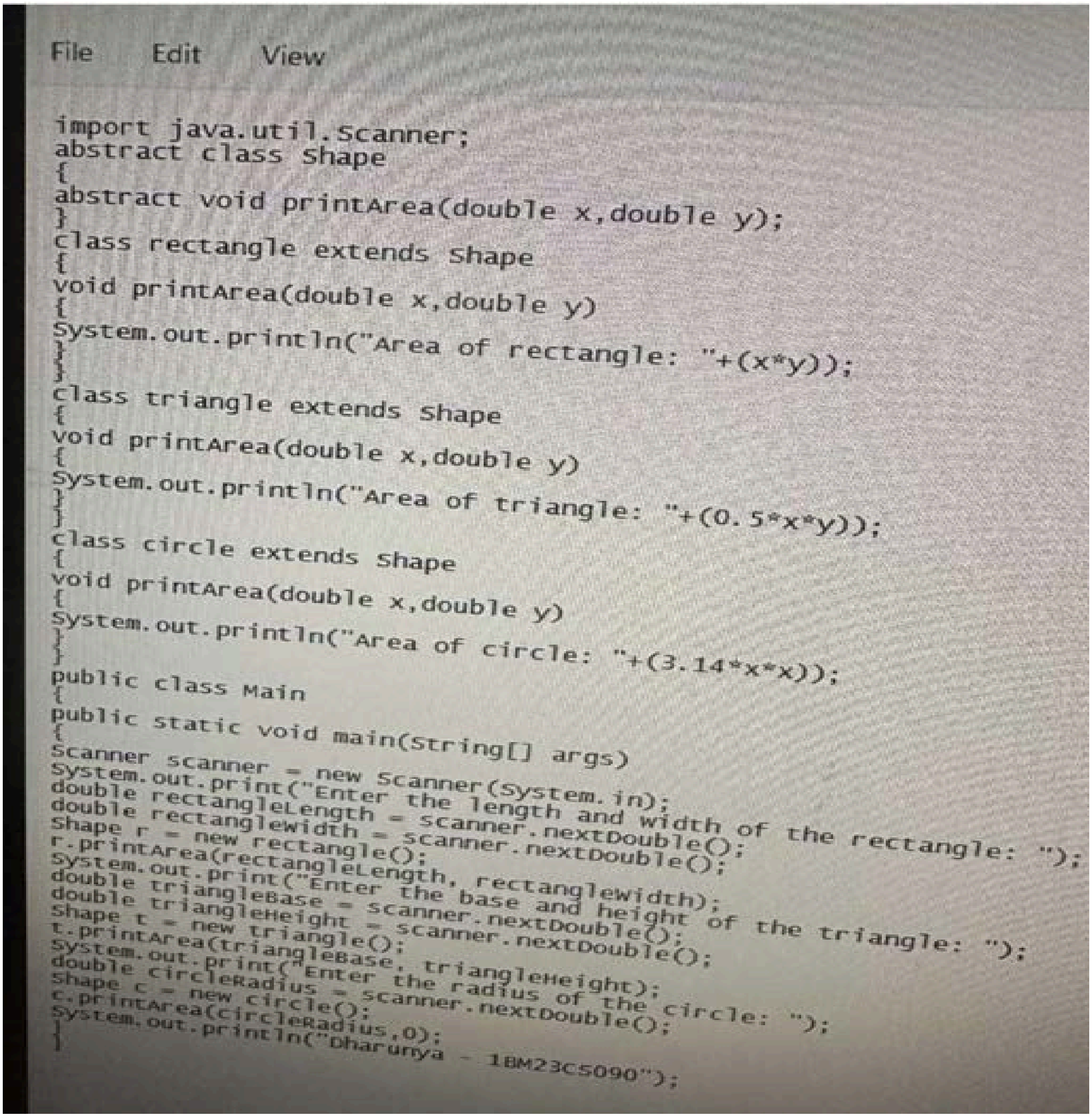
Book Name : Harry potter
Author : Sam
Price : 450
Number of pages : 400

O/p seen.

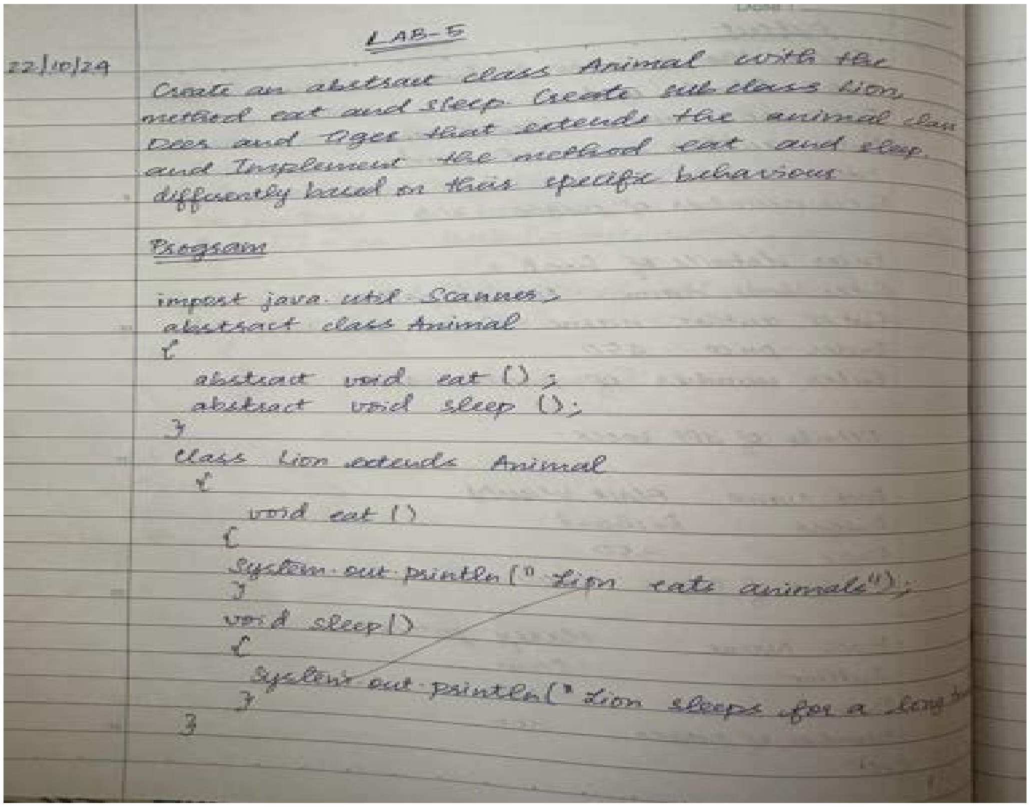


```
C:\Users\Admin\Downloads>javac Main.java

C:\Users\Admin\Downloads>java Main
Enter the number of books: 2

Enter details for book 1:
Enter book name: black beauty
Enter author name: richard
Enter price: 350
Enter number of pages: 300

Enter details for book 2:
Enter book name: harry potter
Enter author name: sam
Enter price: 450
Enter number of pages: 400

Details of all books:
Dharunya -1BM23CS090 Book Name: black beauty
Author: richard
Price:350.0
Number of Pages: 300
Dharunya -1BM23CS090 Book Name: harry potter
Author: sam
Price:450.0
Number of Pages: 400

C:\Users\Admin\Downloads>
```

4.Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method printArea() that prints the area of the given shape.



```
import java.util.Scanner;
abstract class Shape
{
abstract void printArea(double x,double y);
}
class rectangle extends Shape
{
void printArea(double x,double y)
{
System.out.println("Area of rectangle: "+(x*y));
}
}
class triangle extends Shape
{
void printArea(double x,double y)
{
System.out.println("Area of triangle: "+(0.5*x*y));
}
}
class circle extends Shape
{
void printArea(double x,double y)
{
System.out.println("Area of circle: "+(3.14*x*x));
}
}
public class Main
{
public static void main(String[] args)
{
Scanner scanner = new Scanner(System.in);
System.out.print("Enter the length and width of the rectangle: ");
double rectangleLength = scanner.nextDouble();
double rectanglewidth = scanner.nextDouble();
Shape r = new rectangle();
r.printArea(rectangleLength, rectanglewidth);
System.out.print("Enter the base and height of the triangle: ");
double triangleBase = scanner.nextDouble();
double triangleHeight = scanner.nextDouble();
Shape t = new triangle();
t.printArea(triangleBase, triangleHeight);
System.out.print("Enter the radius of the circle: ");
double circleRadius = scanner.nextDouble();
Shape c = new circle();
c.printArea(circleRadius,0);
System.out.println("Dharunya - 1BM23CS090");
}
}
```



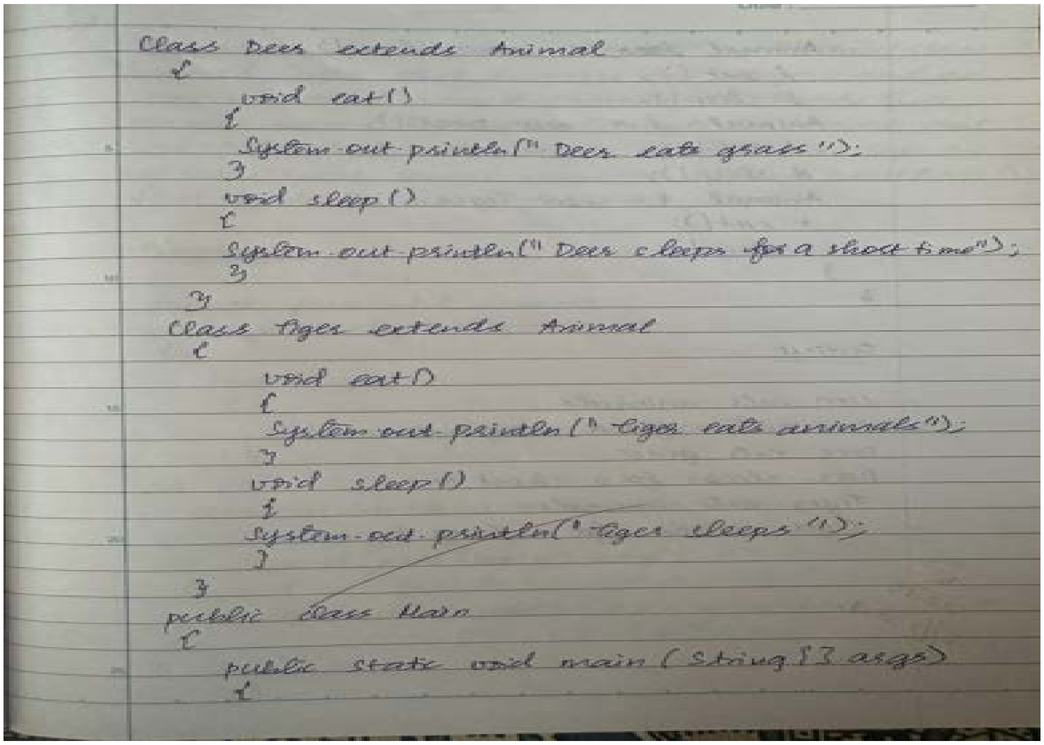
LAB-5

22/10/24

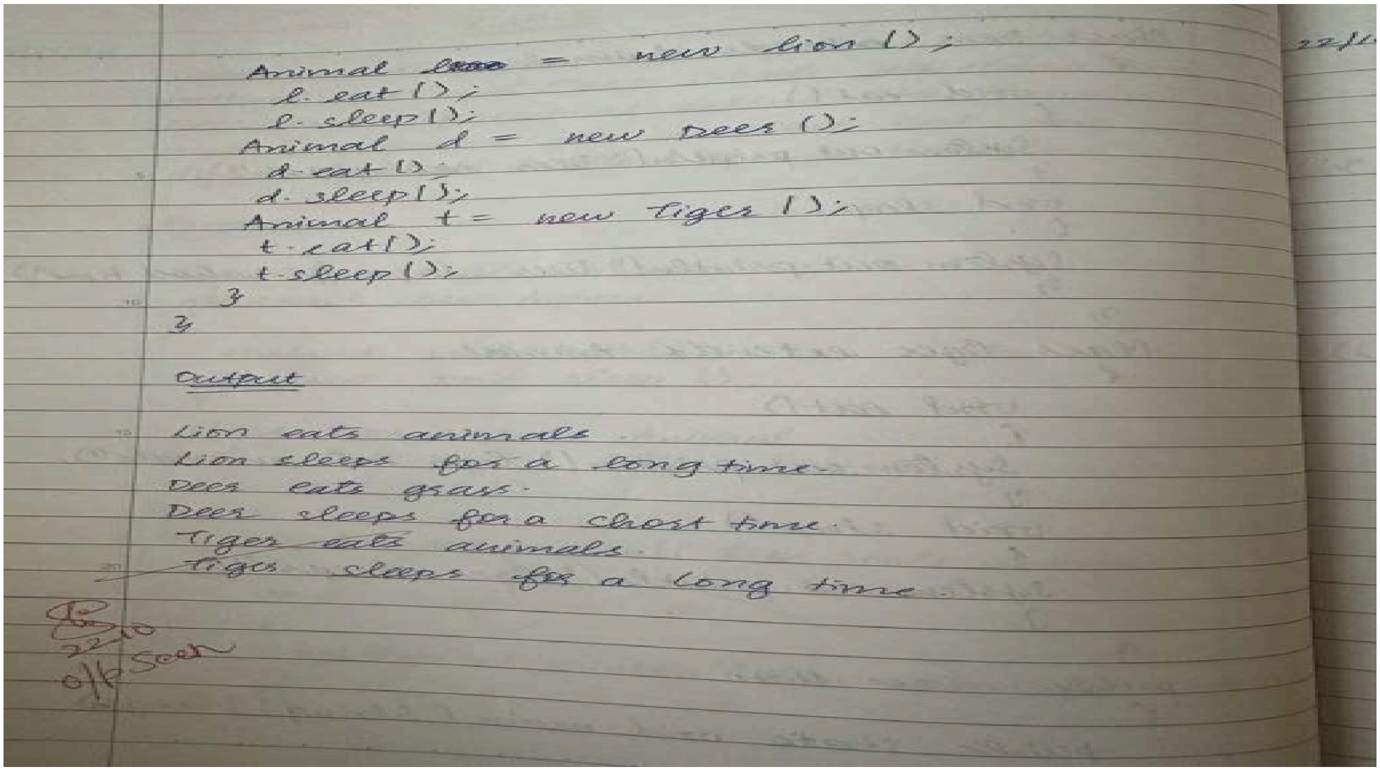
Create an abstract class Animal with the method eat and sleep. Create sub class Lion, Deer and Tiger that extends the animal class and Implement the method eat and sleep differently based on their specific behaviour

Program

```
import java.util.Scanner;
abstract class Animal
{
  abstract void eat();
  abstract void sleep();
}
class Lion extends Animal
 {
    void eat()
   {
   System.out.println("Lion eats animals");
   }
   void sleep()
   {
    System.out.println("Lion sleeps for a long
   }
 }
```



```
Class Deer extends Animal
 {
     void eat()
     {
      System.out.println("Deer eats grass");
     }
     void sleep()
     {
     System.out.println("Deer sleeps for a short time");
     }
  }
 Class Tiger extends Animal
  {
       void eat()
       {
        System.out.println("Tiger eats animals");
       }
       void sleep()
       {
       System.out.println("Tiger sleeps");
       }
  }
 public class Main
  {
     public static void main(String[] args)
     {
```



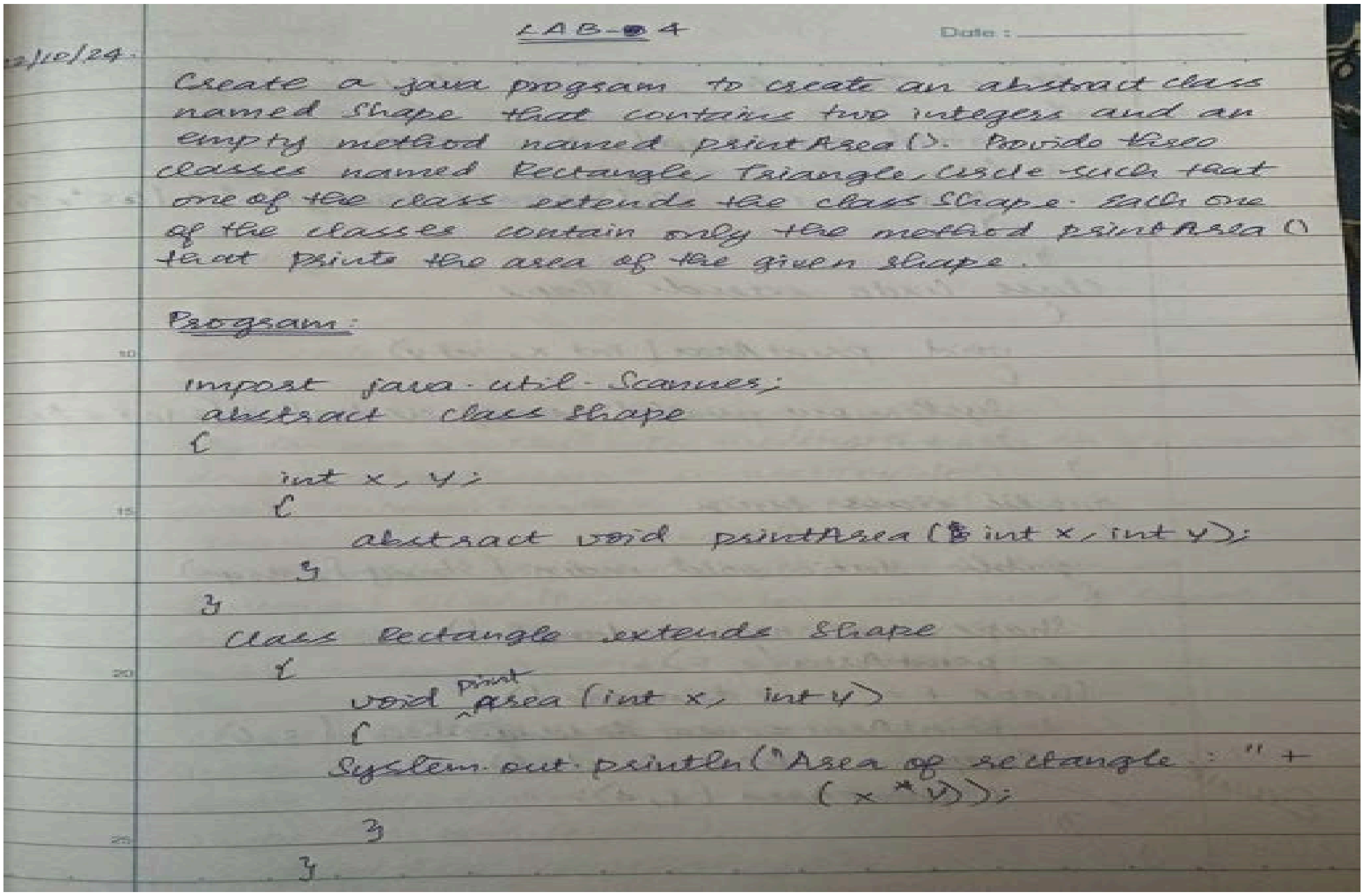
22/1

```
Animal l = new lion ();
  l.eat ();
  l.sleep ();
Animal d = new Deer ();
  d.eat ();
  d.sleep ();
Animal t = new Tiger ();
  t.eat ();
  t.sleep ();
 }
}
```

Output

```
Lion eats animals.
Lion sleeps for a long time.
Deer eats grass.
Deer sleeps for a short time.
Tiger eats animals.
Tiger sleeps for a long time.
```

22/10
o/p Seen



LAB- 4

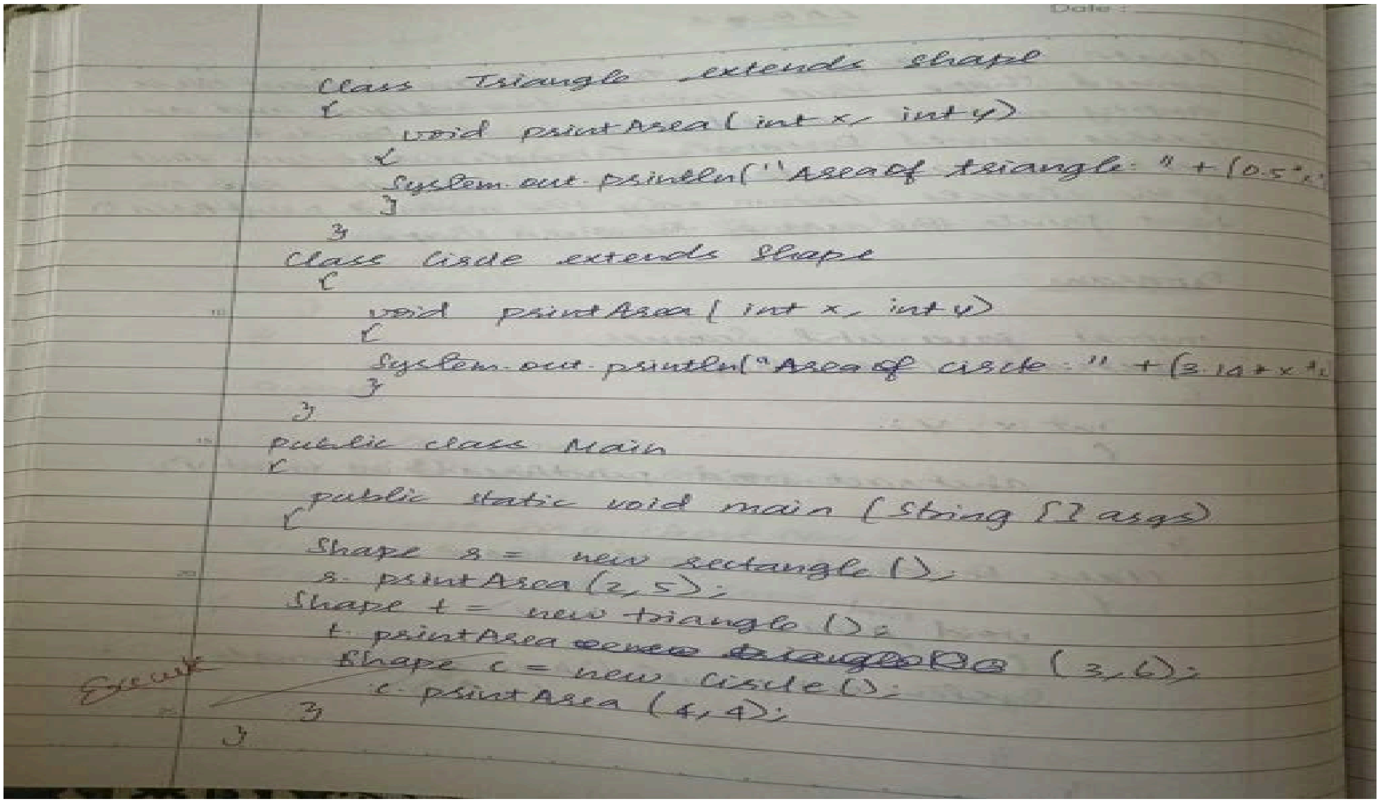
Date :

2/10/24.

Create a java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle, Circle such that one of the class extends the class Shape. Each one of the classes contain only the method printArea () that prints the area of the given shape.

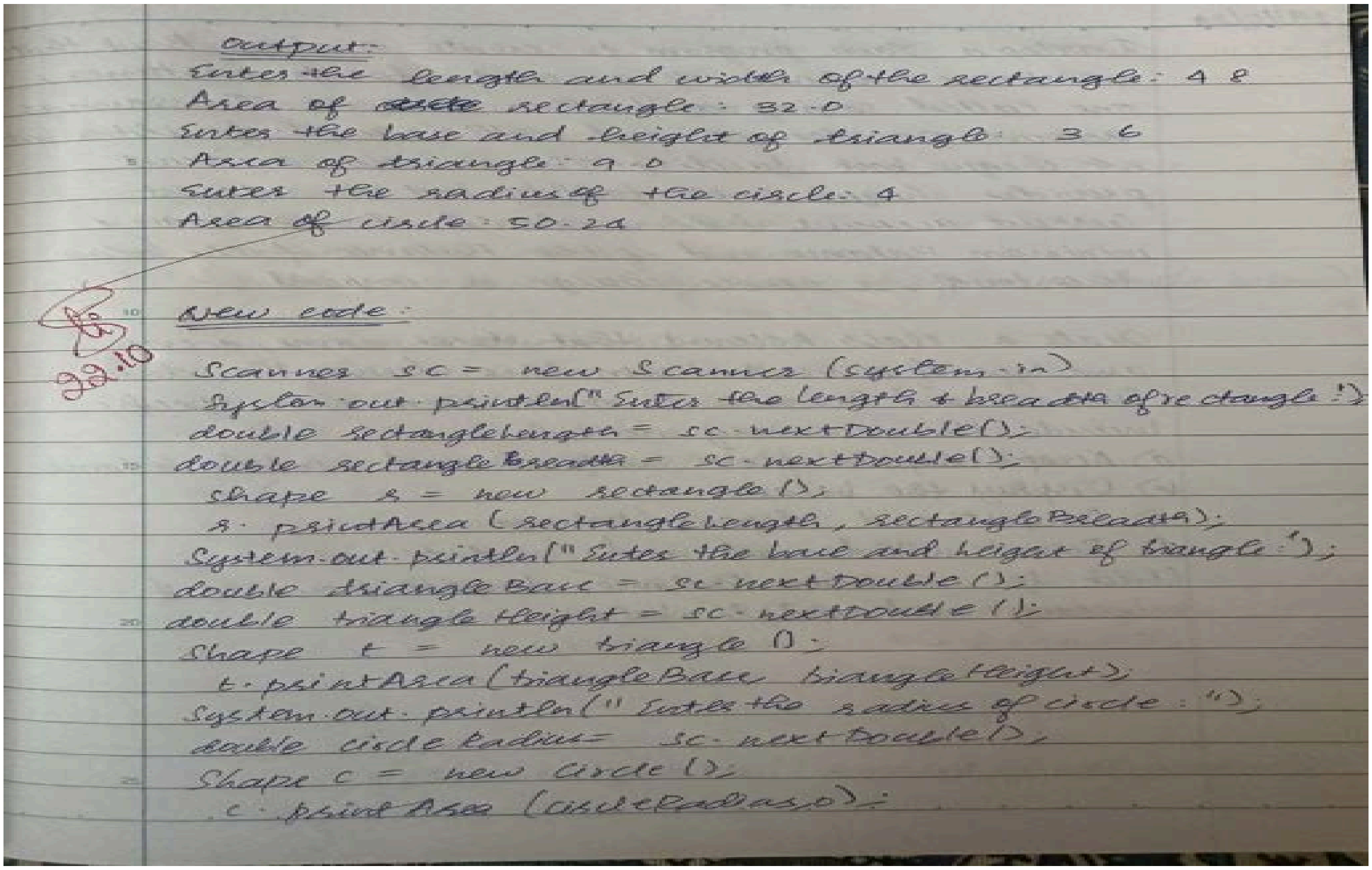
Program:

```
import java.util.Scanner;
abstract class Shape
{
    int x, y;
    {
        abstract void printArea (int x, int y);
    }
}
 class Rectangle extends Shape
    {
        void printArea (int x, int y)
        {
        System.out.println("Area of rectangle : " +
                              (x * y));
        }
    }
```



```
Class Triangle extends shape
{
    void printArea( int x, int y)
   {
   System.out.println("Areaof triangle: " + (0.5*x
   }
 }
Class circle extends Shape
 {
   void printArea( int x, int y)
   {
   System.out.println("Area of circle : " + (3.14*x*x
   }
 }
public class Main
{
 public static void main (String [] args)
 {
  Shape s = new rectangle();
  s.printArea(2,5);
 Shape t = new triangle();
  t.printArea ~~[illegible]~~ (3,6);
  Shape c = new circle();
   c.printArea (4,4);
  }
}
```

Excuk



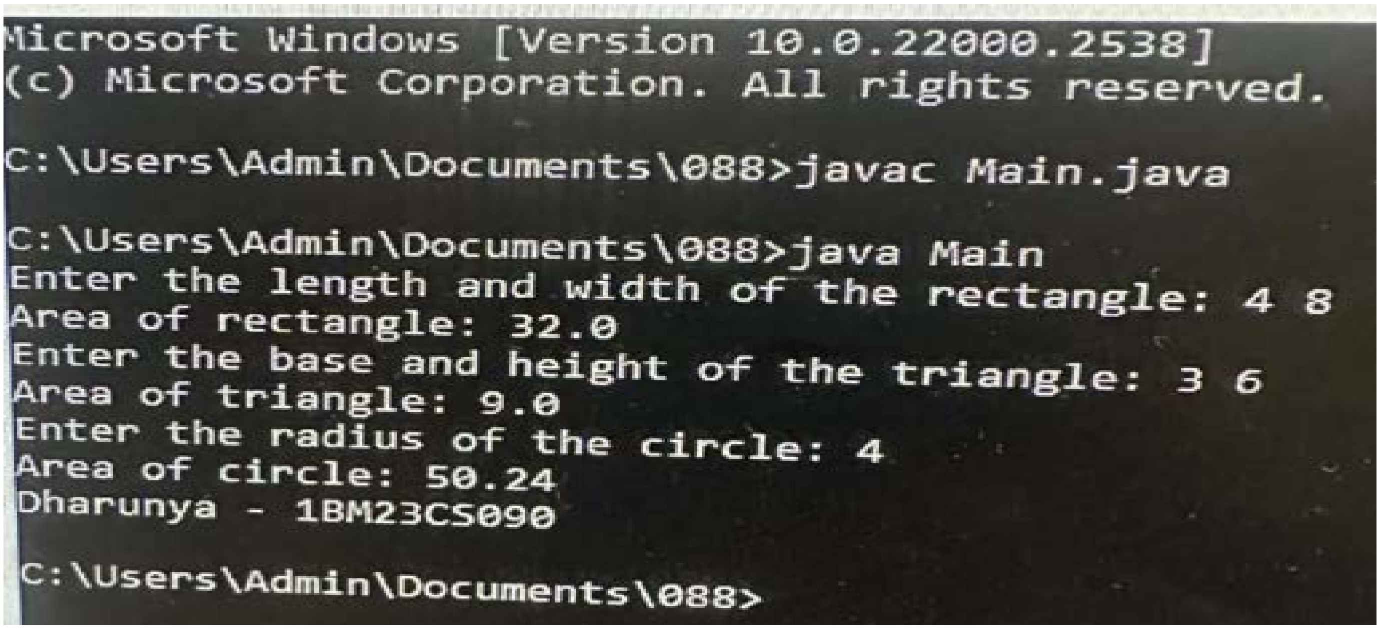
Output:

```
Enter the length and width of the rectangle: 4 8
Area of ~~circle~~ rectangle: 32.0
Enter the base and height of triangle: 3 6
Area of triangle: 9.0
Enter the radius of the circle: 4
Area of circle: 50.24
```

22.10

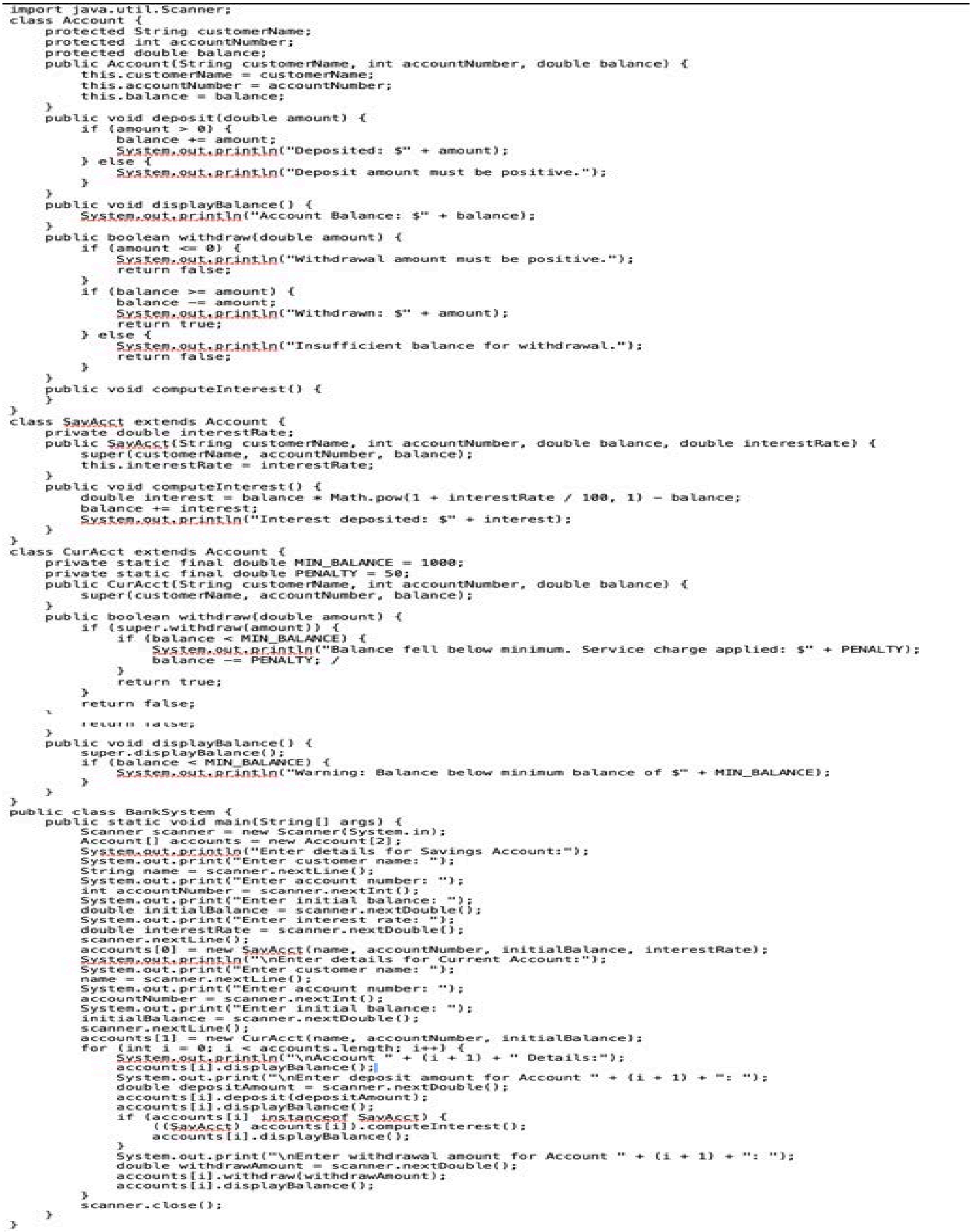
New code:

```
 Scanner sc = new Scanner (system.in)
 System.out.println(" Enter the length & breadth of rectangle:");
double rectangleLength = sc.nextDouble();
double rectangleBreadth = sc.nextDouble();
 shape s = new rectangle();
  s.printArea (rectangleLength, rectangleBreadth);
System.out.println(" Enter the base and height of triangle:");
double triangleBase = sc.nextDouble();
double triangleHeight = sc.nextDouble();
Shape t = new triangle();
 t.printArea(triangleBase triangleHeight);
System.out.println(" Enter the radius of circle : ");
double circleRadius= sc.nextDouble();
Shape c = new Circle();
 c.printArea (circleRadius,0);
```

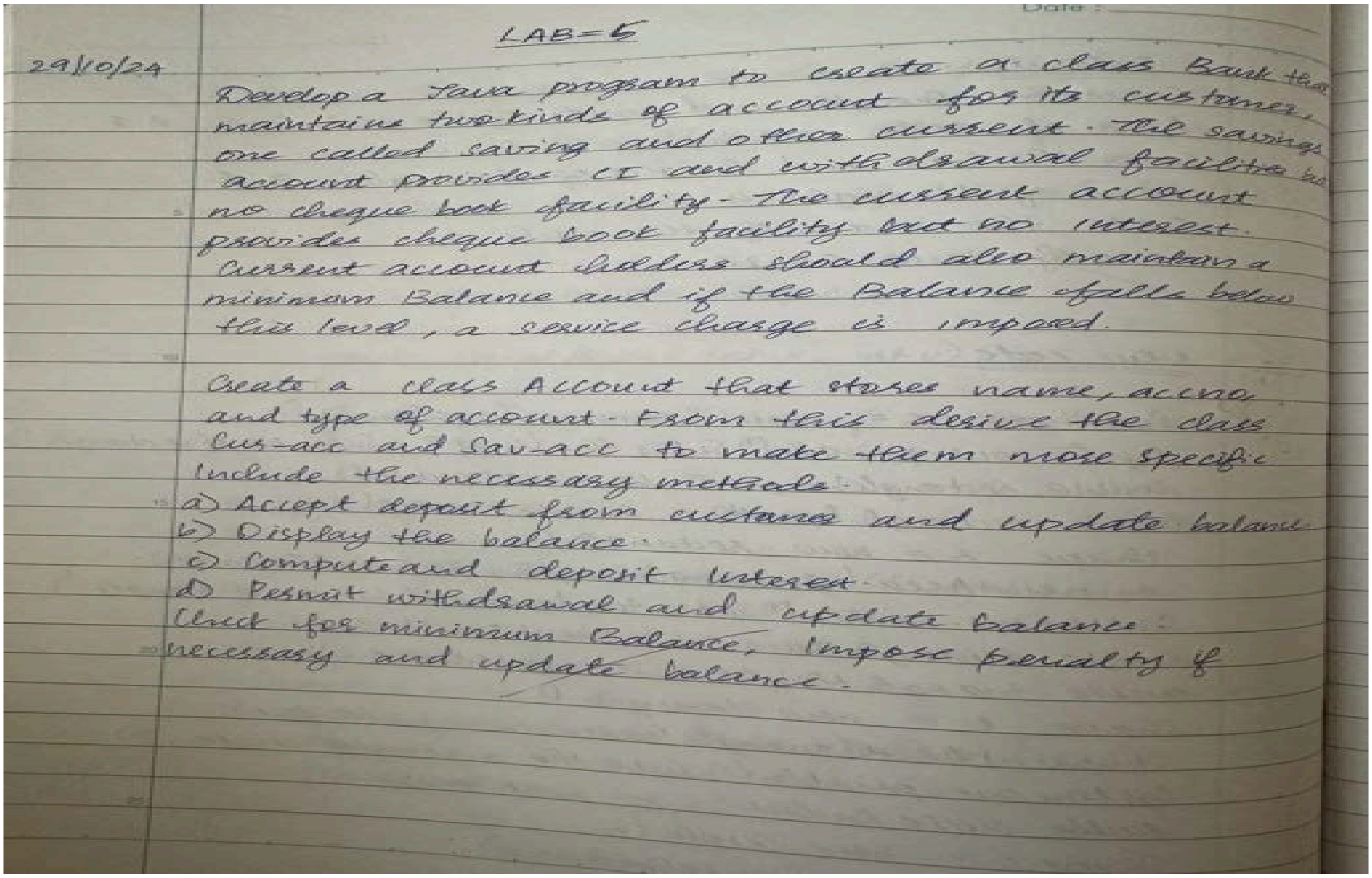


```
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Documents\088>javac Main.java

C:\Users\Admin\Documents\088>java Main
Enter the length and width of the rectangle: 4 8
Area of rectangle: 32.0
Enter the base and height of the triangle: 3 6
Area of triangle: 9.0
Enter the radius of the circle: 4
Area of circle: 50.24
Dharunya - 1BM23CS090

C:\Users\Admin\Documents\088>
```

5.Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks: a) Accept deposit from customer and update the balance. b) Display the balance. c) Compute and deposit interest d) Permit withdrawal and update the balance Check for the minimum balance, impose penalty if necessary and update the balance.



```
import java.util.Scanner;
class Account {
    protected String customerName;
    protected int accountNumber;
    protected double balance;
    public Account(String customerName, int accountNumber, double balance) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.balance = balance;
    }
    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited: $" + amount);
        } else {
            System.out.println("Deposit amount must be positive.");
        }
    }
    public void displayBalance() {
        System.out.println("Account Balance: $" + balance);
    }
    public boolean withdraw(double amount) {
        if (amount <= 0) {
            System.out.println("Withdrawal amount must be positive.");
            return false;
        }
        if (balance >= amount) {
            balance -= amount;
            System.out.println("Withdrawn: $" + amount);
            return true;
        } else {
            System.out.println("Insufficient balance for withdrawal.");
            return false;
        }
    }
    public void computeInterest() {
    }
}
class SavAcct extends Account {
    private double interestRate;
    public SavAcct(String customerName, int accountNumber, double balance, double interestRate) {
        super(customerName, accountNumber, balance);
        this.interestRate = interestRate;
    }
    public void computeInterest() {
        double interest = balance * Math.pow(1 + interestRate / 100, 1) - balance;
        balance += interest;
        System.out.println("Interest deposited: $" + interest);
    }
}
class CurAcct extends Account {
    private static final double MIN_BALANCE = 1000;
    private static final double PENALTY = 50;
    public CurAcct(String customerName, int accountNumber, double balance) {
        super(customerName, accountNumber, balance);
    }
    public boolean withdraw(double amount) {
        if (super.withdraw(amount)) {
            if (balance < MIN_BALANCE) {
                System.out.println("Balance fell below minimum. Service charge applied: $" + PENALTY);
                balance -= PENALTY; /
            }
            return true;
        }
        return false;
    }
        return false;
    }
    public void displayBalance() {
        super.displayBalance();
        if (balance < MIN_BALANCE) {
            System.out.println("Warning: Balance below minimum balance of $" + MIN_BALANCE);
        }
    }
}
public class BankSystem {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Account[] accounts = new Account[2];
        System.out.println("Enter details for Savings Account:");
        System.out.print("Enter customer name: ");
        String name = scanner.nextLine();
        System.out.print("Enter account number: ");
        int accountNumber = scanner.nextInt();
        System.out.print("Enter initial balance: ");
        double initialBalance = scanner.nextDouble();
        System.out.print("Enter interest rate: ");
        double interestRate = scanner.nextDouble();
        scanner.nextLine();
        accounts[0] = new SavAcct(name, accountNumber, initialBalance, interestRate);
        System.out.println("\nEnter details for Current Account:");
        System.out.print("Enter customer name: ");
        name = scanner.nextLine();
        System.out.print("Enter account number: ");
        accountNumber = scanner.nextInt();
        System.out.print("Enter initial balance: ");
        initialBalance = scanner.nextDouble();
        scanner.nextLine();
        accounts[1] = new CurAcct(name, accountNumber, initialBalance);
        for (int i = 0; i < accounts.length; i++) {
            System.out.println("\nAccount " + (i + 1) + " Details:");
            accounts[i].displayBalance();
            System.out.print("\nEnter deposit amount for Account " + (i + 1) + ": ");
            double depositAmount = scanner.nextDouble();
            accounts[i].deposit(depositAmount);
            accounts[i].displayBalance();
            if (accounts[i] instanceof SavAcct) {
                ((SavAcct) accounts[i]).computeInterest();
                accounts[i].displayBalance();
            }
            System.out.print("\nEnter withdrawal amount for Account " + (i + 1) + ": ");
            double withdrawAmount = scanner.nextDouble();
            accounts[i].withdraw(withdrawAmount);
            accounts[i].displayBalance();
        }
        scanner.close();
    }
}
```



Date :

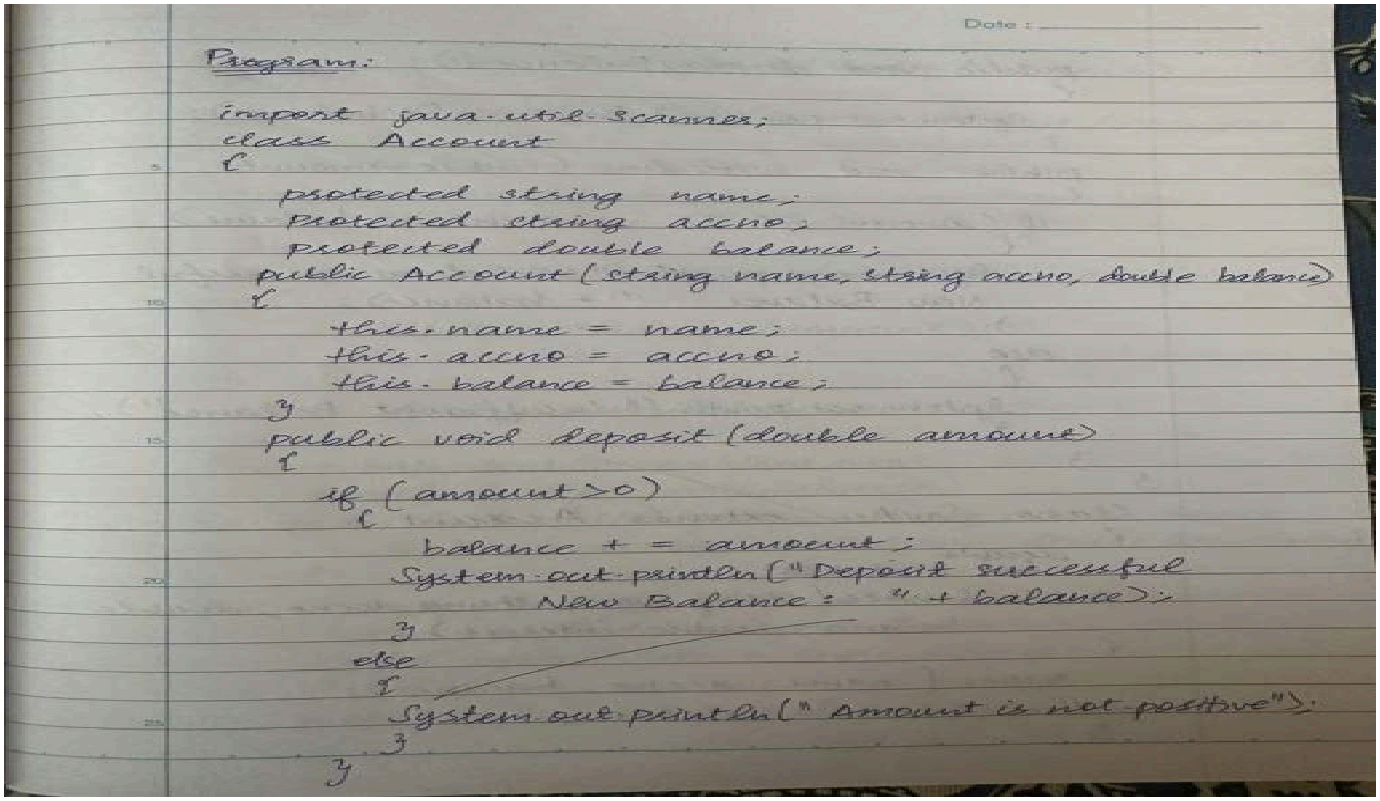
LAB-5

29/10/24

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called saving and other current. The savings account provides CI and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum Balance and if the Balance falls below this level, a service charge is imposed.

Create a class Account that stores name, accno and type of account. From this derive the class Cur-acc and Sav-acc to make them more specific. Include the necessary methods.

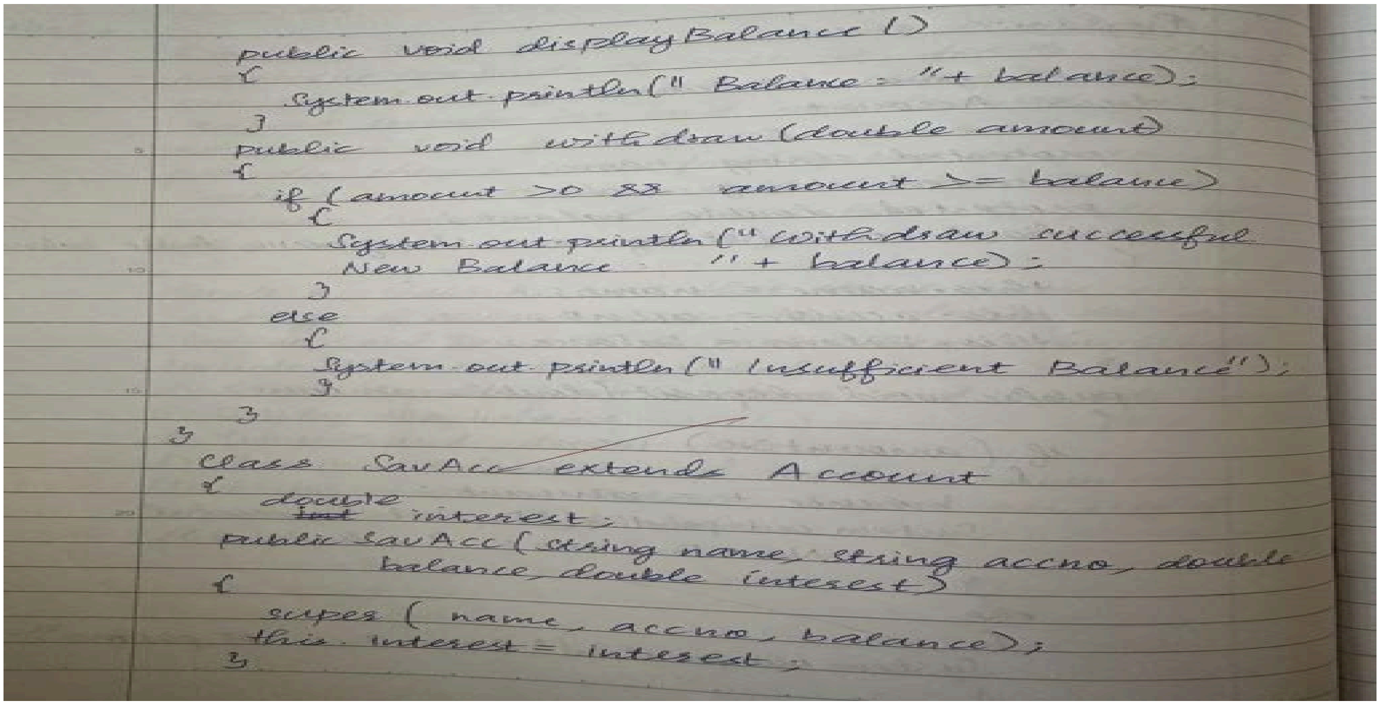
a) Accept deposit from customer and update balance
b) Display the balance.
c) Compute and deposit interest.
d) Permit withdrawal and update balance.
Check for minimum Balance, Impose penalty if necessary and update balance.



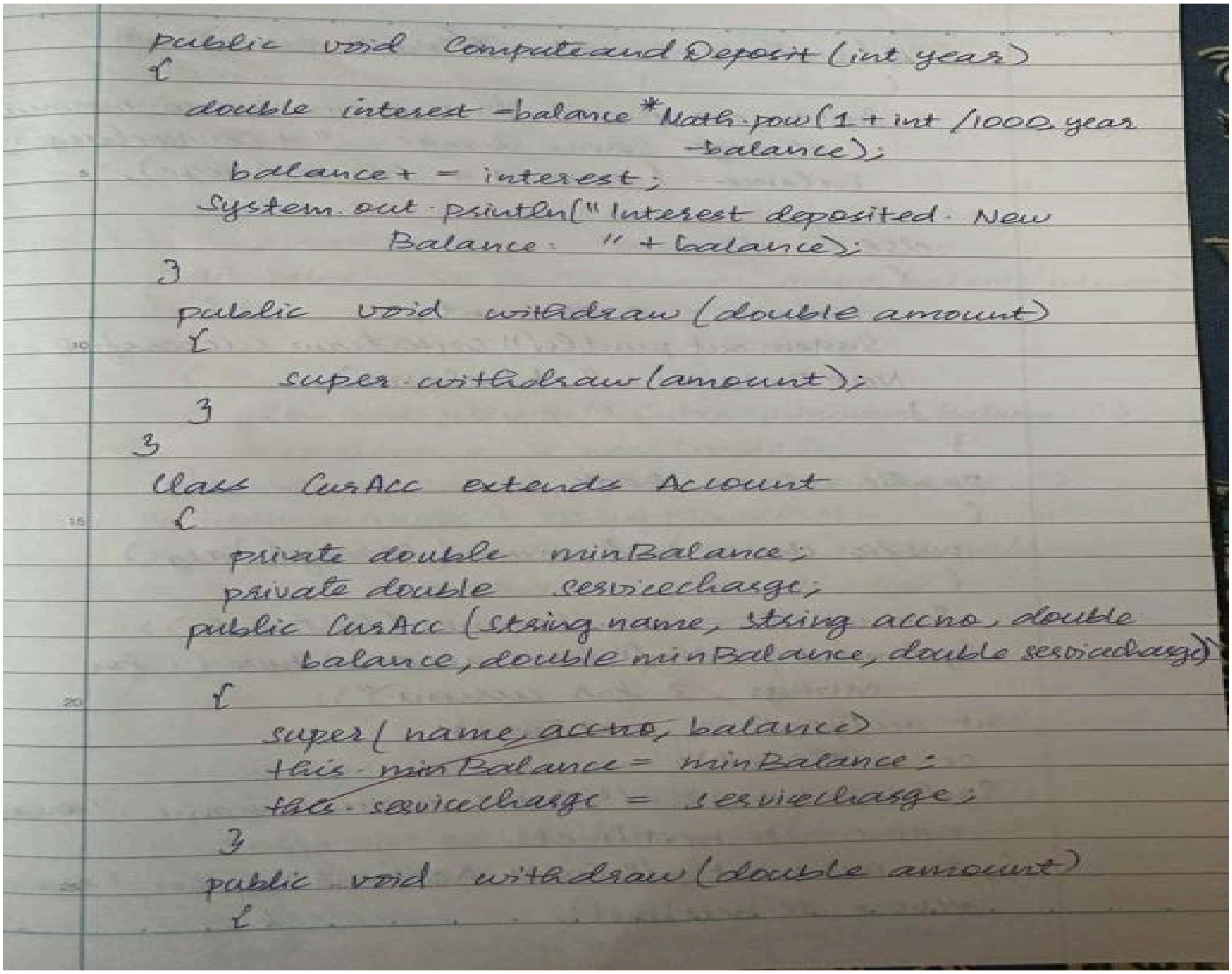
Date :

Program:

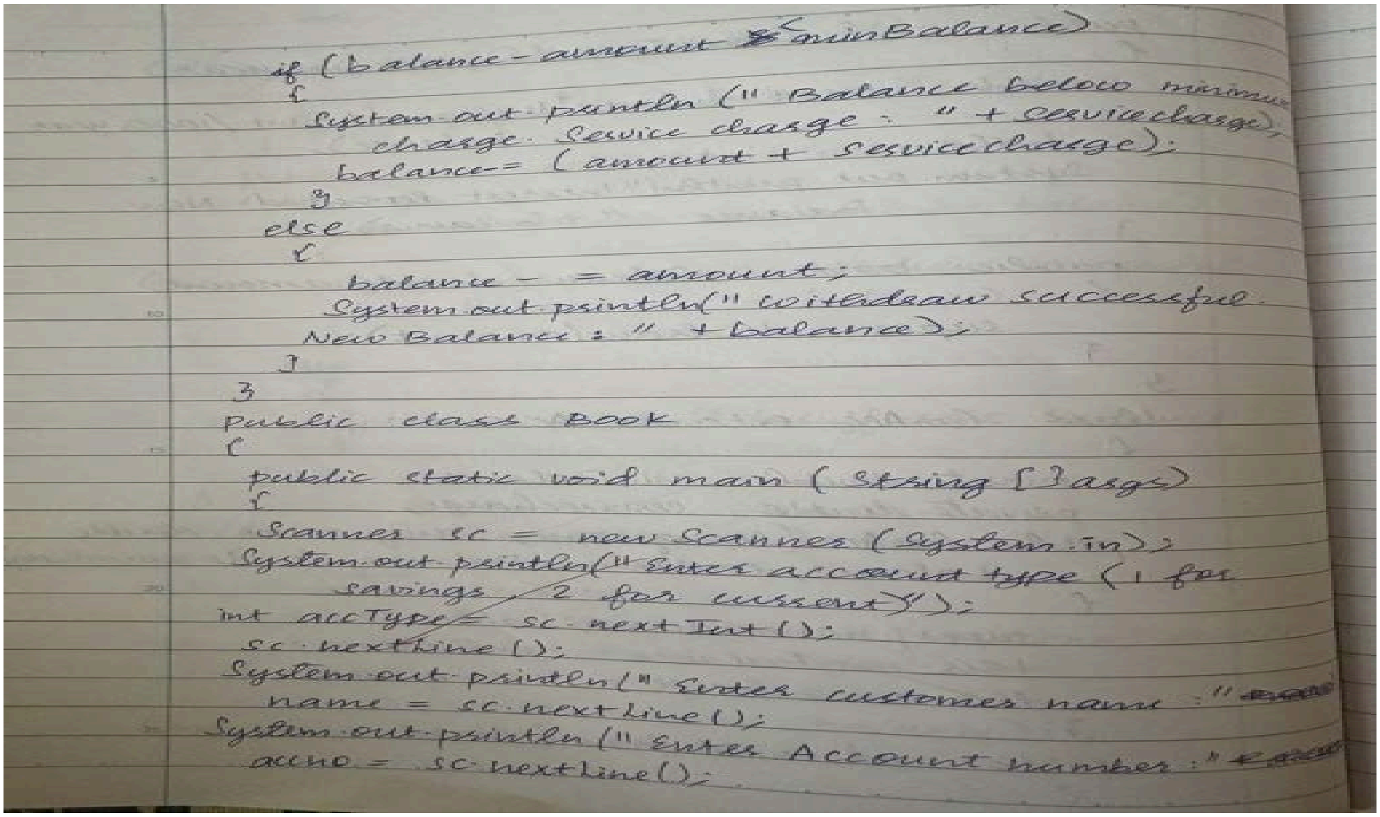
```
import java.util.scanner;
class Account
{
   protected string name;
   protected string accno;
   protected double balance;
  public Account(string name, string accno, double balance)
  {
      this.name = name;
      this.accno = accno;
      this.balance = balance;
  }
  public void deposit(double amount)
  {
     if (amount>0)
      {
         balance + = amount;
        System.out.println("Deposit successful
               New Balance: " + balance);
      }
     else
      {
      System.out.println(" Amount is not positive");
      }
  }
```



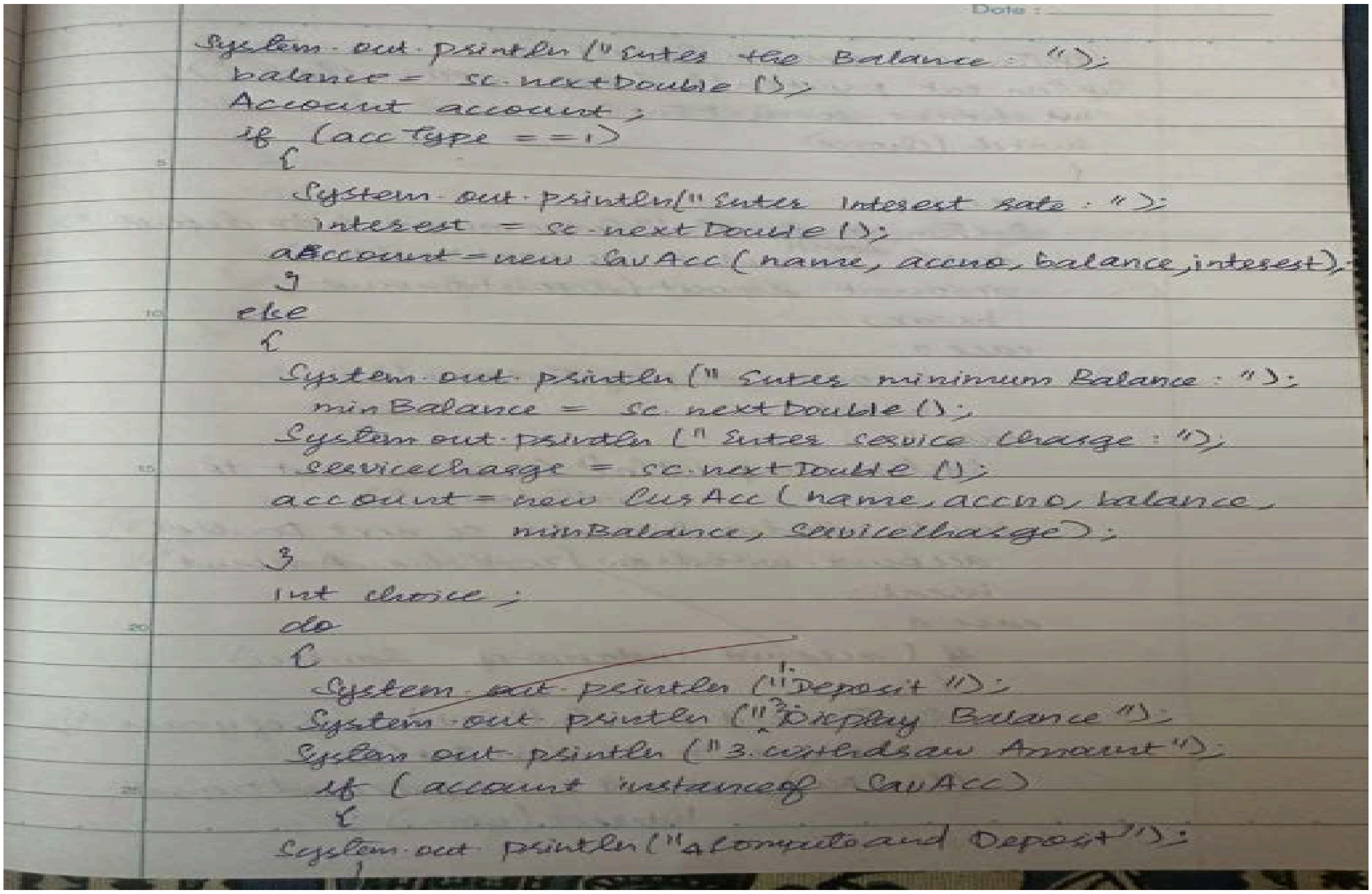
```
public void displayBalance ()
{
  System.out.println(" Balance: "+ balance);
 }
public void withdraw (double amount)
{
  if (amount >0 && amount >= balance)
   {
    System.out.println(" Withdraw successful
    New Balance  "+ balance);
   }
  else
   {
   System.out.println(" Insufficient Balance");
   }
 }
}
Class SavAcc extends Account
{ double
  ~~int~~ interest;
public savAcc ( string name, string accno, double
       balance, double interest)
{
  super ( name, accno, balance);
 this. interest = interest;
}
```



```
public void Computeand Deposit (int year)
{
 double interest -balance *Math.pow(1 + int /1000, year
                                -balance);
   bdlance+ = interest;
  System.out.println(" Interest deposited. New
            Balance:  " + balance);
 }
 public void withdraw (double amount)
  {
      super.withdraw(amount);
  }
}
Class CurAcc extends Account
 {
   private double minBalance;
   private double servicecharge;
 public CurAcc (string name, string accno, double
       balance, double minBalance, double servicecharge))
  {
    super( name, accno, balance)
    this.minBalance = minBalance;
    this.servicecharge = servicecharge;
  }
 public void withdraw (double amount)
  {
```

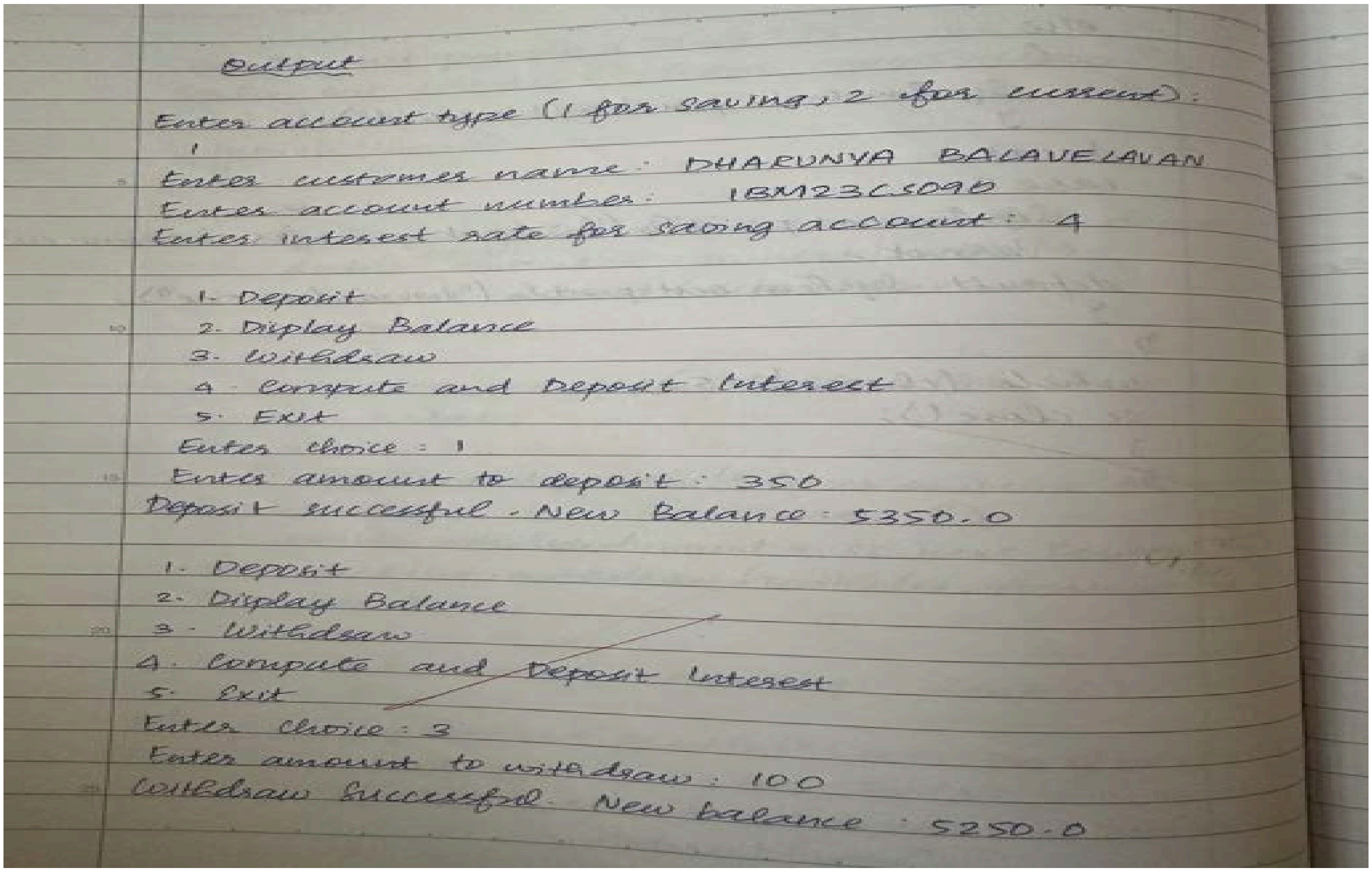


```
if (balance - amount ~~>~~ < minBalance)
 {
  System.out.println (" Balance below minimum
     charge. Service charge : " + servicecharge);
  balance-= (amount + servicecharge);
  }
else
 {
   balance - = amount;
  System.out.println(" withdraw successful.
  New Balance : " + balance);
  }
 }
public class BOOK
{
 public static void main ( String []args)
 {
 Scanner sc = new Scanner (System.in);
 System.out.println(" Enter account type (1 for
     savings, 2 for current)");
int accType = sc.nextInt();
 sc.nextLine();
System.out.println(" Enter customer name :"
  name = sc.nextLine();
System.out.println(" Enter Account number :"
  accno = sc.nextLine();
```



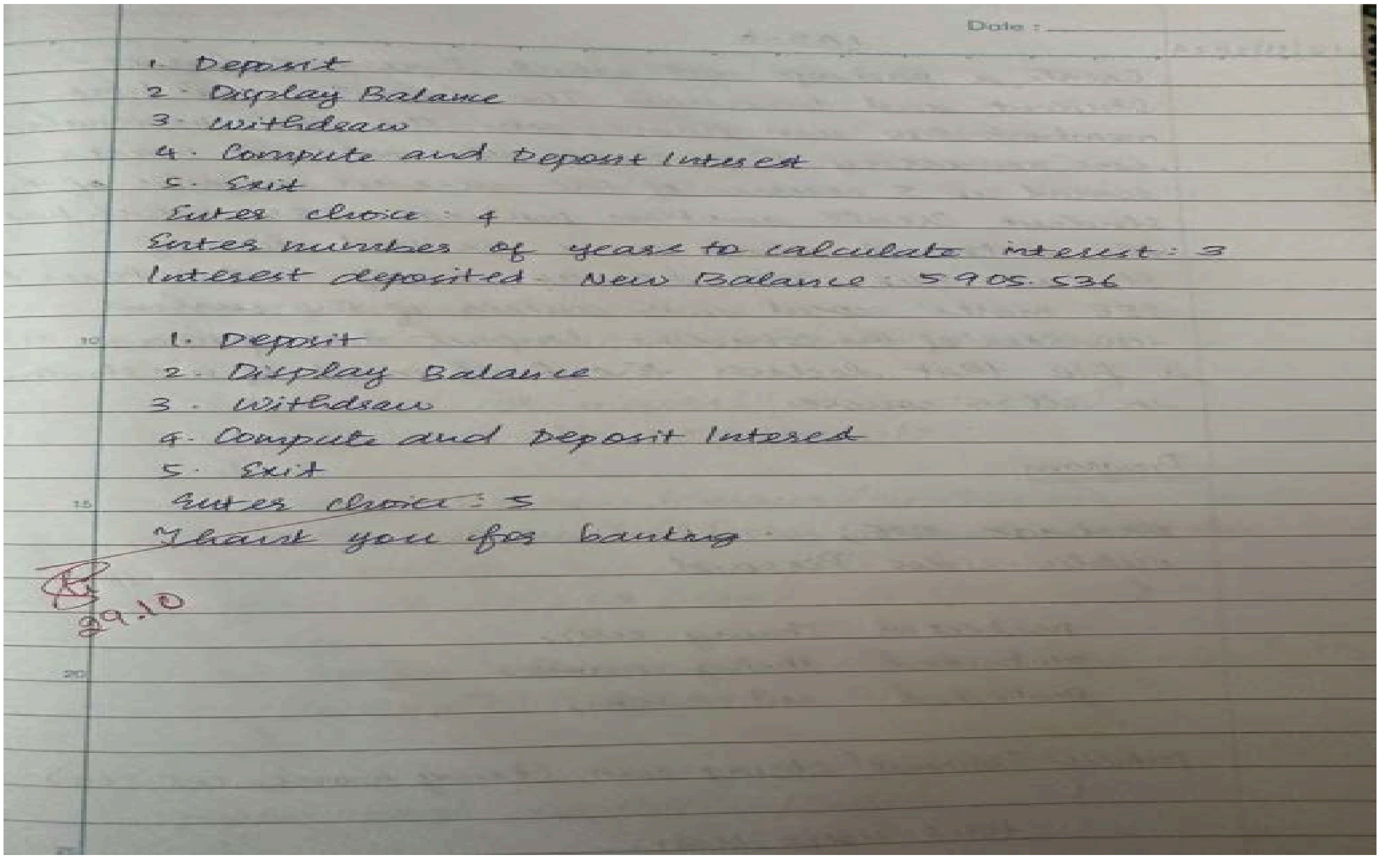
```
System.out.println("Enter the Balance : ");
 balance = sc.nextDouble ();
 Account account ;
 if (accType ==1)
  {
   System.out.println("Enter Interest rate : ");
    interest = sc.nextDouble();
  account = new savAcc(name, accno, balance, interest);
  }
 else
  {
   System.out.println (" Enter minimum Balance : ");
    minBalance = sc.nextDouble();
   System.out.println (" Enter service Charge : ");
    serviceCharge = sc.nextDouble ();
   account = new CurAcc(name, accno, balance,
              minBalance, serviceCharge);
   }
   int choice ;
   do
   {
    System.out.println ("1.Deposit ");
    System.out.println ("2.Display Balance ");
    System.out.println ("3.withdraw Amount ");
    if (account instanceof savAcc)
     {
   System.out.println("4.Compute and Deposit");
System.out.println ("5.Exit");
System.out.println (" Enter your choice:");
 int choice= sc.nextInt();
 switch (choice)
 {
   case 1:
   System.out.println(" Enter amount to deposit :");
   double depositamount = sc.nextDouble();
   account.deposit (depositamount);
    break;
   case 2:
     account.display Balance();
     break;
   case 3:
    System.out.println (" Enter amount to
                    withdraw:");
   double withdrawAmount = sc.next Double();
    account.withdraw (withdrawAmount);
    break;
   case 4:
     if (account instance of  SavAcc)
     {
      System.out.println (" Enter no.of years :");
         int year = sc.next Int();
     ((SavAcc) account. computeandDeposit
                   Interest (year);
     }
   else
     {
       System.out.println (" Invalid");
      }
   break;
   case 5:
     System.out.println (" Thank you for choosing us"
       break;
   default: System.out.println ("Invalid choice");
   }
 }
 while (choice != 5);
 sc.close();
  }
 }
```

29.10



Output

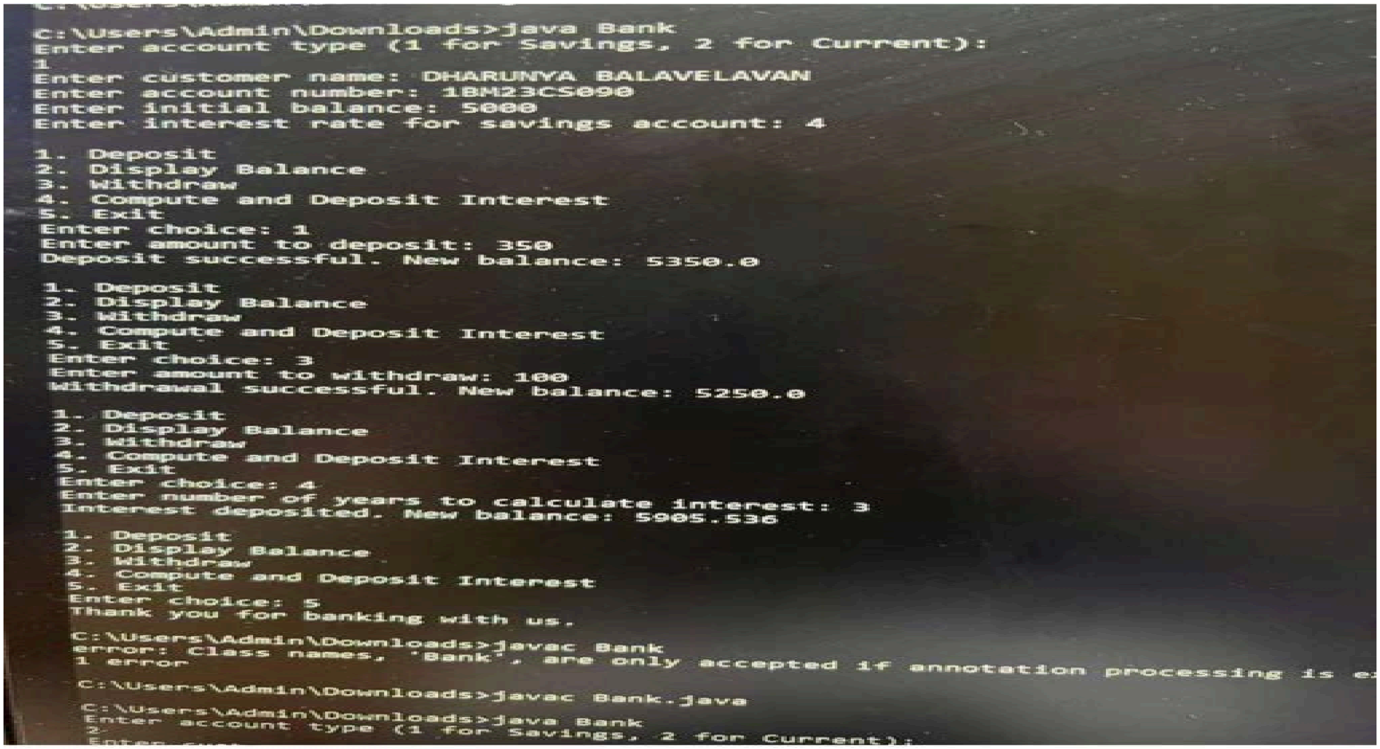
```
Enter account type (1 for saving, 2 for current):
 1
Enter customer name: DHARUNYA BALAVELAVAN
Enter account number: 1BM23CS090
Enter interest rate for saving account: 4

 1. Deposit
 2. Diplay Balance
 3. Withdraw
 4. Compute and Deposit Interest
 5. Exit
 Enter choice: 1
 Enter amount to deposit: 350
Deposit successful. New Balance: 5350.0

 1. Deposit
 2. Display Balance
 3. Withdraw
4. Compute and Deposit Interest
5. Exit
Enter choice: 3
Enter amount to withdraw: 100
Withdraw successful. New balance: 5250.0
```



Date :

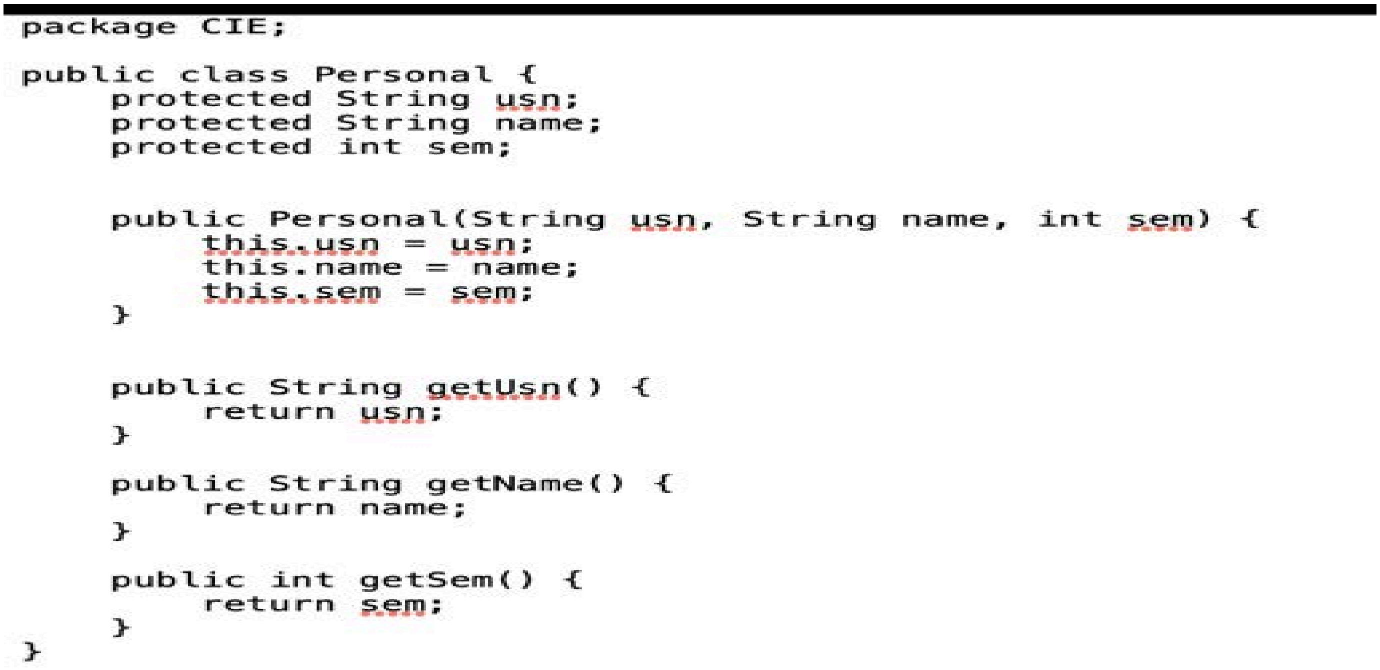
```
 1. Deposit
 2. Display Balance
 3. Withdraw
 4. Compute and Deposit Interest
 5. Exit
 Enter choice: 4
Enter number of years to calculate interest: 3
Interest deposited. New Balance: 5905.536

 1. Deposit
 2. Display Balance
 3. Withdraw
 4. Compute and Deposit Interest
 5. Exit
 Enter choice: 5
Thank you for banking.
```

29.10

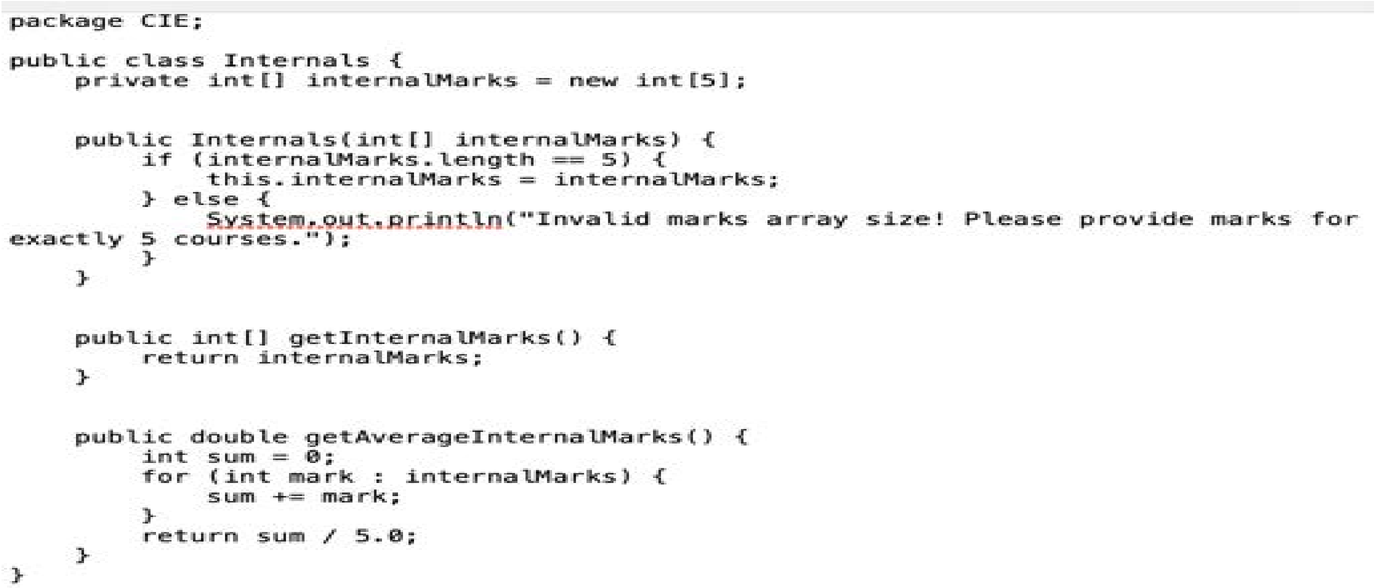


```
C:\Users\Admin\Downloads>java Bank
Enter account type (1 for Savings, 2 for Current):
1
Enter customer name: DHARUNYA BALAVELAVAN
Enter account number: 1BM23CS090
Enter initial balance: 5000
Enter interest rate for savings account: 4

1. Deposit
2. Display Balance
3. Withdraw
4. Compute and Deposit Interest
5. Exit
Enter choice: 1
Enter amount to deposit: 350
Deposit successful. New balance: 5350.0

1. Deposit
2. Display Balance
3. Withdraw
4. Compute and Deposit Interest
5. Exit
Enter choice: 3
Enter amount to withdraw: 100
Withdrawal successful. New balance: 5250.0

1. Deposit
2. Display Balance
3. Withdraw
4. Compute and Deposit Interest
5. Exit
Enter choice: 4
Enter number of years to calculate interest: 3
Interest deposited. New balance: 5905.536

1. Deposit
2. Display Balance
3. Withdraw
4. Compute and Deposit Interest
5. Exit
Enter choice: 5
Thank you for banking with us.

C:\Users\Admin\Downloads>javac Bank
error: Class names, 'Bank', are only accepted if annotation processing is e
1 error

C:\Users\Admin\Downloads>javac Bank.java

C:\Users\Admin\Downloads>java Bank
Enter account type (1 for Savings, 2 for Current):
2
```

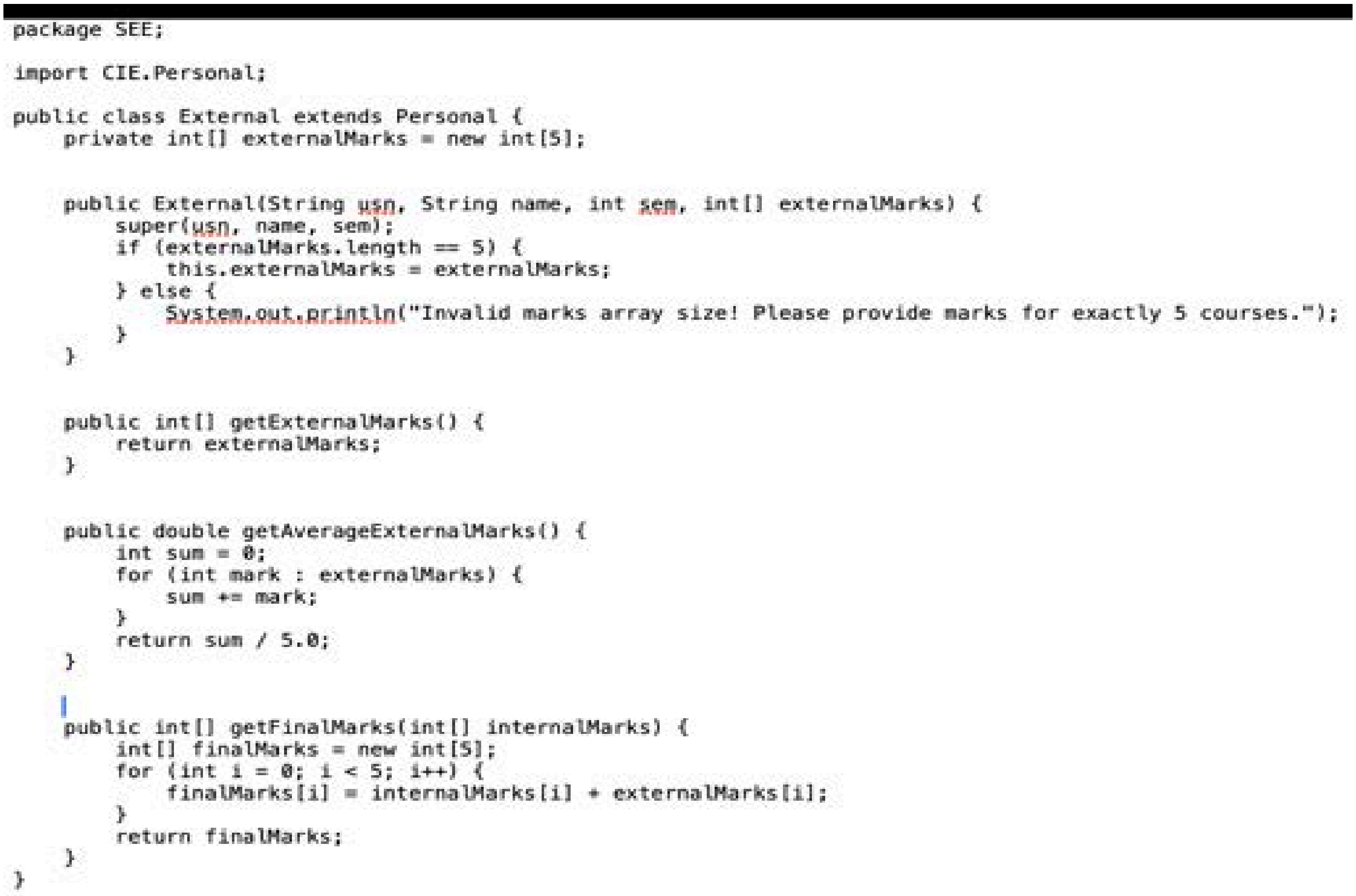
6.Create a package CIE which has two classes - Personal and Internals. The class Personal has members like usn, name, sem. The class Internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Personal. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.



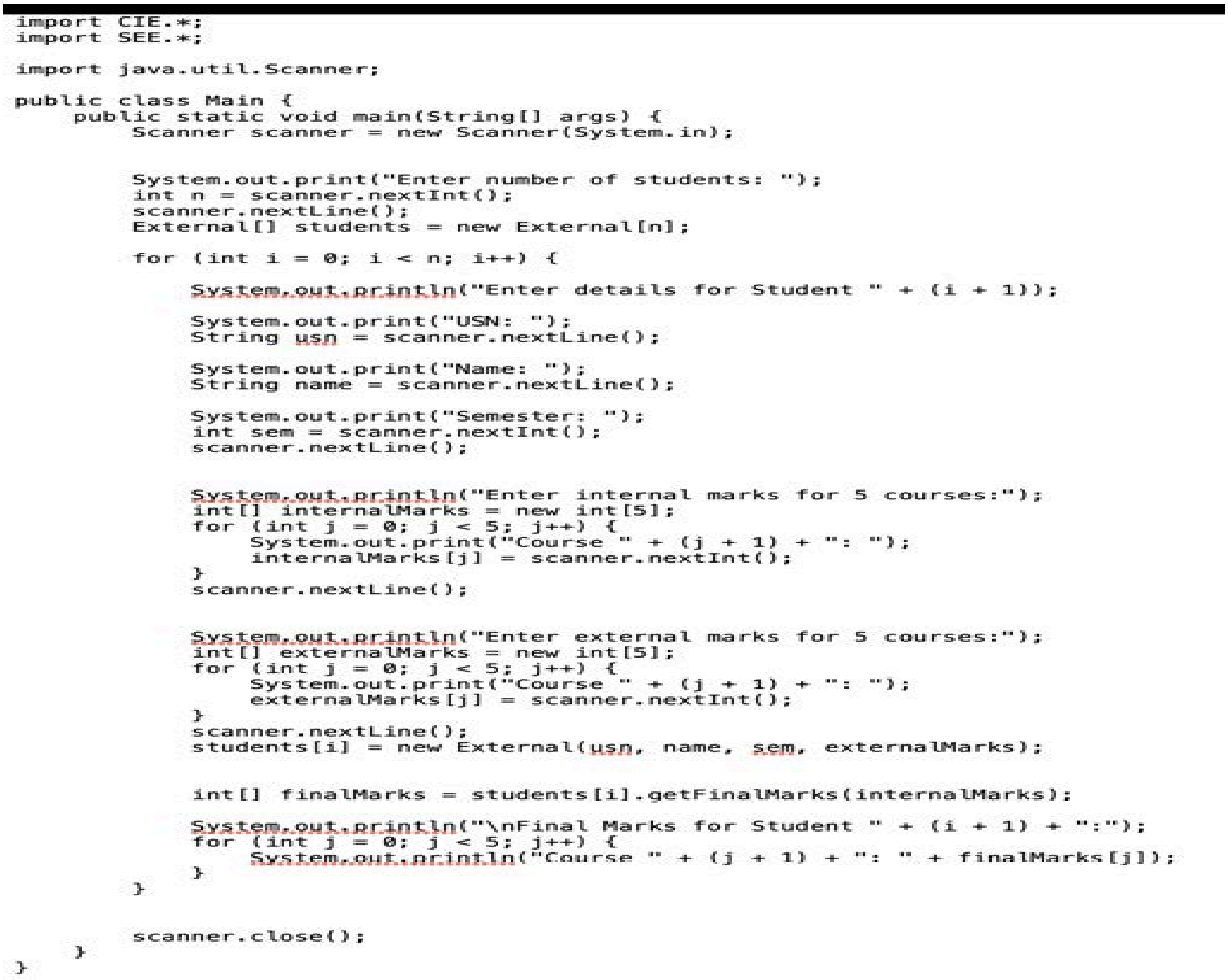
```
package CIE;

public class Personal {
    protected String usn;
    protected String name;
    protected int sem;


    public Personal(String usn, String name, int sem) {
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }


    public String getUsn() {
        return usn;
    }

    public String getName() {
        return name;
    }

    public int getSem() {
        return sem;
    }
}
```



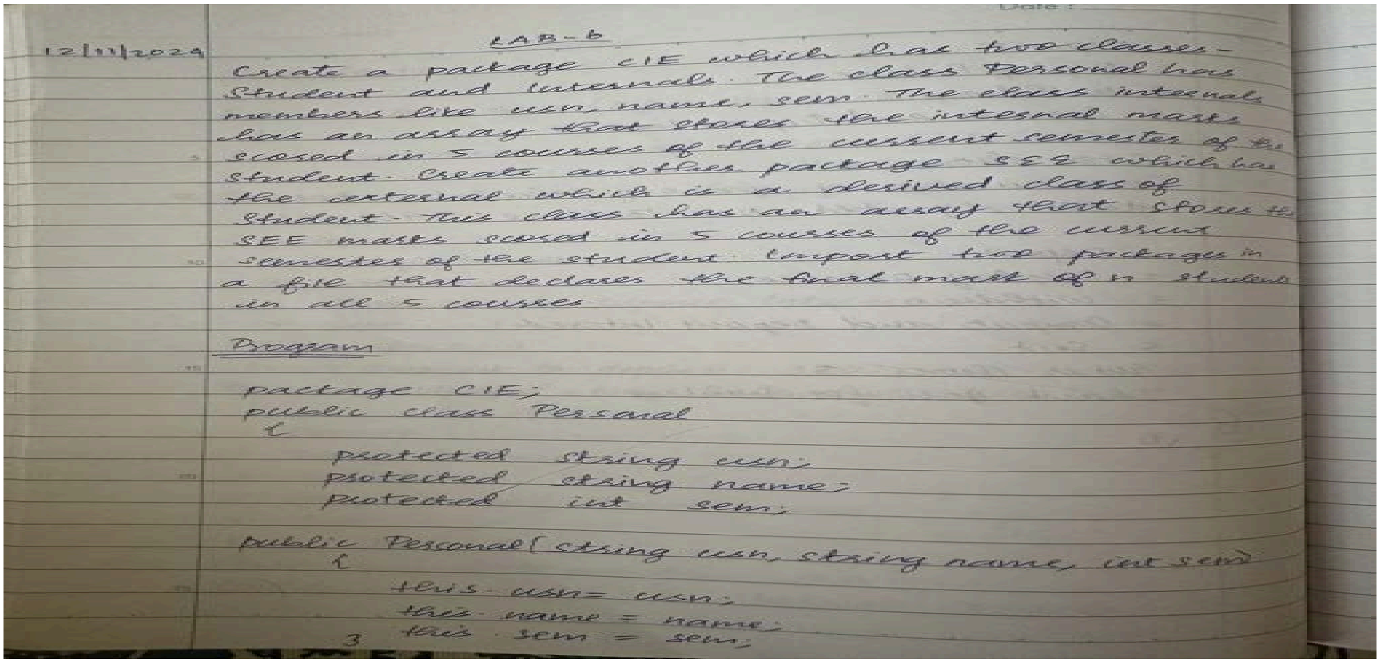
```java
package CIE;

public class Internals {
    private int[] internalMarks = new int[5];


    public Internals(int[] internalMarks) {
        if (internalMarks.length == 5) {
            this.internalMarks = internalMarks;
        } else {
            System.out.println("Invalid marks array size! Please provide marks for
exactly 5 courses.");
        }
    }


    public int[] getInternalMarks() {
        return internalMarks;
    }


    public double getAverageInternalMarks() {
        int sum = 0;
        for (int mark : internalMarks) {
            sum += mark;
        }
        return sum / 5.0;
    }
}
```



```java
package SEE;

import CIE.Personal;

public class External extends Personal {
    private int[] externalMarks = new int[5];


    public External(String usn, String name, int sem, int[] externalMarks) {
        super(usn, name, sem);
        if (externalMarks.length == 5) {
            this.externalMarks = externalMarks;
        } else {
            System.out.println("Invalid marks array size! Please provide marks for exactly 5 courses.");
        }
    }


    public int[] getExternalMarks() {
        return externalMarks;
    }


    public double getAverageExternalMarks() {
        int sum = 0;
        for (int mark : externalMarks) {
            sum += mark;
        }
        return sum / 5.0;
    }


    public int[] getFinalMarks(int[] internalMarks) {
        int[] finalMarks = new int[5];
        for (int i = 0; i < 5; i++) {
            finalMarks[i] = internalMarks[i] + externalMarks[i];
        }
        return finalMarks;
    }
}
```



```
import CIE.*;
import SEE.*;

import java.util.Scanner;

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

        System.out.print("Enter number of students: ");
        int n = scanner.nextInt();
        scanner.nextLine();
        External[] students = new External[n];

        for (int i = 0; i < n; i++) {

            System.out.println("Enter details for Student " + (i + 1));

            System.out.print("USN: ");
            String usn = scanner.nextLine();

            System.out.print("Name: ");
            String name = scanner.nextLine();

            System.out.print("Semester: ");
            int sem = scanner.nextInt();
            scanner.nextLine();

            System.out.println("Enter internal marks for 5 courses:");
            int[] internalMarks = new int[5];
            for (int j = 0; j < 5; j++) {
                System.out.print("Course " + (j + 1) + ": ");
                internalMarks[j] = scanner.nextInt();
            }
            scanner.nextLine();

            System.out.println("Enter external marks for 5 courses:");
            int[] externalMarks = new int[5];
            for (int j = 0; j < 5; j++) {
                System.out.print("Course " + (j + 1) + ": ");
                externalMarks[j] = scanner.nextInt();
            }
            scanner.nextLine();
            students[i] = new External(usn, name, sem, externalMarks);

            int[] finalMarks = students[i].getFinalMarks(internalMarks);

            System.out.println("\nFinal Marks for Student " + (i + 1) + ":");
            for (int j = 0; j < 5; j++) {
                System.out.println("Course " + (j + 1) + ": " + finalMarks[j]);
            }
        }

        scanner.close();
    }
}
```



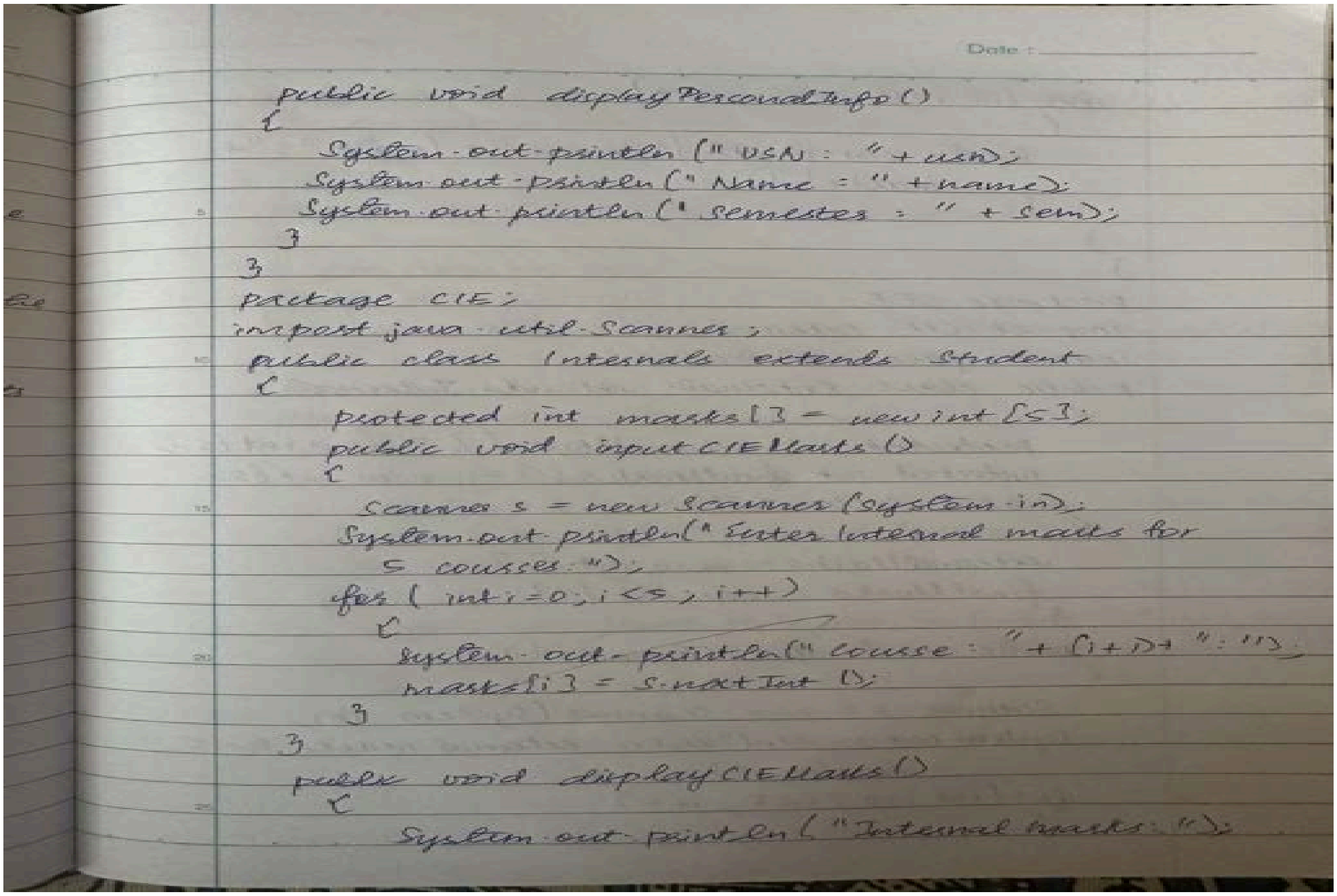
12/11/2024

LAB-6

Create a package CIE which has two classes - Student and Internals. The class Personal has members like usn, name, sem. The class internals has an array that stores the internal marks scored in 5 courses of the current semester of the student. Create another package SEE which has the external which is a derived class of Student. This class has an array that stores the SEE marks scored in 5 courses of the current semester of the student. Import two packages in a file that declares the final mark of n students in all 5 courses

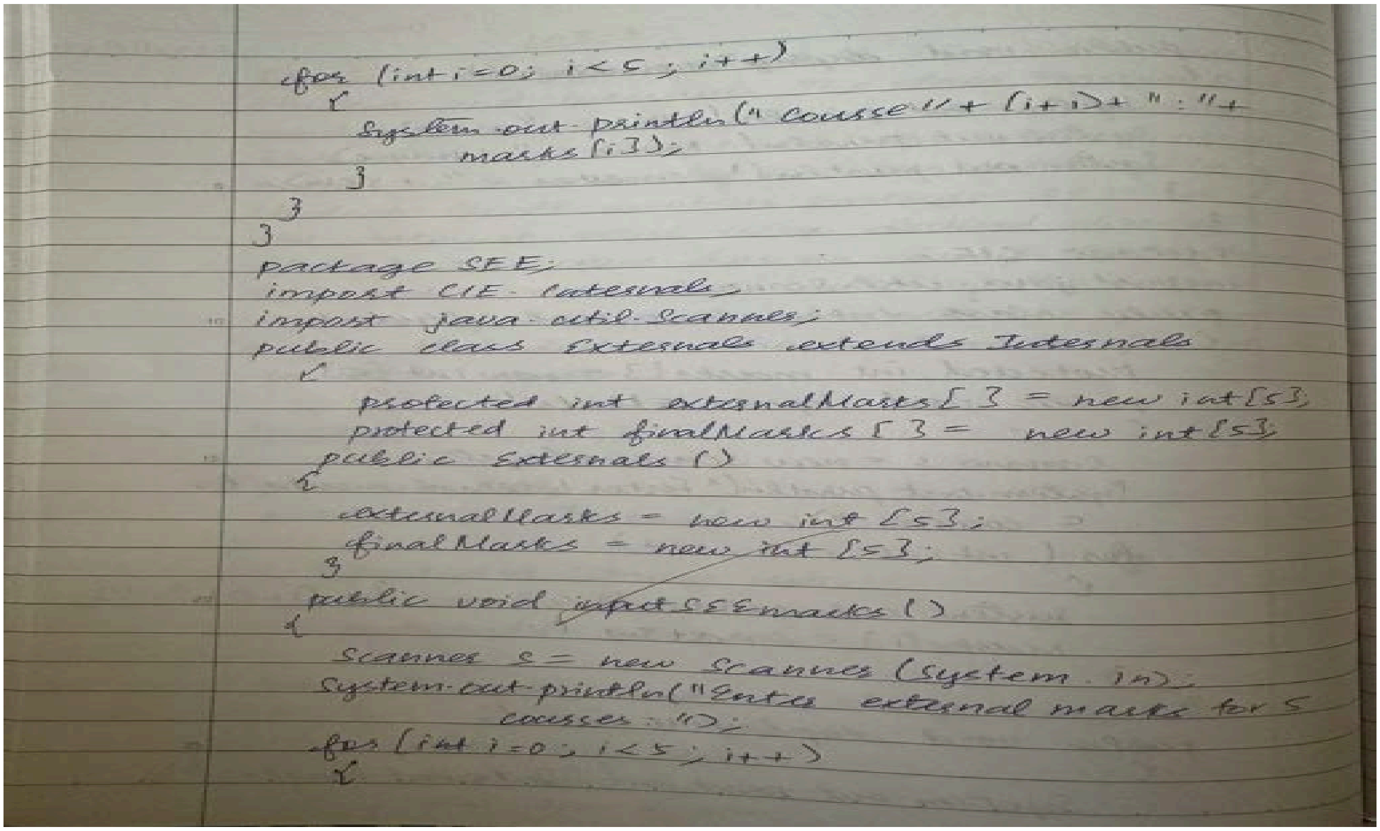
Program

```
package CIE;
public class Personal
 {
      protected string usn;
      protected string name;
      protected int sem;

public Personal(string usn, string name, int sem)
   {
        this.usn= usn;
        this.name = name;
    }   this.sem = sem;
```

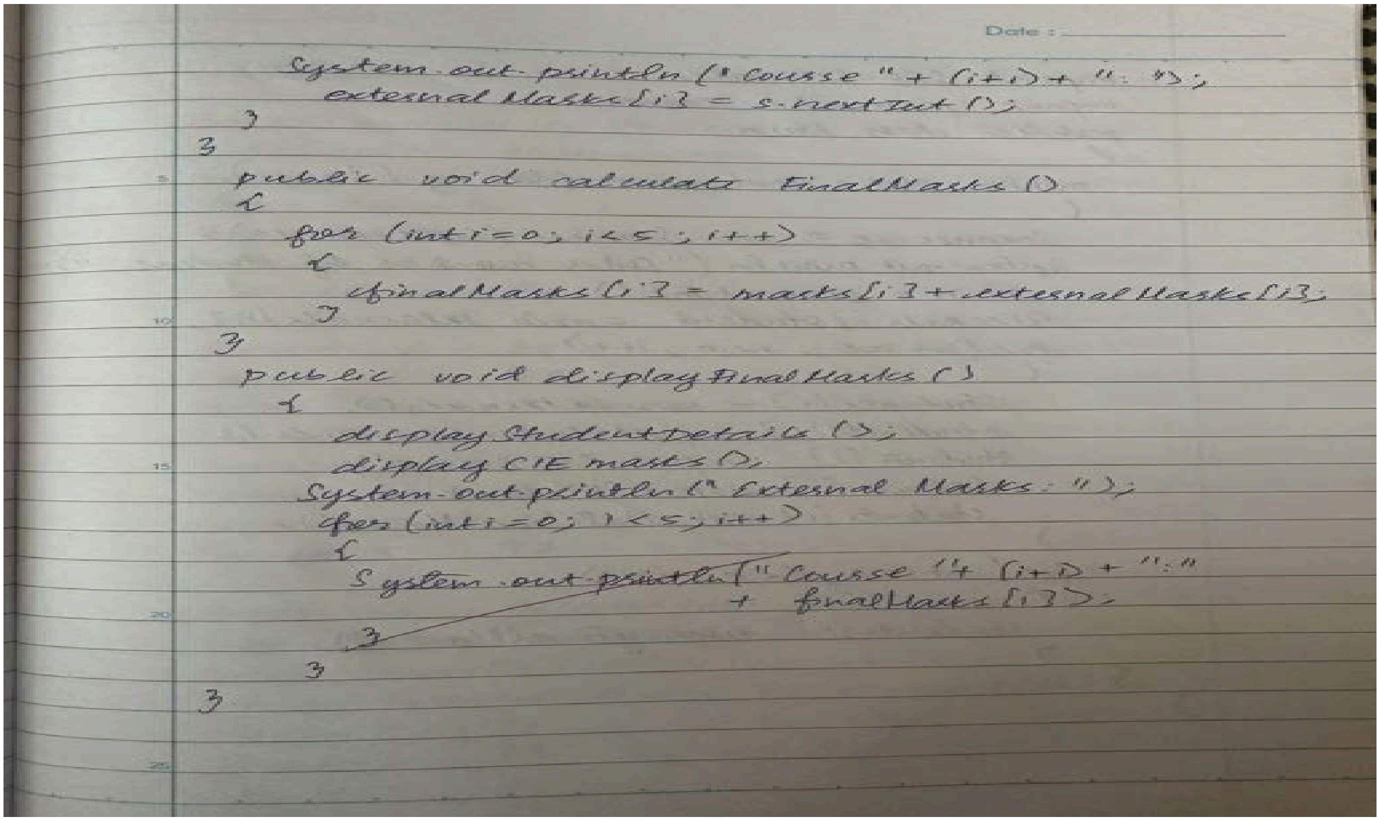


Date :

```
public void displayPersonalInfo()
{
   System.out.println(" USN : " + usn);
   System.out.println(" Name : " + name);
   System.out.println(" Semester : " + sem);
  }
}
package CIE;
import java.util.Scanner;
public class Internals extends Student
 {
     protected int marks[] = new int [5];
     public void inputCIEMarks ()
     {
       Scanner s = new Scanner (System.in);
      System.out.println(" Enter Internal marks for
         5 courses:");
     for ( int i=0; i<5; i++)
       {
         System.out.println(" course : " + (i+1)+ ":");
         marks[i] = s.nextInt ();
       }
   }
   public void displayCIEMarks()
     {
        System.out.println("Internal marks:");
```

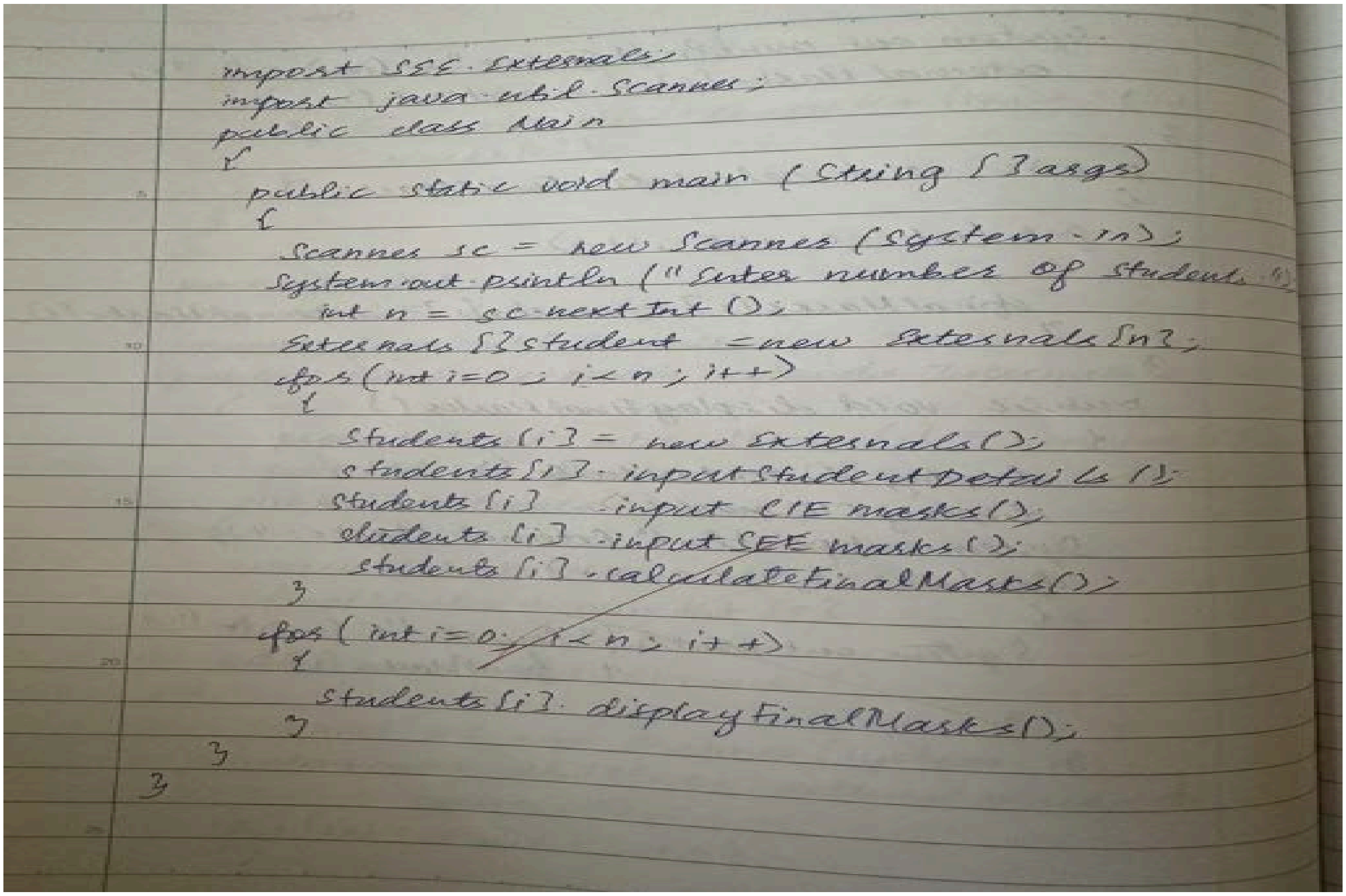


```
for (int i=0; i<5; i++)
  {
    System.out.println(" course " + (i+1)+ " : "+
            marks[i]);
   }
 }
}
package SEE;
import CIE.Internals;
import java.util.Scanner;
public class Externals extends Internals
  {
     protected int externalMarks[ ] = new int[5];
     protected int finalMarks [ ] =  new int[5];
   public Externals ()
  {
    externalMarks = new int [5];
    finalMarks = new int [5];
   }
  public void inputSEEmarks ()
 {
   Scanner s = new Scanner (System. in);
  System.out.println("Enter external marks for 5
            courses :");
  for (int i=0; i<5; i++)
   {
```

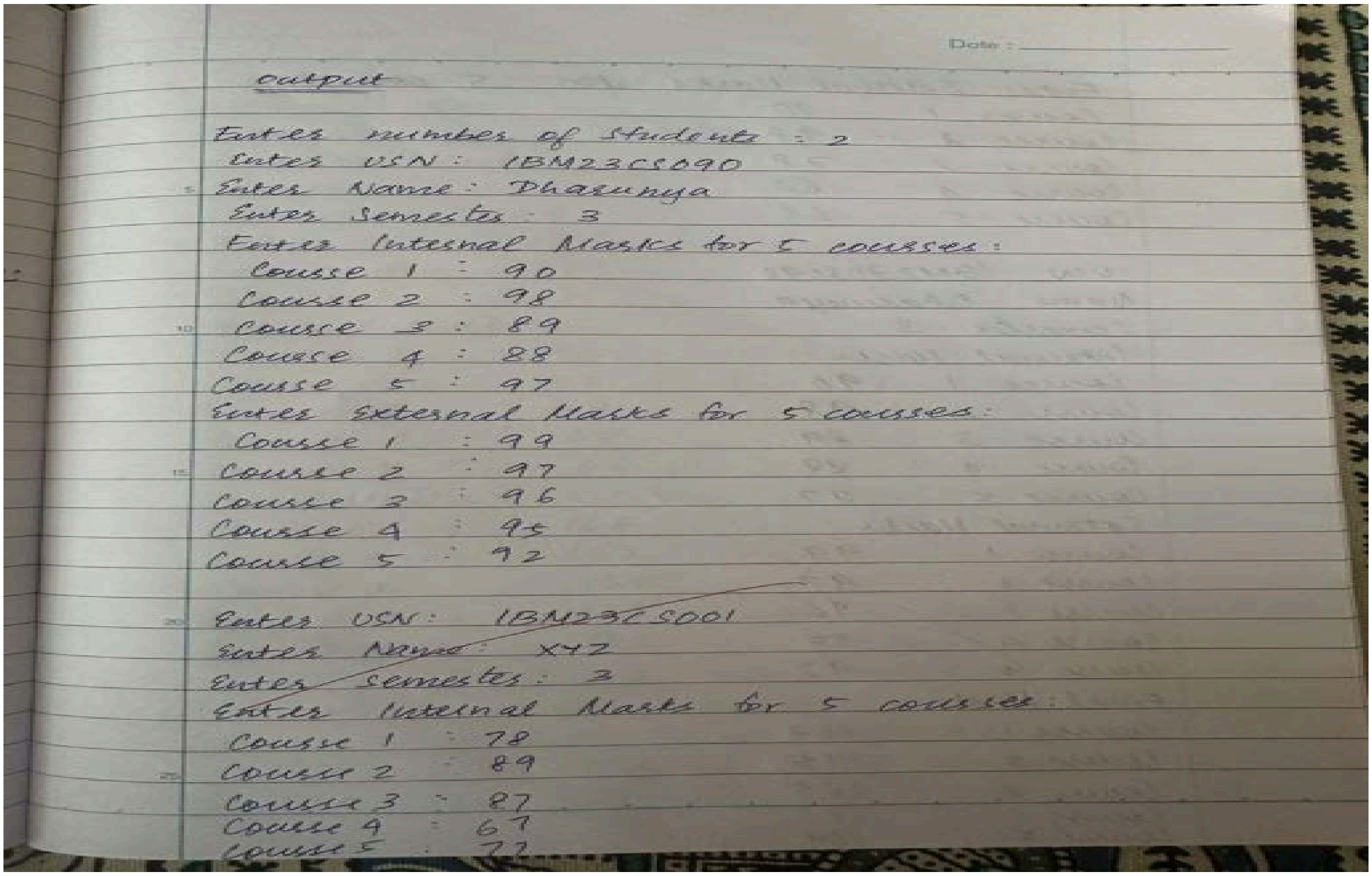


Date :

```
    System.out.println("Course " + (i+1) + ": ");
      externalMarks[i] = s.nextInt();
   }
 }
  public void calculate FinalMarks()
  {
    for (int i=0; i<5; i++)
     {
       finalMarks[i] = marks[i] + externalMarks[i];
     }
 }
  public void displayFinalMarks()
   {
     displayStudentDetails();
     displayCIE marks();
    System.out.println(" External Marks: ");
     for (int i=0; i<5; i++)
     {
       System.out.println(" Course "+ (i+1) + ":"
                          + finalMarks[i]);
       }
    }
 }
```



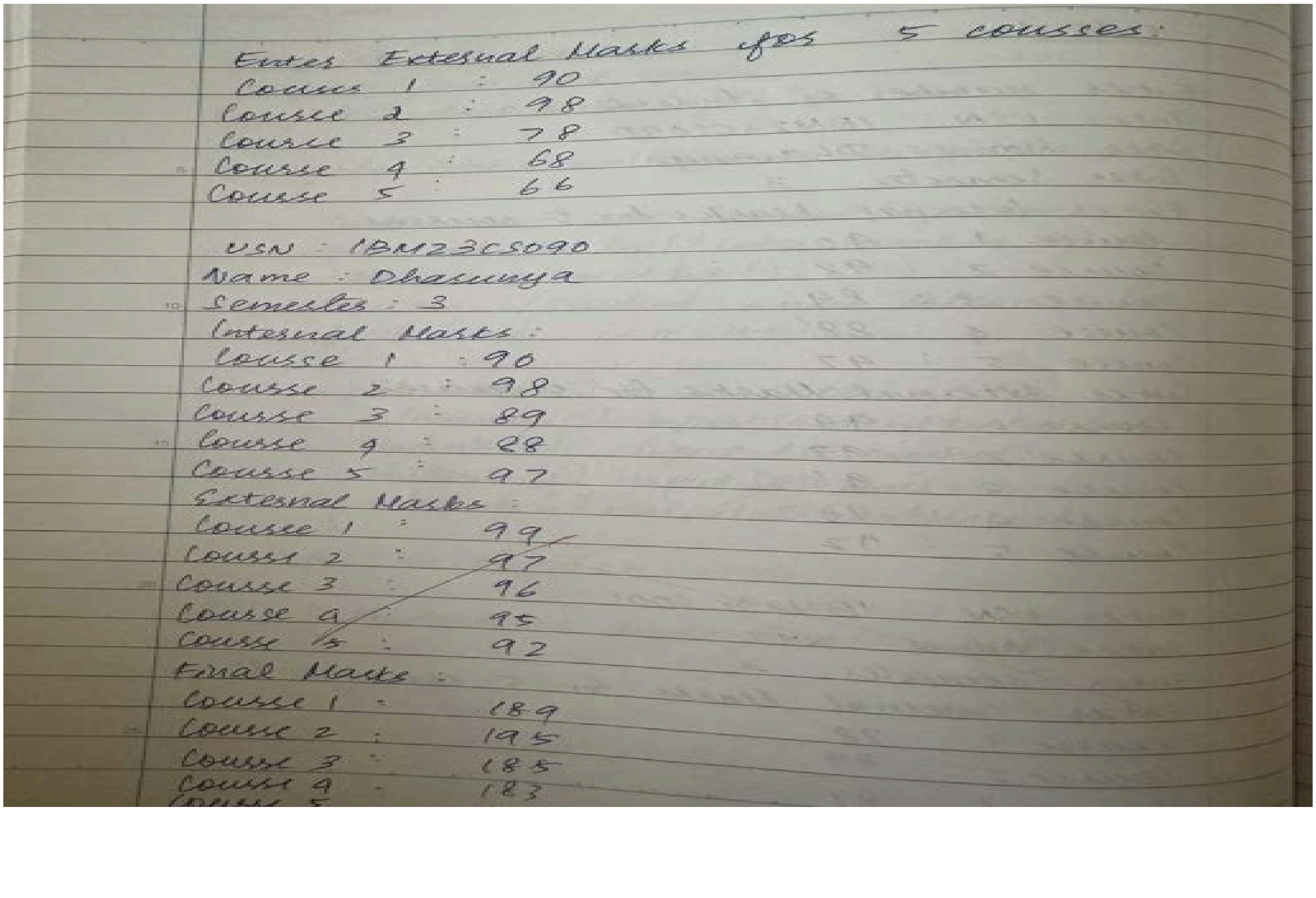
```
import SEE.Externals;
import java.util.Scanner;
public class Main
{
  public static void main (String []args)
  {
   Scanner sc = new Scanner (System.in);
   System.out.println ("Enter number of students");
     int n = sc.nextInt();
   Externals []student =new Externals[n];
   for (int i=0 ; i<n; i++)
    {
      Students[i] = new Externals();
      students[i].inputStudentDetails();
      students[i]   .input CIE marks();
      students[i]  .input SEE marks();
       students[i].calculateFinalMarks();
    }
   for ( int i=0; i<n; i++)
    {
     students[i]. displayFinalMarks();
    }
  }
}
```



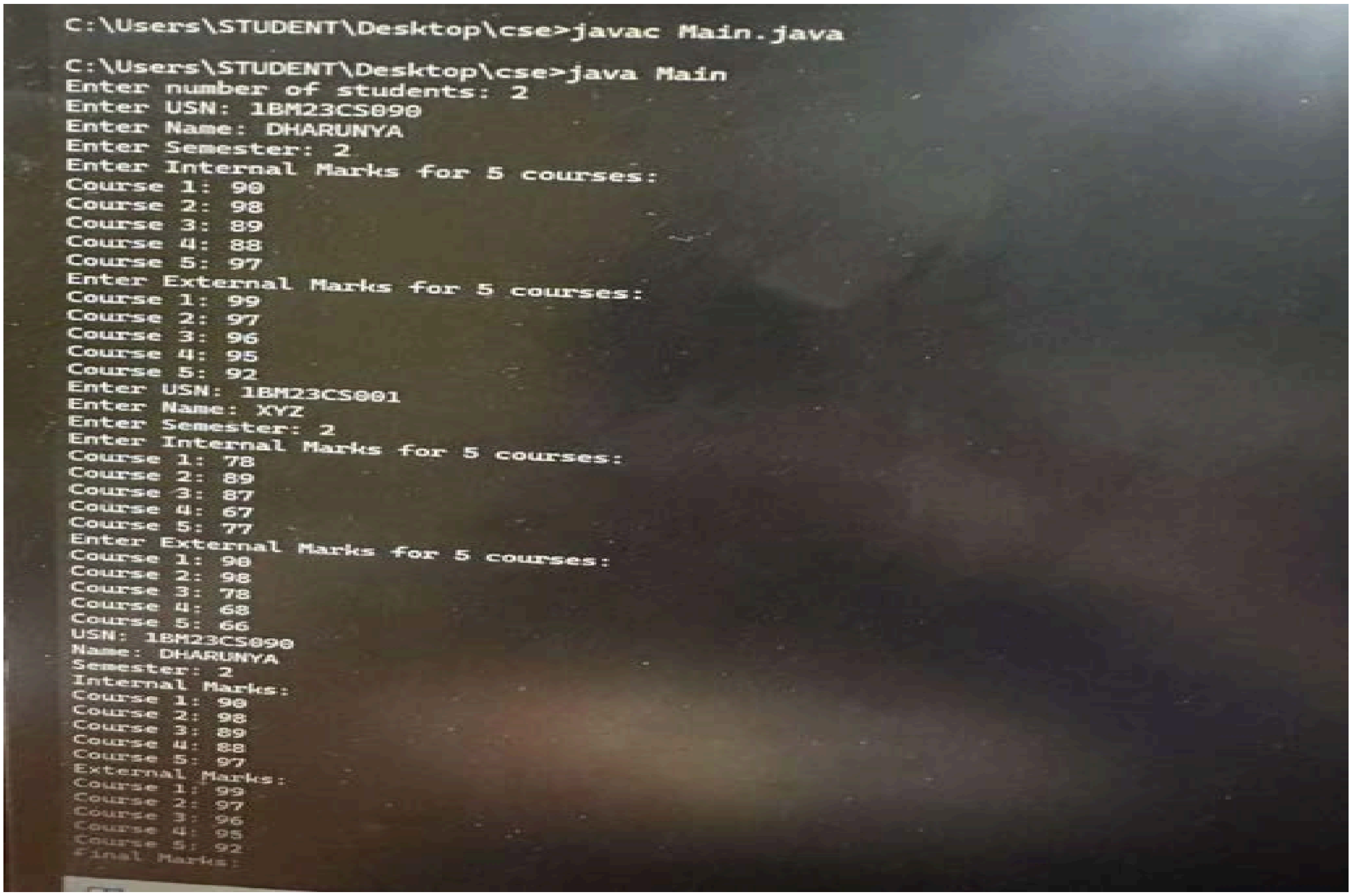
Date :

<u>output</u>

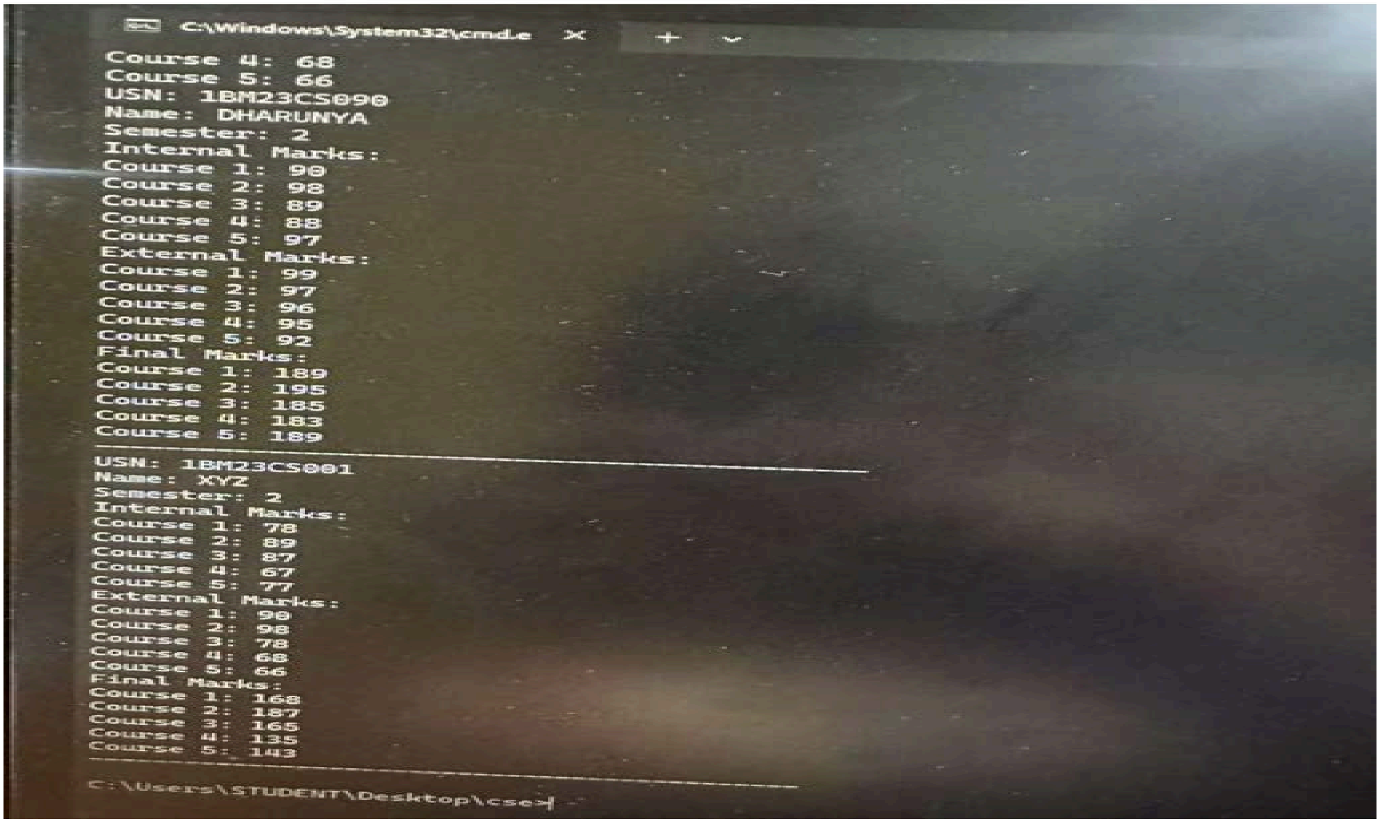
```
Enter number of Students : 2
 Enter USN : 1BM23CS090
Enter Name : Dharunya
 Enter Semester : 3
 Enter Internal Marks for 5 courses :
  Course 1 : 90
  Course 2 : 98
  Course 3 : 89
 Course 4 : 88
Course 5 : 97
Enter External Marks for 5 courses :
 Course 1 : 99
 Course 2 : 97
 Course 3 : 96
Course 4 : 95
Course 5 : 92

Enter USN : 1BM23CS001
Enter Name : XYZ
Enter semester : 3
Enter Internal Marks for 5 courses :
 Course 1 : 78
 Course 2 : 89
 Course 3 : 87
 Course 4 : 67
 Course 5 : 77
```



```
 Enter External Marks for 5 courses :
  Course 1 : 90
 Course 2 : 98
 Course 3 : 78
 Course 4 : 68
 Course 5 : 66

  USN : 1BM23CS090
 Name : Dharunya
 Semester : 3
 Internal Marks :
  Course 1 : 90
 Course 2 : 98
 Course 3 : 89
 Course 4 : 88
 Course 5 : 97
 External Marks :
 Course 1 : 99
 Course 2 : 97
Course 3 : 96
Course 4 : 95
Course 5 : 92
Final Marks :
Course 1 : 189
Course 2 : 195
Course 3 : 185
Course 4 : 183
Course 5
```

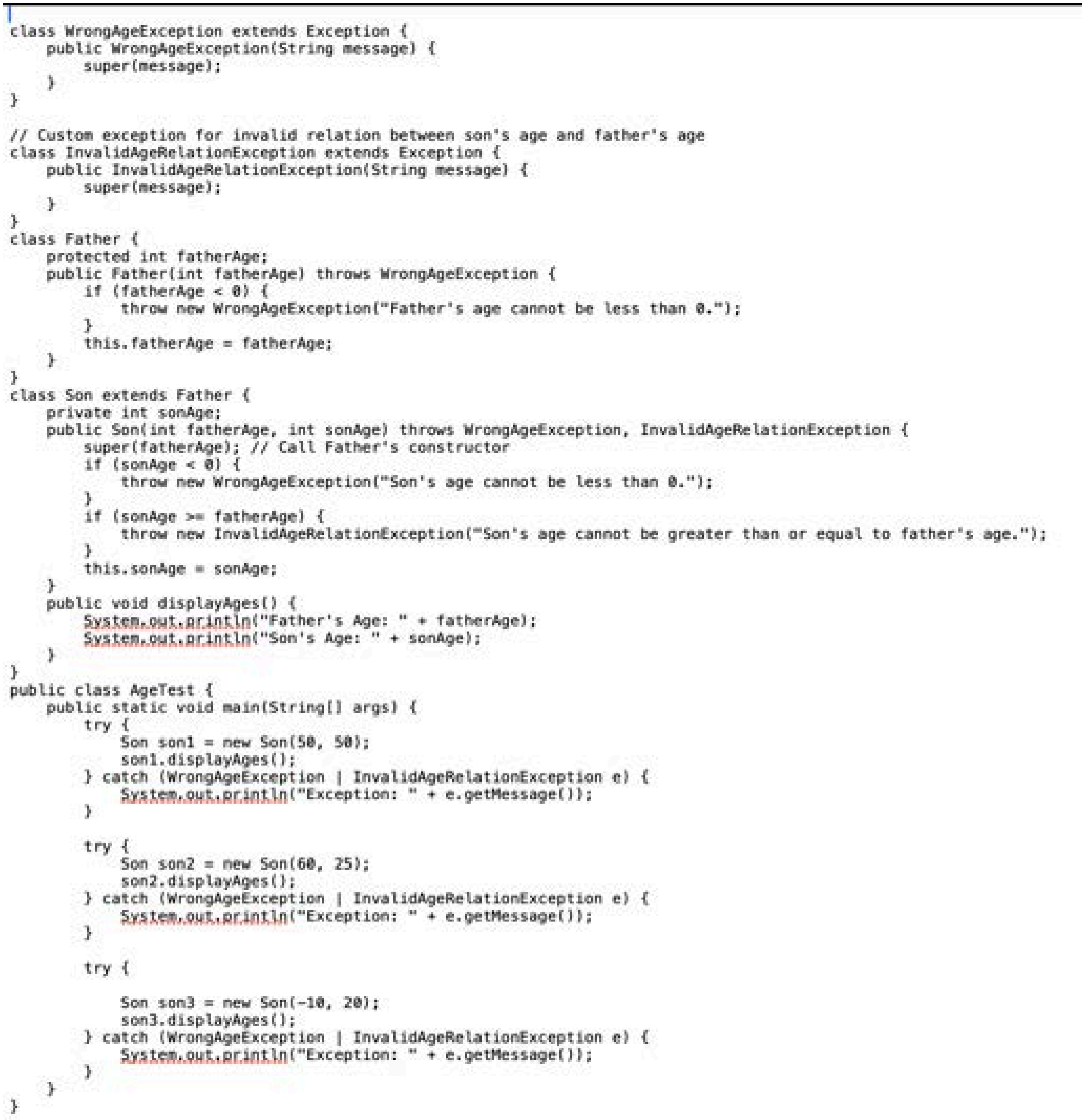


```
C:\Users\STUDENT\Desktop\cse>javac Main.java

C:\Users\STUDENT\Desktop\cse>java Main
Enter number of students: 2
Enter USN: 1BM23CS090
Enter Name: DHARUNYA
Enter Semester: 2
Enter Internal Marks for 5 courses:
Course 1: 90
Course 2: 98
Course 3: 89
Course 4: 88
Course 5: 97
Enter External Marks for 5 courses:
Course 1: 99
Course 2: 97
Course 3: 96
Course 4: 95
Course 5: 92
Enter USN: 1BM23CS001
Enter Name: XYZ
Enter Semester: 2
Enter Internal Marks for 5 courses:
Course 1: 78
Course 2: 89
Course 3: 87
Course 4: 67
Course 5: 77
Enter External Marks for 5 courses:
Course 1: 90
Course 2: 98
Course 3: 78
Course 4: 68
Course 5: 66
USN: 1BM23CS090
Name: DHARUNYA
Semester: 2
Internal Marks:
Course 1: 90
Course 2: 98
Course 3: 89
Course 4: 88
Course 5: 97
External Marks:
Course 1: 99
Course 2: 97
Course 3: 96
Course 4: 95
Course 5: 92
Final Marks:
```

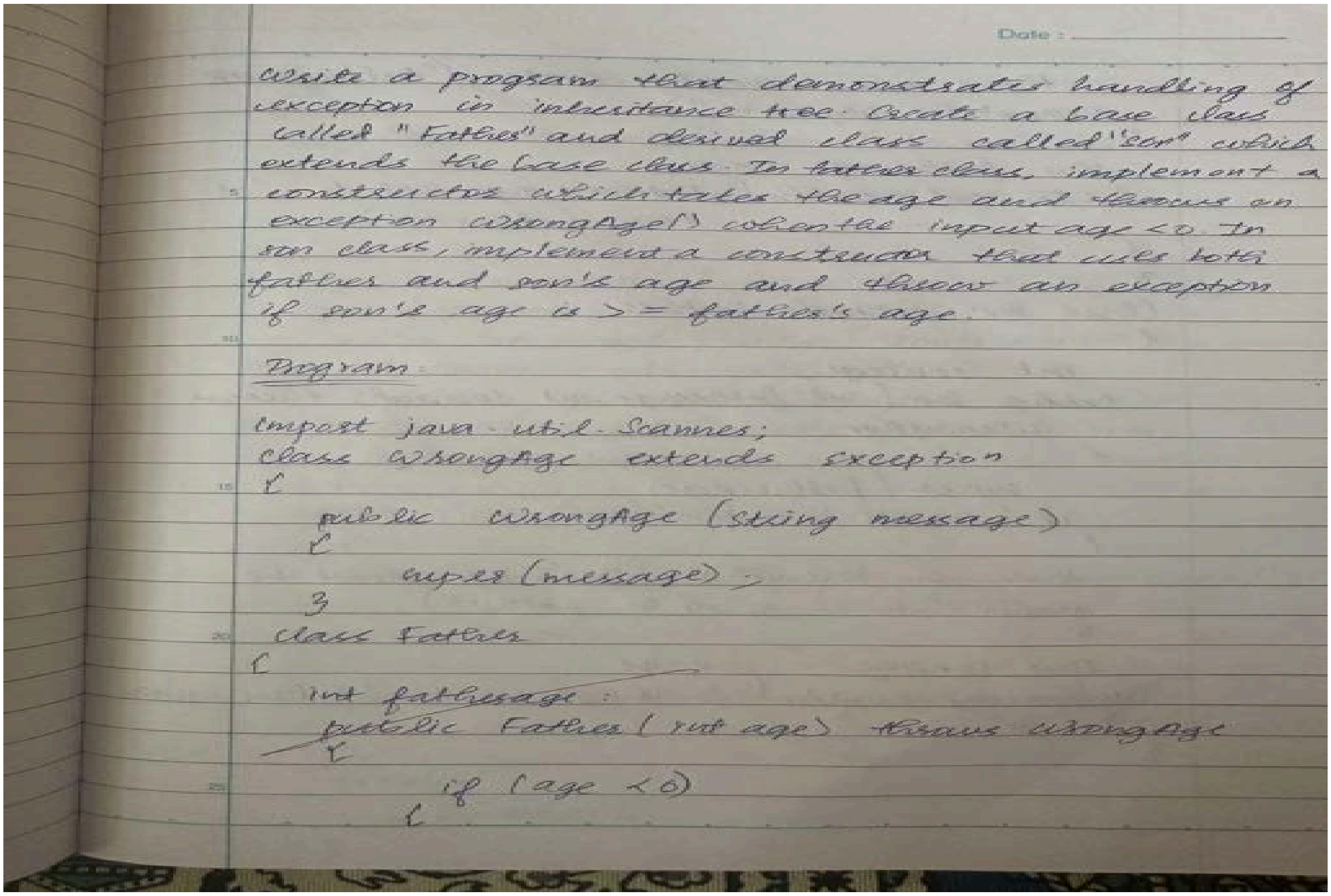


```
Course 4: 68
Course 5: 66
USN: 1BM23CS090
Name: DHARUNYA
Semester: 2
Internal Marks:
Course 1: 90
Course 2: 98
Course 3: 89
Course 4: 88
Course 5: 97
External Marks:
Course 1: 99
Course 2: 97
Course 3: 96
Course 4: 95
Course 5: 92
Final Marks:
Course 1: 189
Course 2: 195
Course 3: 185
Course 4: 183
Course 5: 189
--------------------------------------------------
USN: 1BM23CS001
Name: XYZ
Semester: 2
Internal Marks:
Course 1: 78
Course 2: 89
Course 3: 87
Course 4: 67
Course 5: 77
External Marks:
Course 1: 90
Course 2: 98
Course 3: 78
Course 4: 68
Course 5: 66
Final Marks:
Course 1: 168
Course 2: 187
Course 3: 165
Course 4: 135
Course 5: 143
--------------------------------------------------

C:\Users\STUDENT\Desktop\cse>
```

7.Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age<0. In Son class, implement a constructor that uses both father and son’s age and throws an exception if son’s age is >=father’s age.



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

// Custom exception for invalid relation between son's age and father's age
class InvalidAgeRelationException extends Exception {
    public InvalidAgeRelationException(String message) {
        super(message);
    }
}
class Father {
    protected int fatherAge;
    public Father(int fatherAge) throws WrongAgeException {
        if (fatherAge < 0) {
            throw new WrongAgeException("Father's age cannot be less than 0.");
        }
        this.fatherAge = fatherAge;
    }
}
class Son extends Father {
    private int sonAge;
    public Son(int fatherAge, int sonAge) throws WrongAgeException, InvalidAgeRelationException {
        super(fatherAge); // Call Father's constructor
        if (sonAge < 0) {
            throw new WrongAgeException("Son's age cannot be less than 0.");
        }
        if (sonAge >= fatherAge) {
            throw new InvalidAgeRelationException("Son's age cannot be greater than or equal to father's age.");
        }
        this.sonAge = sonAge;
    }
    public void displayAges() {
        System.out.println("Father's Age: " + fatherAge);
        System.out.println("Son's Age: " + sonAge);
    }
}
public class AgeTest {
    public static void main(String[] args) {
        try {
            Son son1 = new Son(50, 50);
            son1.displayAges();
        } catch (WrongAgeException | InvalidAgeRelationException e) {
            System.out.println("Exception: " + e.getMessage());
        }

        try {
            Son son2 = new Son(60, 25);
            son2.displayAges();
        } catch (WrongAgeException | InvalidAgeRelationException e) {
            System.out.println("Exception: " + e.getMessage());
        }

        try {

            Son son3 = new Son(-10, 20);
            son3.displayAges();
        } catch (WrongAgeException | InvalidAgeRelationException e) {
            System.out.println("Exception: " + e.getMessage());
        }
    }
}
```

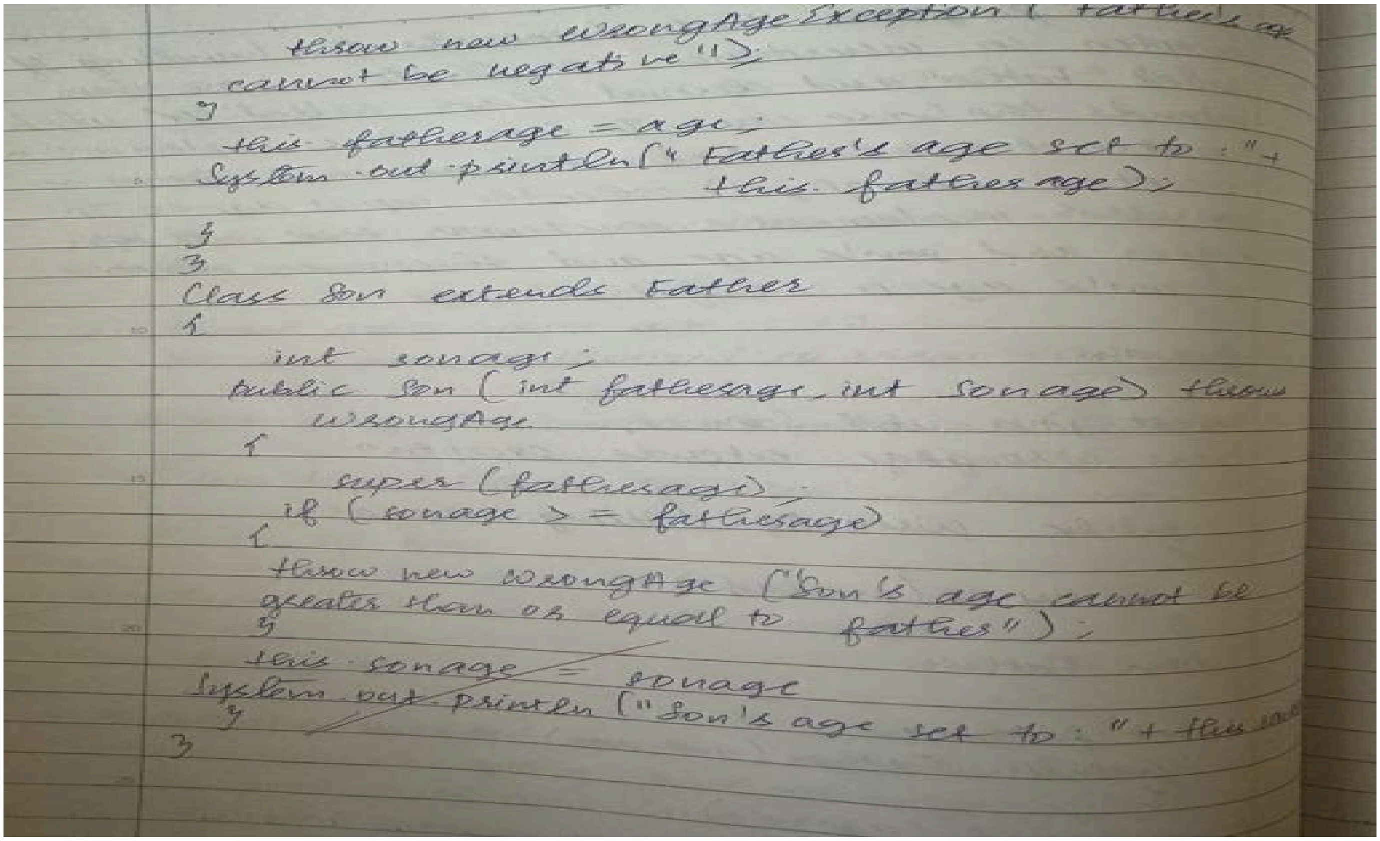


Date :

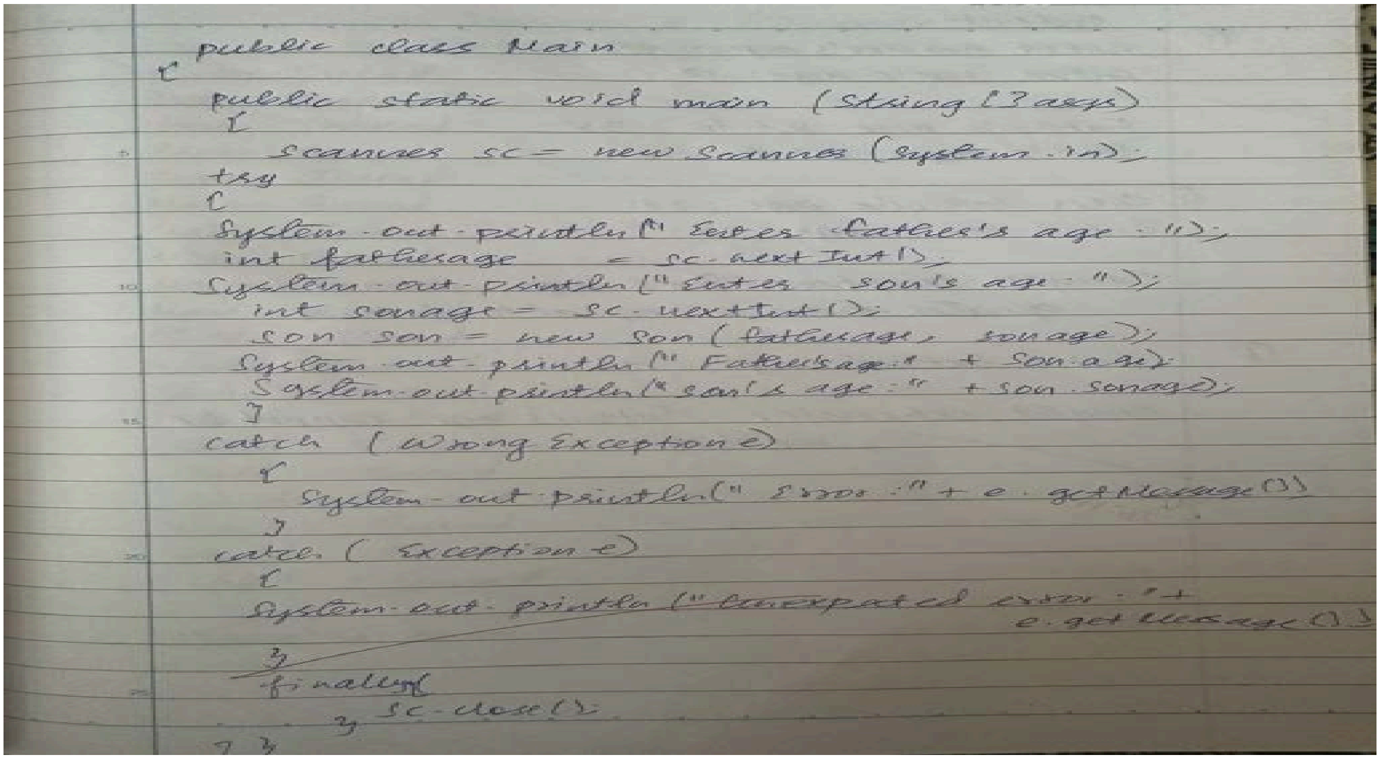
Write a program that demonstrates handling of exception in inheritance tree. Create a base class called "Father" and derived class called "Son" which extends the base class. In father class, implement a constructor which takes the age and throws an exception WrongAge() when the input age <0. In son class, implement a constructor that uses both father and son's age and throw an exception if son's age is >= father's age.

Program

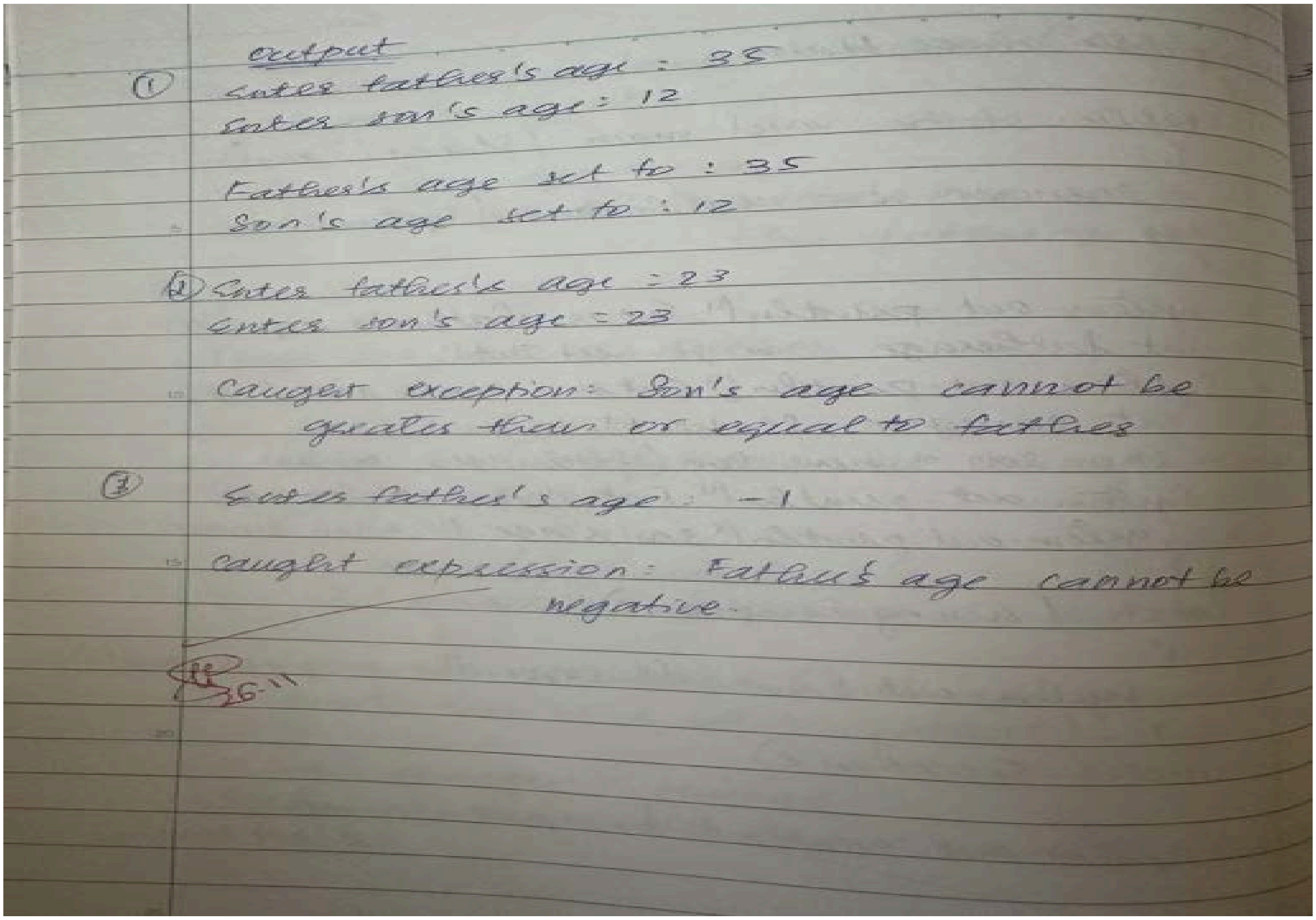
```
import java.util.Scanner;
Class WrongAge extends Exception
{
   public WrongAge (String message)
   {
        super (message);
   }
 class Father
{
   int fatherage;
   public Father ( int age) throws WrongAge
   {
        if (age <0)
      {
```



```
      throw new WrongAgeException ( Father's age
  cannot be negative");
  }
  this.fatherage = age;
  System.out.println(" Father's age set to : " +
                          this.fatherage);

 }
}
Class Son extends Father
{
   int sonage;
  public Son ( int fatherage, int sonage) throws
      WrongAge
   {
        super (fatherage);
     if (sonage >= fatherage)
   {
    throw new WrongAge ("Son's age cannot be
  greater than or equal to father");
   }
   this.sonage = sonage
System.out.println ("Son's age set to : " + this.son
  }
 }
```



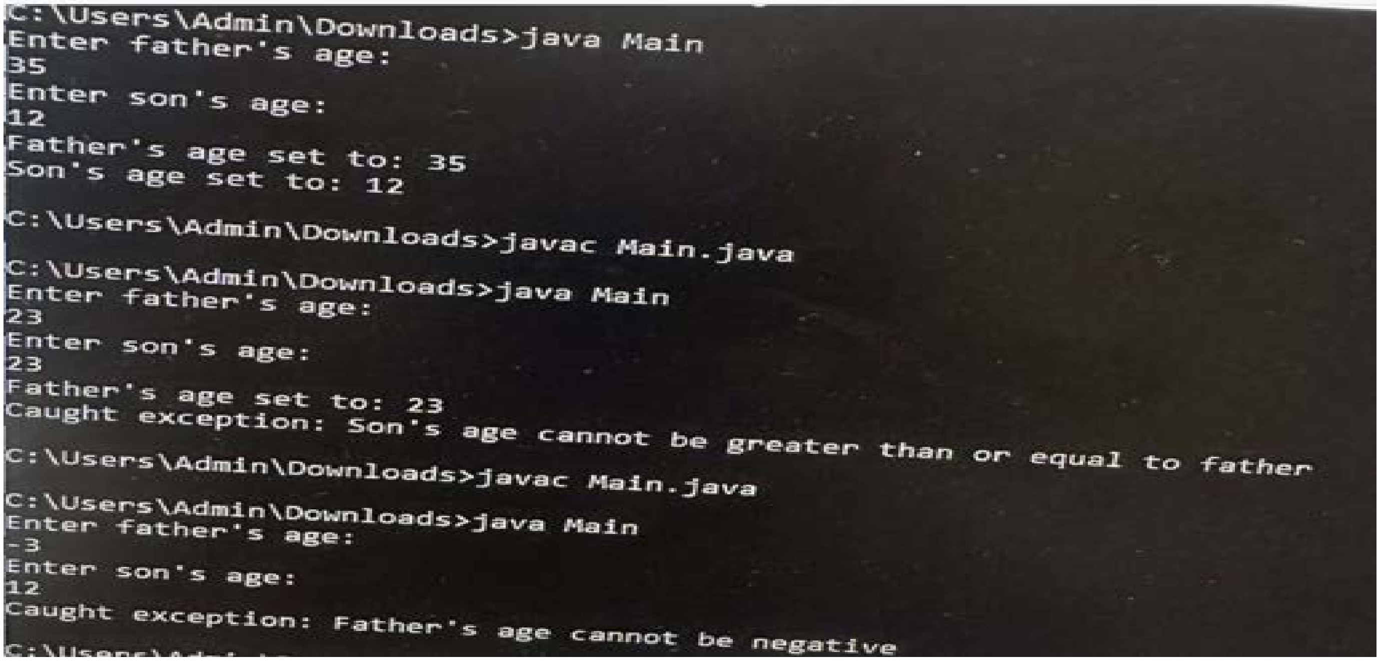
```
public class Main
{
 public static void main (String[] args)
  {
    Scanner sc = new Scanner (System.in);
 try
 {
 System.out.println(" Enter father's age : ");
 int fatherage    = sc.nextInt();
 System.out.println(" Enter  son's age : ");
  int sonage = sc.nextInt();
  Son son = new Son(fatherage, sonage);
  System.out.println(" Father'sage:" + son.age);
  System.out.println(" son's age :" + son.sonage);
  }
 catch (WrongException e)
  {
   System.out.println(" Error :" + e.getMessage())
  }
 catch ( Exception e)
  {
 System.out.println(" Unexpated error :" +
                              e.getMessage())
  }
  finally{
     sc.close();
    }
 }}
```



output

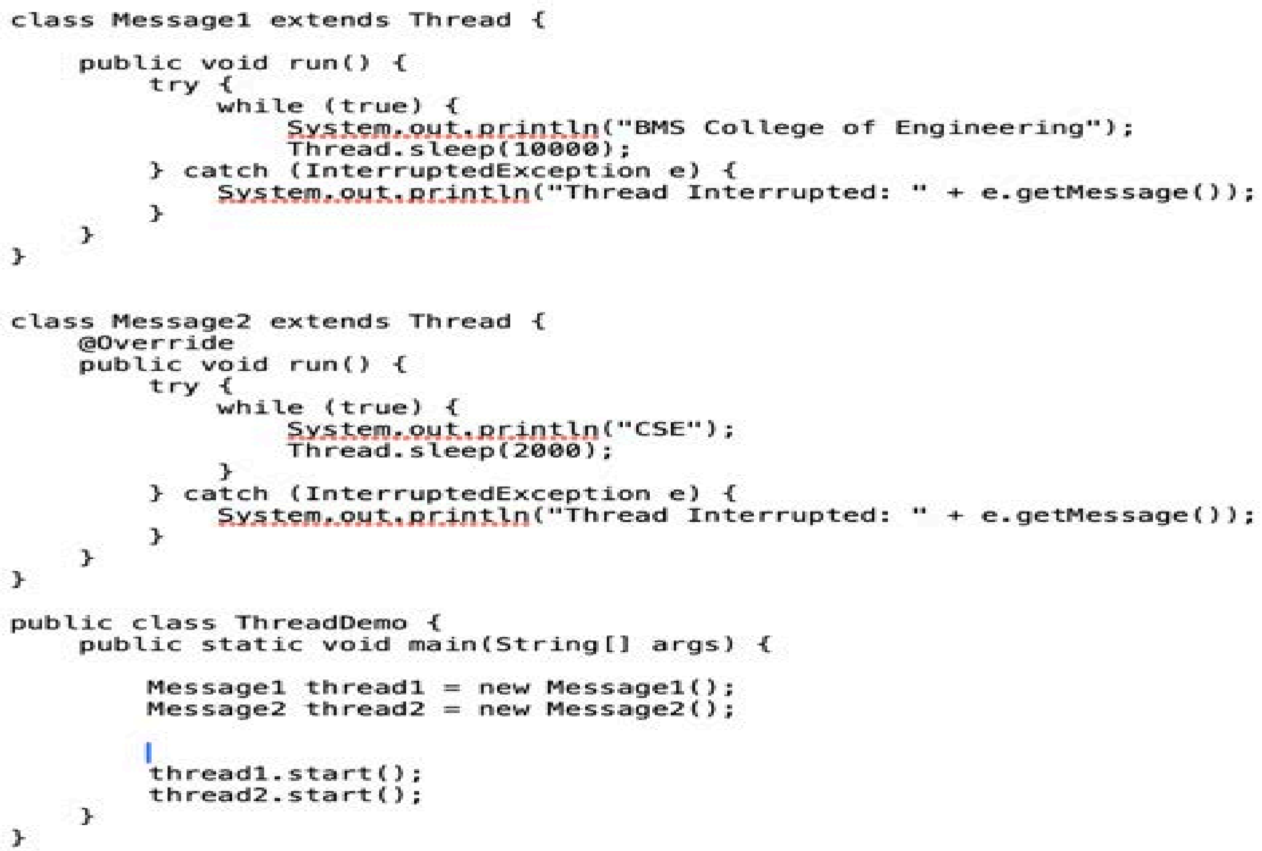
```
①  Enter father's age : 35
    Enter son's age: 12

    Father's age set to : 35
    Son's age set to : 12

②  Enter father's age : 23
    Enter son's age : 23

    Caught exception: Son's age cannot be
         greater than or equal to father

③   Enter father's age : -1

    caught expression: Father's age cannot be
                 negative.
```

26-11

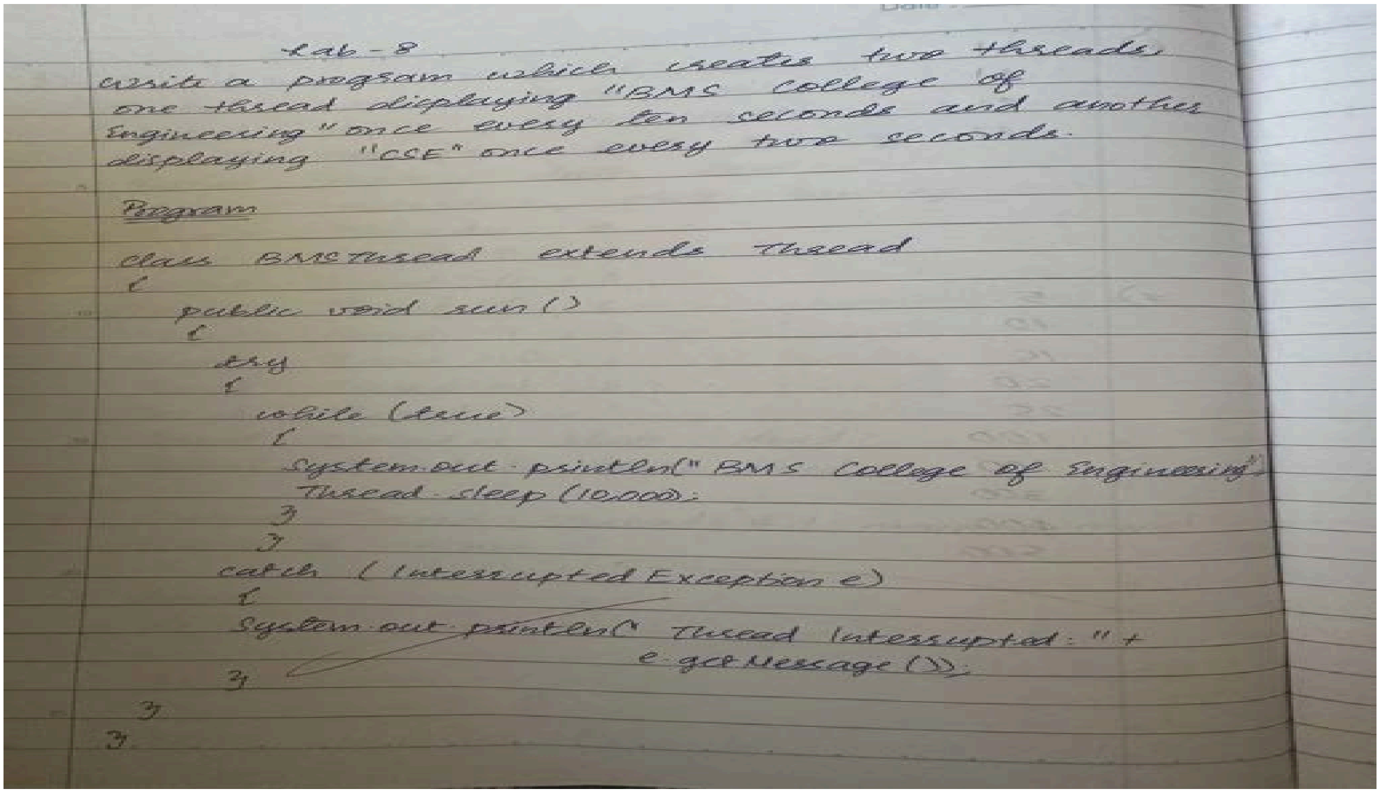


```
C:\Users\Admin\Downloads>java Main
Enter father's age:
35
Enter son's age:
12
Father's age set to: 35
Son's age set to: 12

C:\Users\Admin\Downloads>javac Main.java

C:\Users\Admin\Downloads>java Main
Enter father's age:
23
Enter son's age:
23
Father's age set to: 23
Caught exception: Son's age cannot be greater than or equal to father

C:\Users\Admin\Downloads>javac Main.java

C:\Users\Admin\Downloads>java Main
Enter father's age:
-3
Enter son's age:
12
Caught exception: Father's age cannot be negative

C:\Users\Adm
```

8.Write a program which creates two threads, one thread displaying “BMS College of Engineering” once every ten seconds and another displaying “CSE” once every two seconds.



```
class Message1 extends Thread {

    public void run() {
        try {
            while (true) {
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000);
        } catch (InterruptedException e) {
            System.out.println("Thread Interrupted: " + e.getMessage());
        }
    }
}


class Message2 extends Thread {
    @Override
    public void run() {
        try {
            while (true) {
                System.out.println("CSE");
                Thread.sleep(2000);
            }
        } catch (InterruptedException e) {
            System.out.println("Thread Interrupted: " + e.getMessage());
        }
    }
}

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

        Message1 thread1 = new Message1();
        Message2 thread2 = new Message2();


        thread1.start();
        thread2.start();
    }
}
```

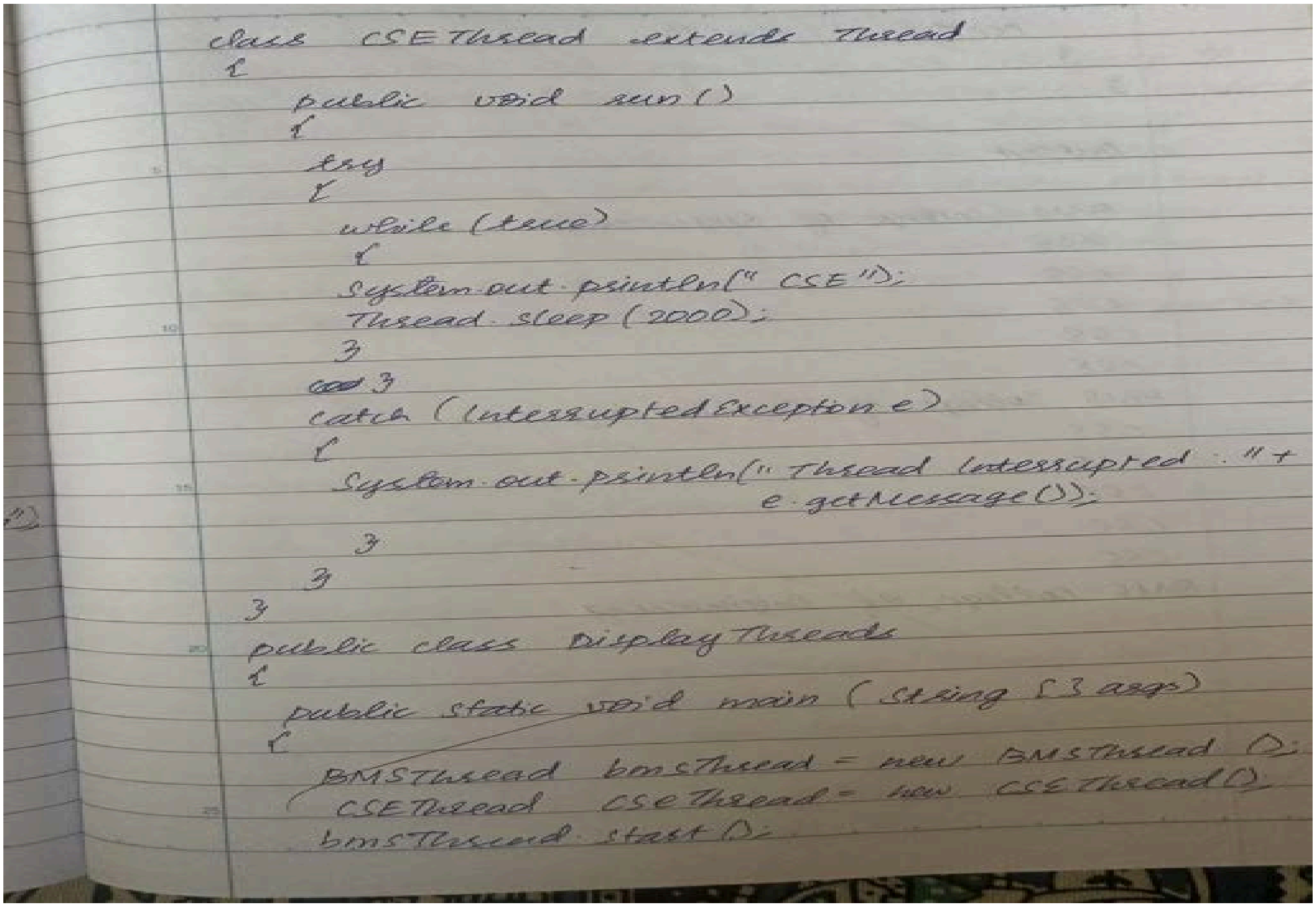


Lab-8

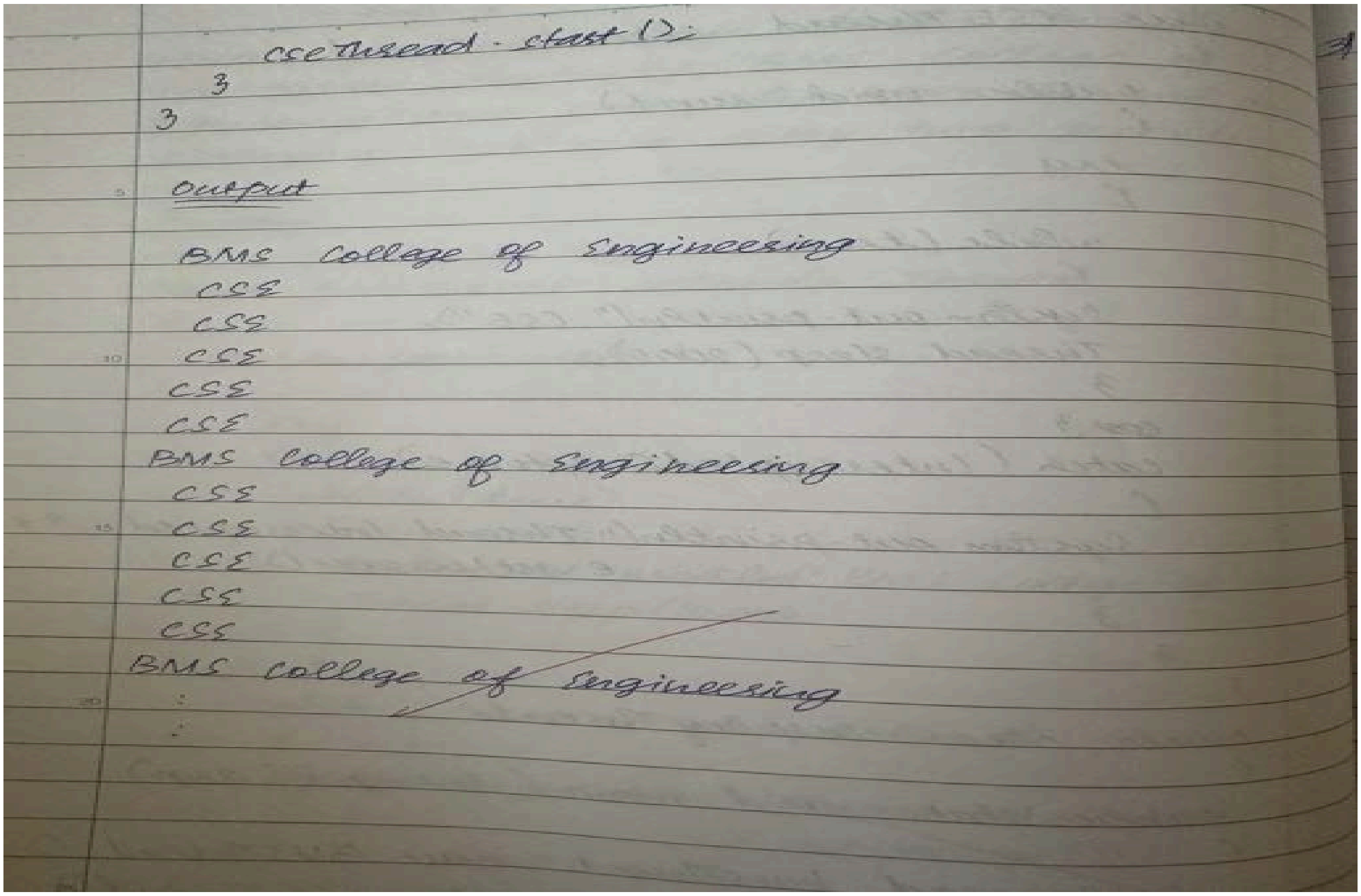
Write a program which creates two threads, one thread displaying "BMS College of Engineering" once every ten seconds and another displaying "CSE" once every two seconds.

Program

```
class BMSThread extends Thread
{
  public void run()
  {
    try
    {
     while (true)
      {
       System.out.println("BMS College of Engineering");
       Thread.sleep(10,000);
      }
     }
    catch (InterruptedException e)
    {
    System.out.println(" Thread Interrupted: " +
                          e.getMessage());
    }
  }
}
```



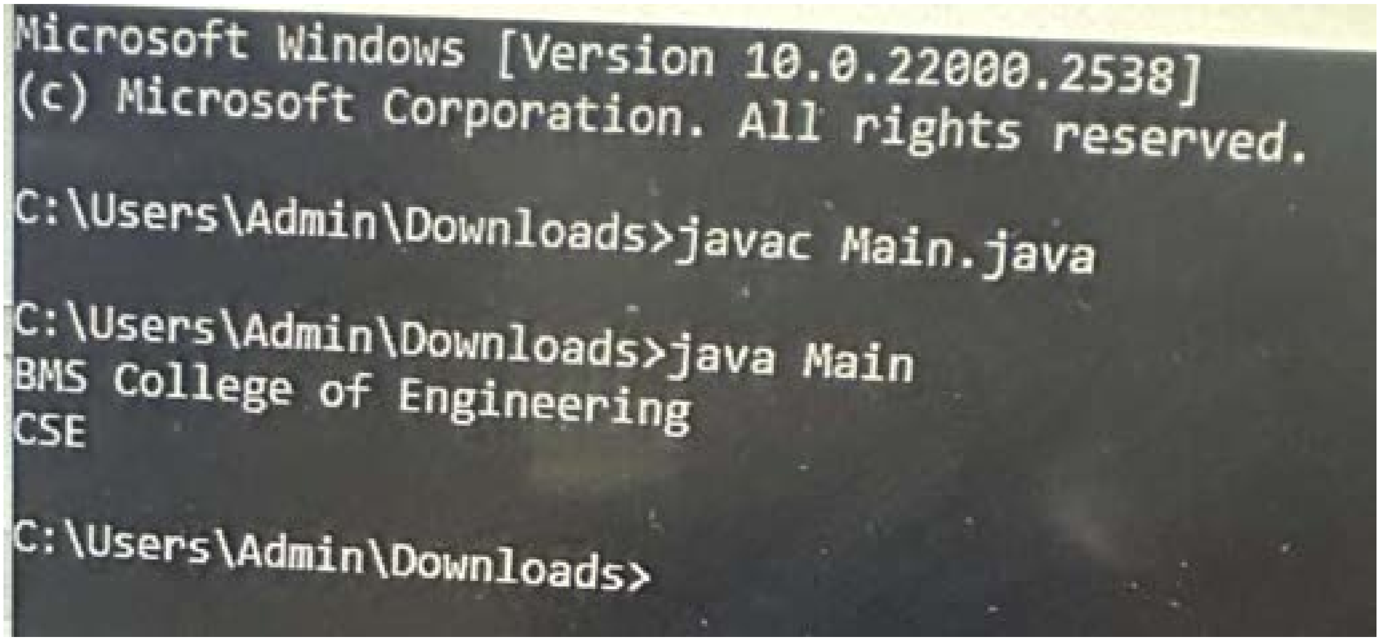
```
class CSEThread extends Thread
 {
      public void run()
      {
      try
      {
        while (true)
         {
        System.out.println(" CSE");
        Thread.sleep(2000);
        }
      }
      catch (InterruptedException e)
      {
       System.out.println(" Thread Interrupted : " +
                               e.getMessage());
        }
      }
 }
public class DisplayThreads
 {
  public static void main (String [] args)
  {
    BMSThread bmsThread = new BMSThread();
    CSEThread cseThread = new CSEThread();
   bmsThread.start();
```



```
        cseThread.start();
    }
}
```

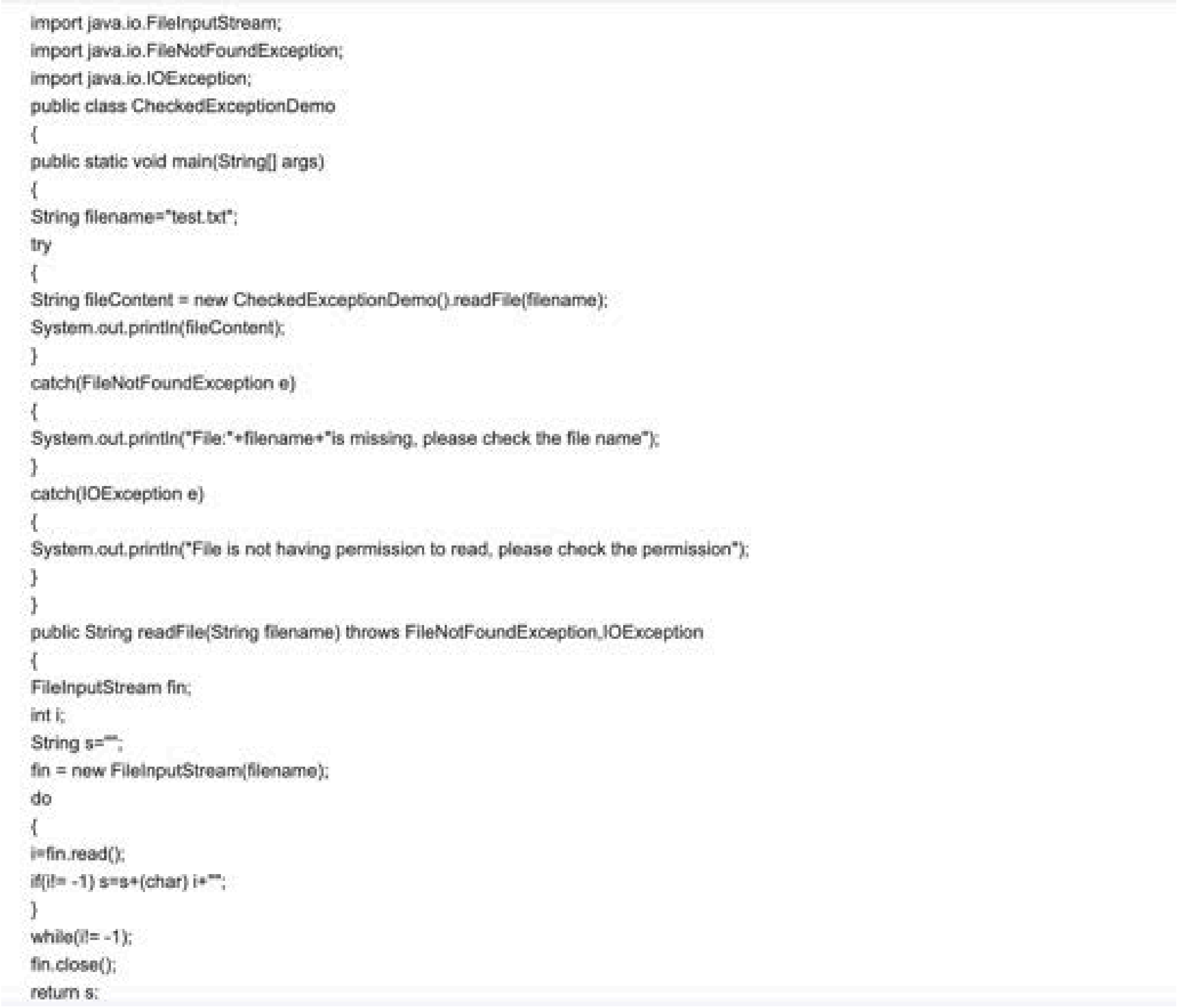
Output

```
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
:
:
```

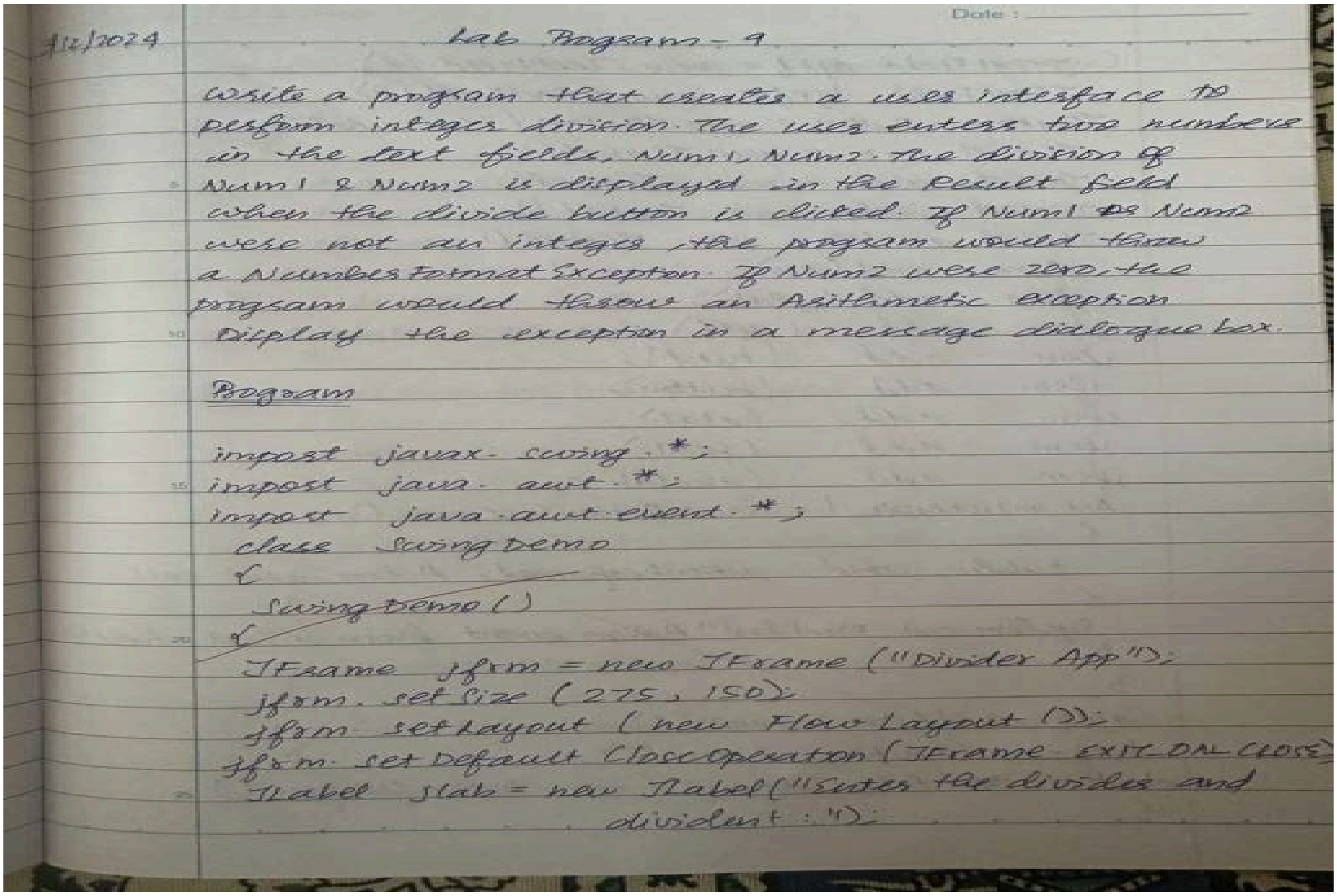


```
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Downloads>javac Main.java

C:\Users\Admin\Downloads>java Main
BMS College of Engineering
CSE

C:\Users\Admin\Downloads>
```

9.Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.



```
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
public class CheckedExceptionDemo
{
public static void main(String[] args)
{
String filename="test.txt";
try
{
String fileContent = new CheckedExceptionDemo().readFile(filename);
System.out.println(fileContent);
}
catch(FileNotFoundException e)
{
System.out.println("File:"+filename+"is missing, please check the file name");
}
catch(IOException e)
{
System.out.println("File is not having permission to read, please check the permission");
}
}
public String readFile(String filename) throws FileNotFoundException,IOException
{
FileInputStream fin;
int i;
String s="";
fin = new FileInputStream(filename);
do
{
i=fin.read();
if(i!= -1) s=s+(char) i+"";
}
while(i!= -1);
fin.close();
return s;
```



Date :

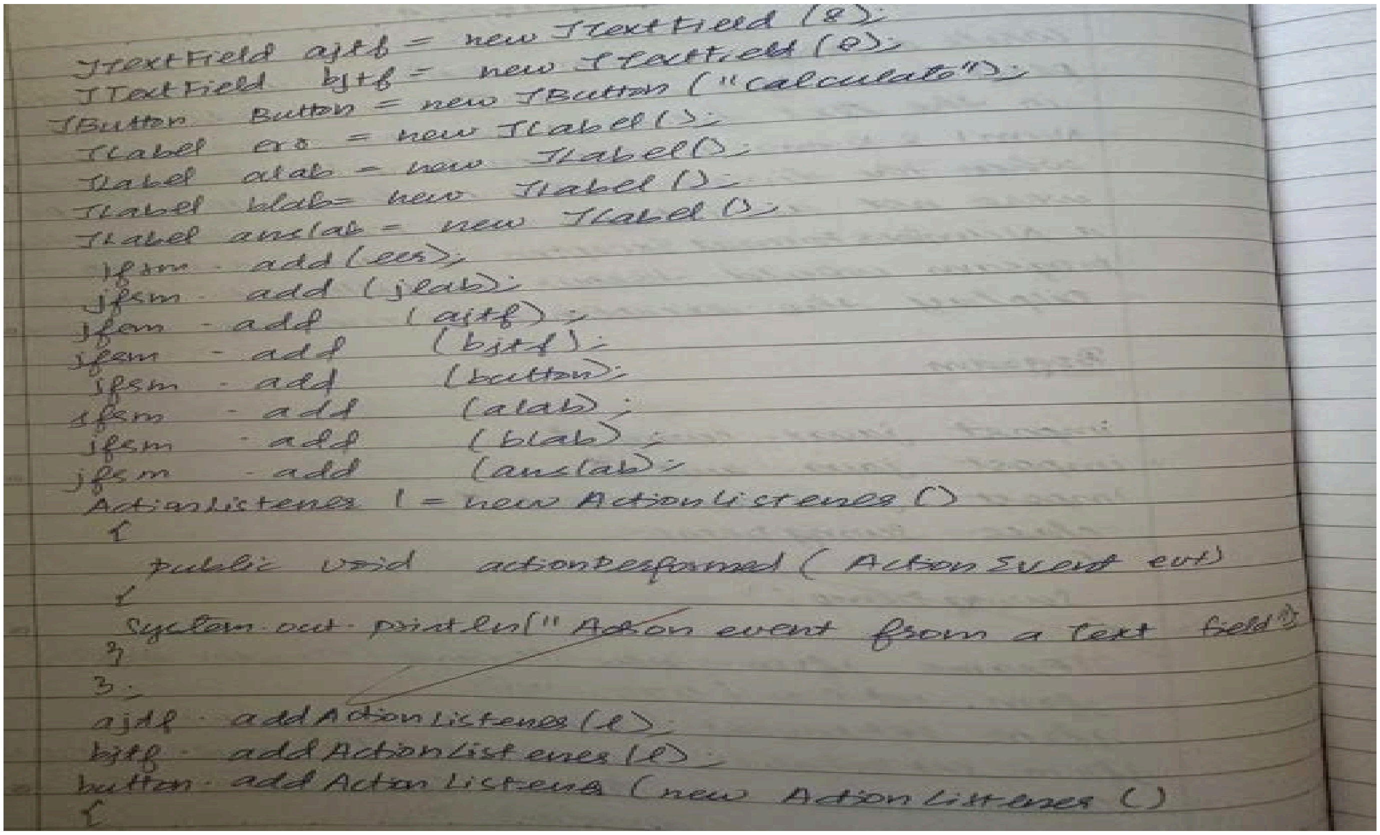
1/12/2024

Lab Program - 9

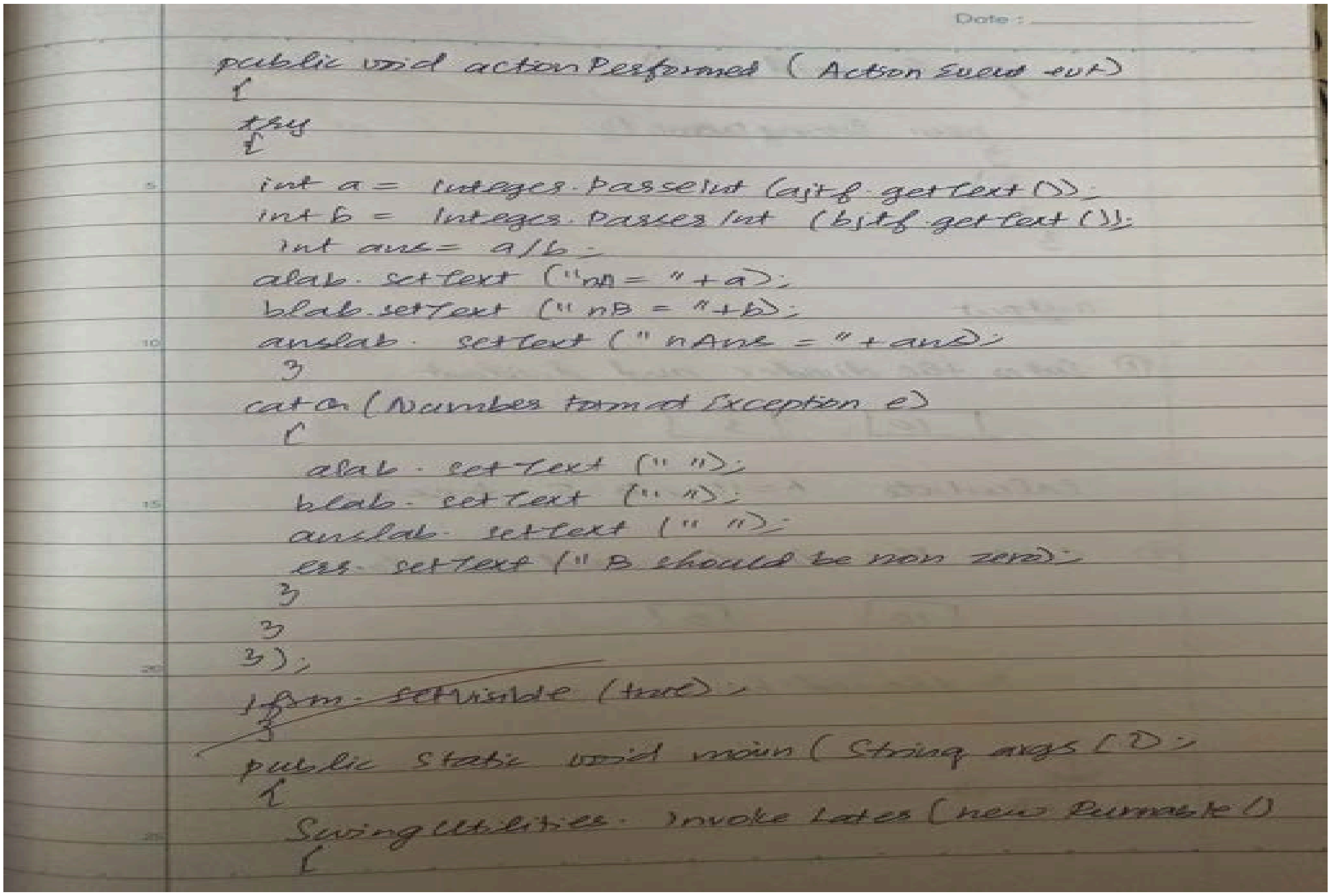
Write a program that creates a user interface to perform integer division. The user enters two numbers in the text fields, Num1, Num2. The division of Num1 & Num2 is displayed in the Result field when the divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were zero, the program would throw an Arithmetic exception Display the exception in a message dialogue box.

Program

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
 class SwingDemo
 {
  SwingDemo()
 {
  JFrame jfrm = new JFrame ("Divider App");
  jfrm. setSize (275, 150);
  jfrm setLayout (new FlowLayout ());
 jfrm. setDefaultCloseOperation (JFrame EXIT_ON_CLOSE);
  JLabel jlab = new JLabel("Enter the divider and
                    divident :");
```

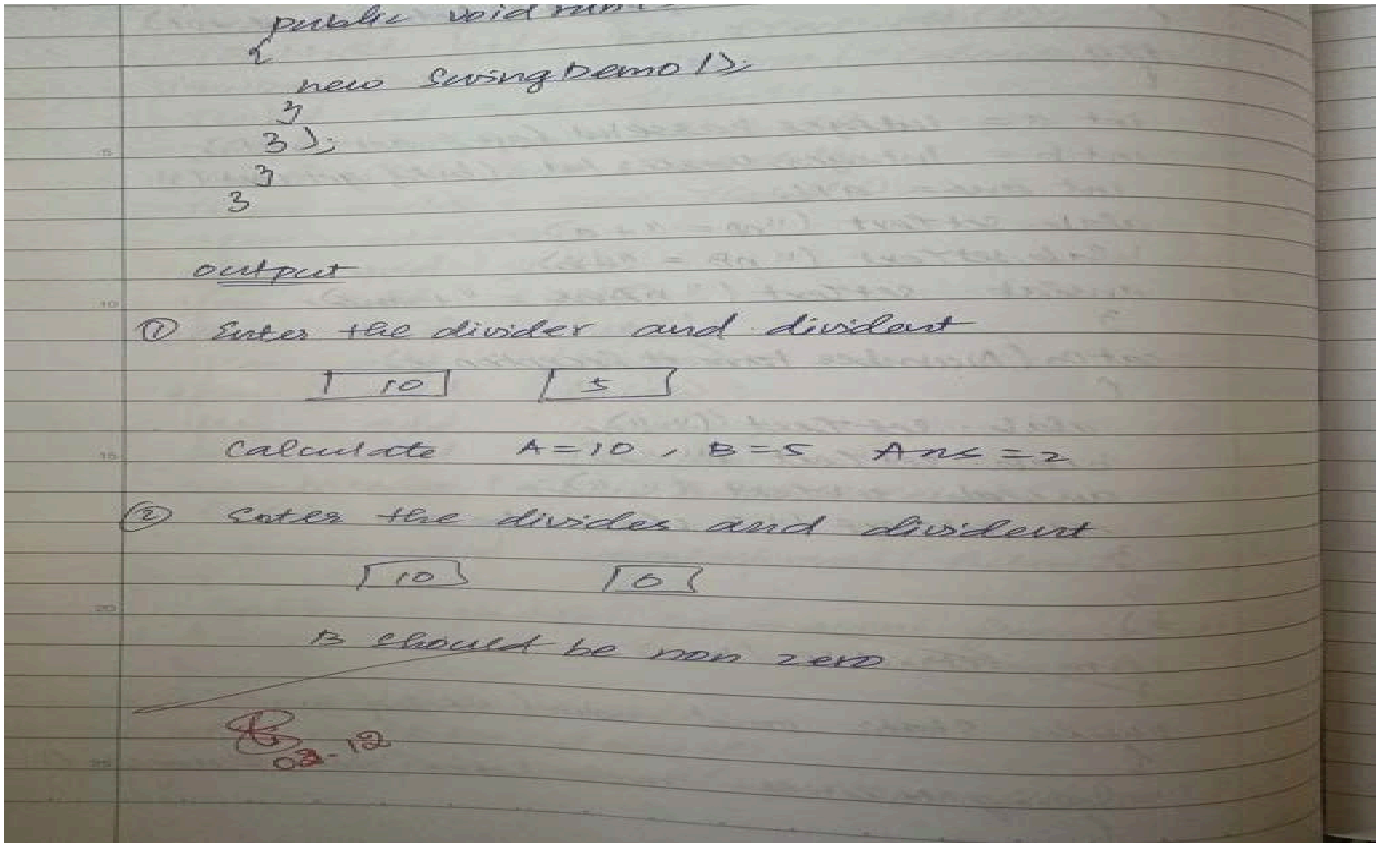


```
JTextField ajtf = new JTextField (8);
JTextField bjtf = new JTextField (8);
JButton Button = new JButton ("Calculate");
JLabel err = new JLabel();
JLabel alab = new JLabel();
JLabel blab= new JLabel ();
JLabel anslab = new JLabel ();
 jfrm . add(err);
 jfrm . add (jlab);
jfrm - add  (ajtf);
jfrm - add  (bjtf);
 jfrm . add  (button);
jfrm - add  (alab);
 jfrm . add  (blab);
jfrm  . add  (anslab);
ActionListener l = new ActionListener ()
 {
  public void actionPerformed ( ActionEvent evt)
 {
 System.out.println(" Action event from a Text field");
 }
 };
ajtf . addActionListener (l);
bjtf . addActionListener (l);
button. addActionListener (new ActionListener ()
{
```



Date :

```
public void actionPerformed ( Action Event evt)
 {
 try
 {
  int a = Integes.passeInt (ajtf.getText ());
  int b = Integes.Passes Int  (bjtf.getText ());
   int ans= a/b;
  alab. setText ("nA = "+a);
  blab.setText (" nB = "+b);
  anslab. setText (" nAns = "+ans);
   }
  catch (Number formet Exception e)
   {
    alab. setText (" ");
    blab. setText (" ");
    anslab. setText (" ");
    ers. setText (" B should be non zero);
    }
   }
  });
  ~~jfrm. setVisible (true);~~
  ~~}~~
 public Static void main ( String args [])
  {
   SwingUtilities. invoke Later (new Runnable()
   {
```



```
    public void run()
   {
    new SwingDemo();
   }
  });
  }
 }
```

output

① Enter the divider and divident

| 10 | | 5 |

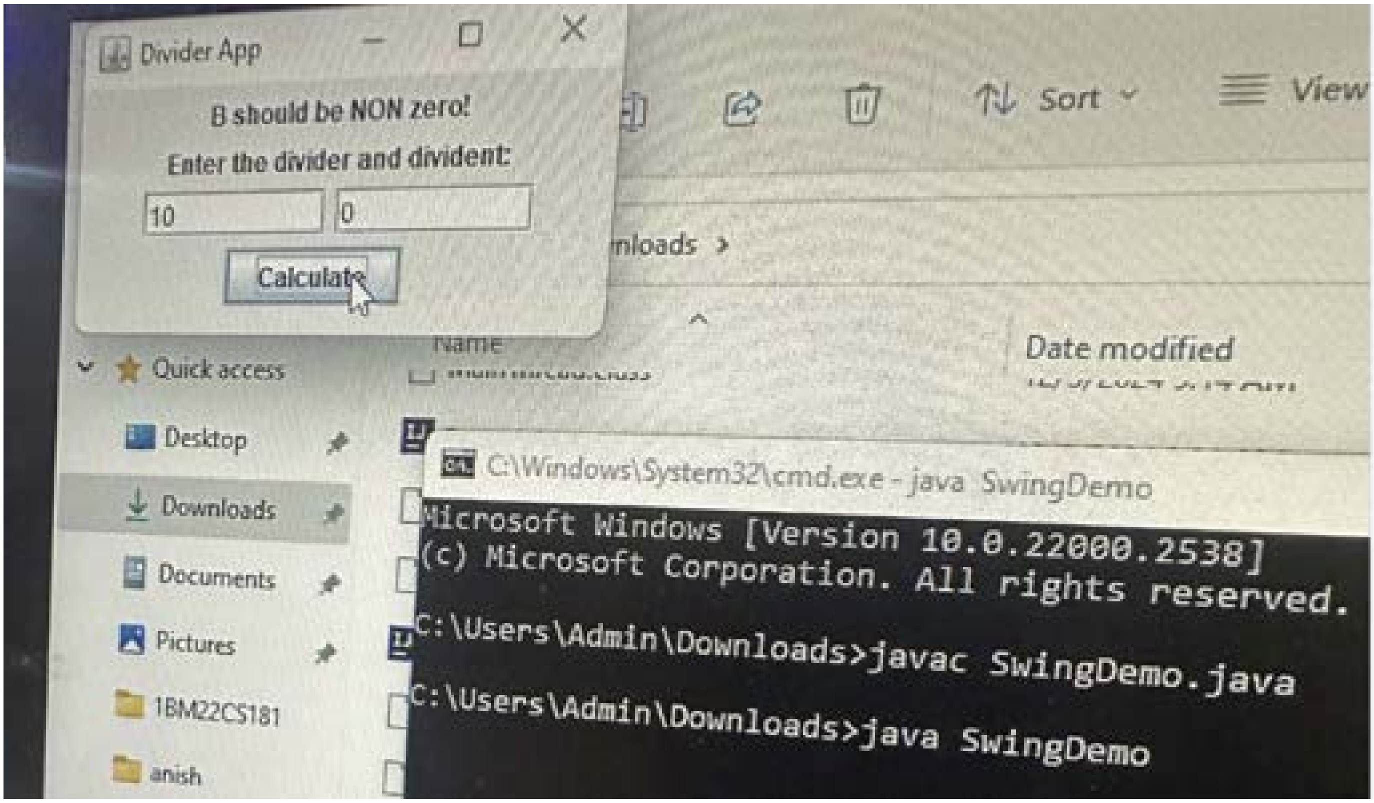
Calculate   A=10 , B=5 Ans = 2

② Enter the divides and divident

| 10 | | 0 |

B should be non zero

03-12



B should be NON zero!

Enter the divider and divident:

10 | 0

```
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Downloads>javac SwingDemo.java

C:\Users\Admin\Downloads>java SwingDemo
```

EXTRA PROGRAM

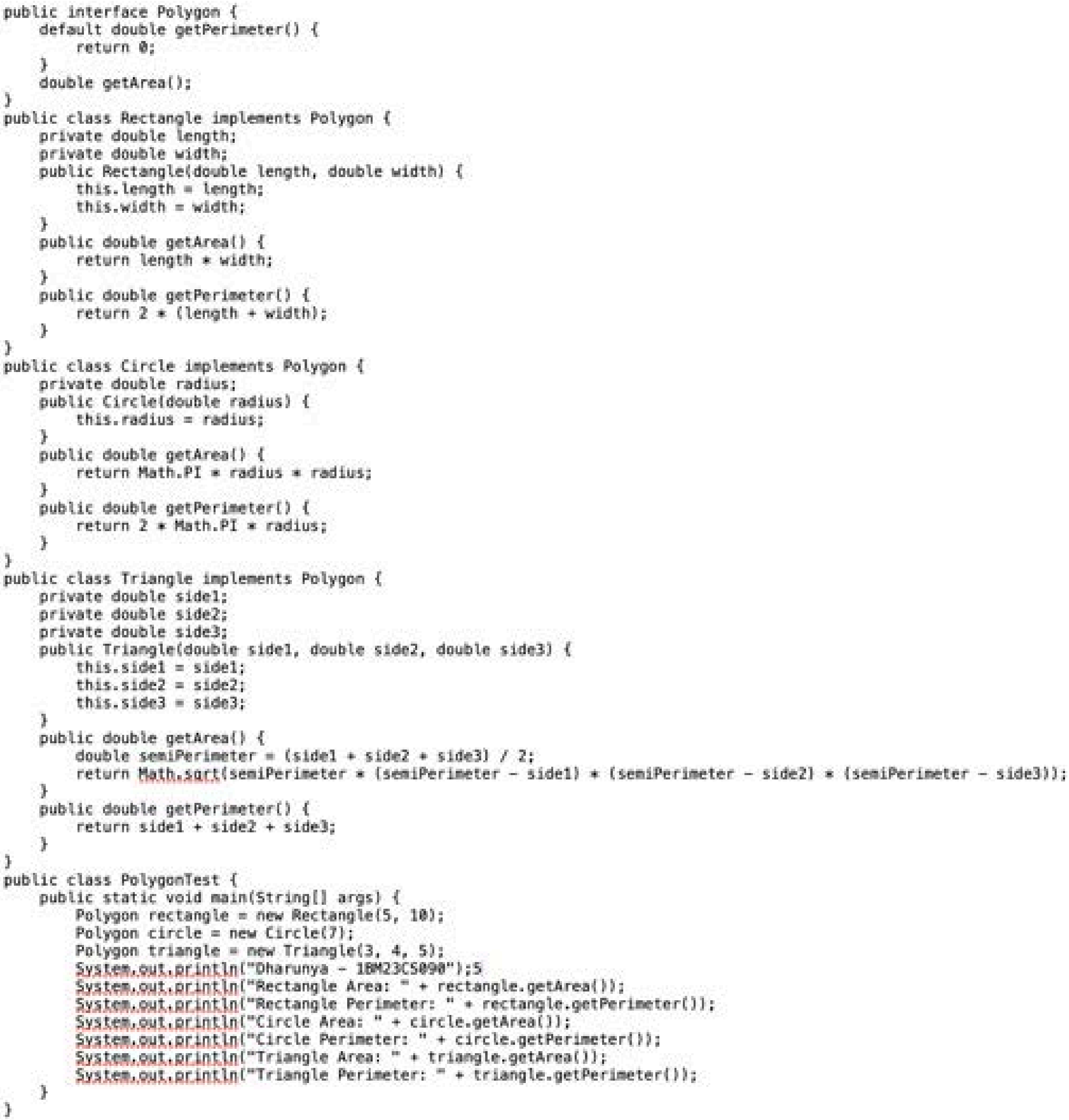
We have created an interface named Polygon. It includes a default method getPerimeter() and an abstract method getArea().

We can calculate the perimeter of all polygons in the same manner so we implemented the body of getPerimeter() in Polygon.

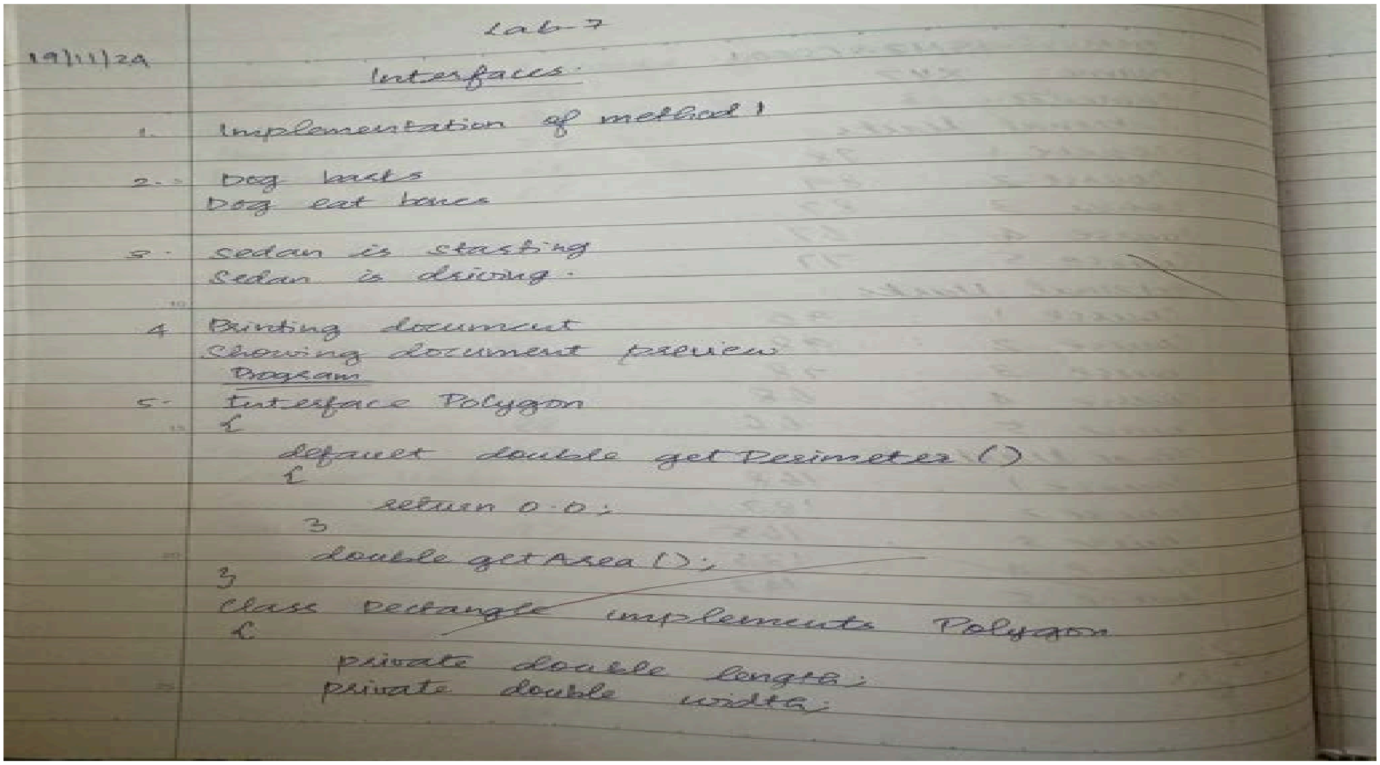
Now, all polygons that implement Polygon can use getPerimeter() to calculate perimeter.

However, the rule for calculating the area is different for different polygons. Hence, getArea() is included without implementation.

Any class that implements Polygon must provide an implementation of getArea().



```
public interface Polygon {
    default double getPerimeter() {
        return 0;
    }
    double getArea();
}
public class Rectangle implements Polygon {
    private double length;
    private double width;
    public Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
    }
    public double getArea() {
        return length * width;
    }
    public double getPerimeter() {
        return 2 * (length + width);
    }
}
public class Circle implements Polygon {
    private double radius;
    public Circle(double radius) {
        this.radius = radius;
    }
    public double getArea() {
        return Math.PI * radius * radius;
    }
    public double getPerimeter() {
        return 2 * Math.PI * radius;
    }
}
public class Triangle implements Polygon {
    private double side1;
    private double side2;
    private double side3;
    public Triangle(double side1, double side2, double side3) {
        this.side1 = side1;
        this.side2 = side2;
        this.side3 = side3;
    }
    public double getArea() {
        double semiPerimeter = (side1 + side2 + side3) / 2;
        return Math.sqrt(semiPerimeter * (semiPerimeter - side1) * (semiPerimeter - side2) * (semiPerimeter - side3));
    }
    public double getPerimeter() {
        return side1 + side2 + side3;
    }
}
public class PolygonTest {
    public static void main(String[] args) {
        Polygon rectangle = new Rectangle(5, 10);
        Polygon circle = new Circle(7);
        Polygon triangle = new Triangle(3, 4, 5);
        System.out.println("Dharunya - 1BM23CS090");5
        System.out.println("Rectangle Area: " + rectangle.getArea());
        System.out.println("Rectangle Perimeter: " + rectangle.getPerimeter());
        System.out.println("Circle Area: " + circle.getArea());
        System.out.println("Circle Perimeter: " + circle.getPerimeter());
        System.out.println("Triangle Area: " + triangle.getArea());
        System.out.println("Triangle Perimeter: " + triangle.getPerimeter());
    }
}
```



19/11/24

Lab-7

Interfaces

1. Implementation of method!

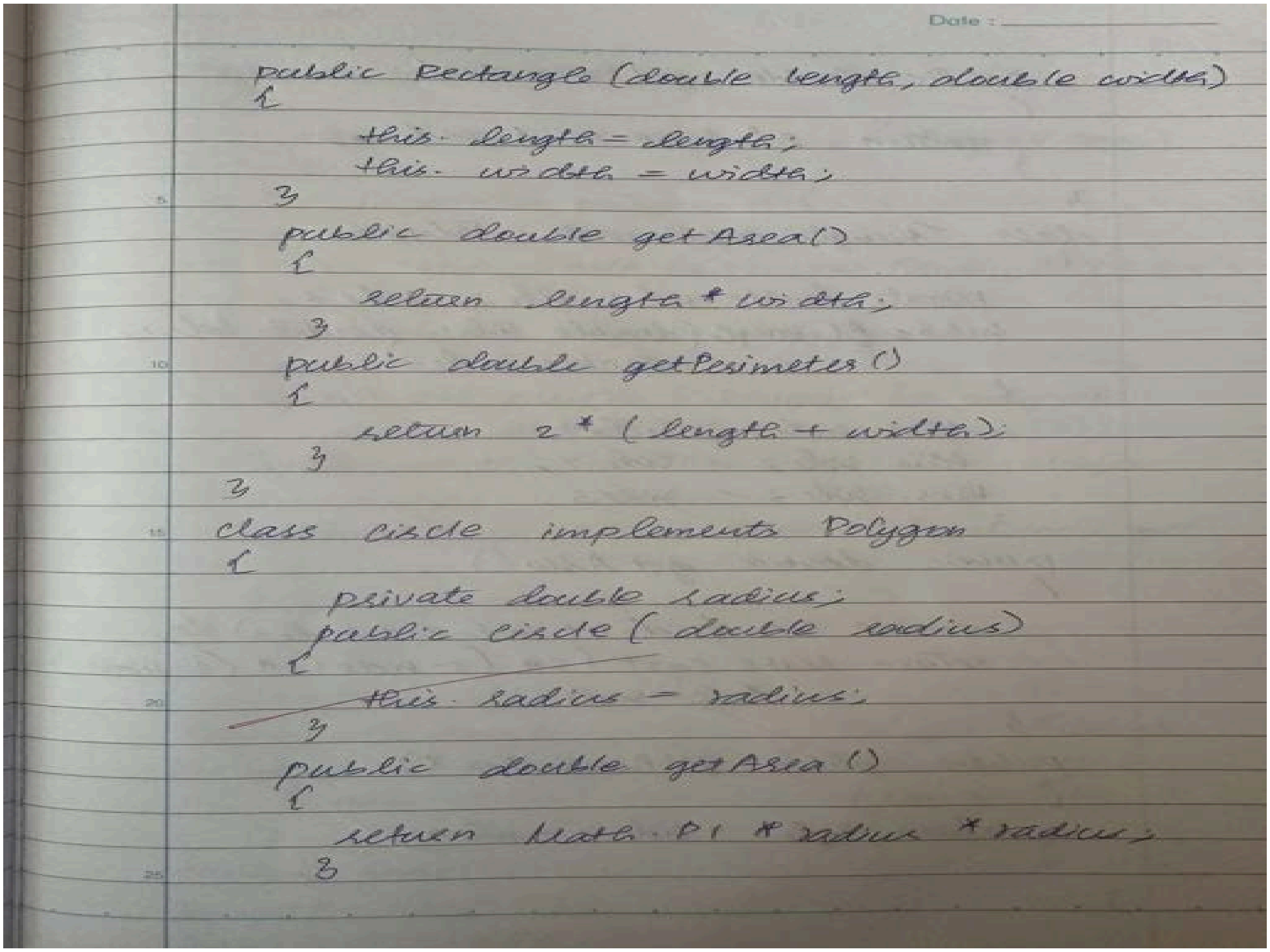
2. Dog barks
Dog eat bones

3. sedan is starting
Sedan is driving.

4 Printing document
Showing document preview

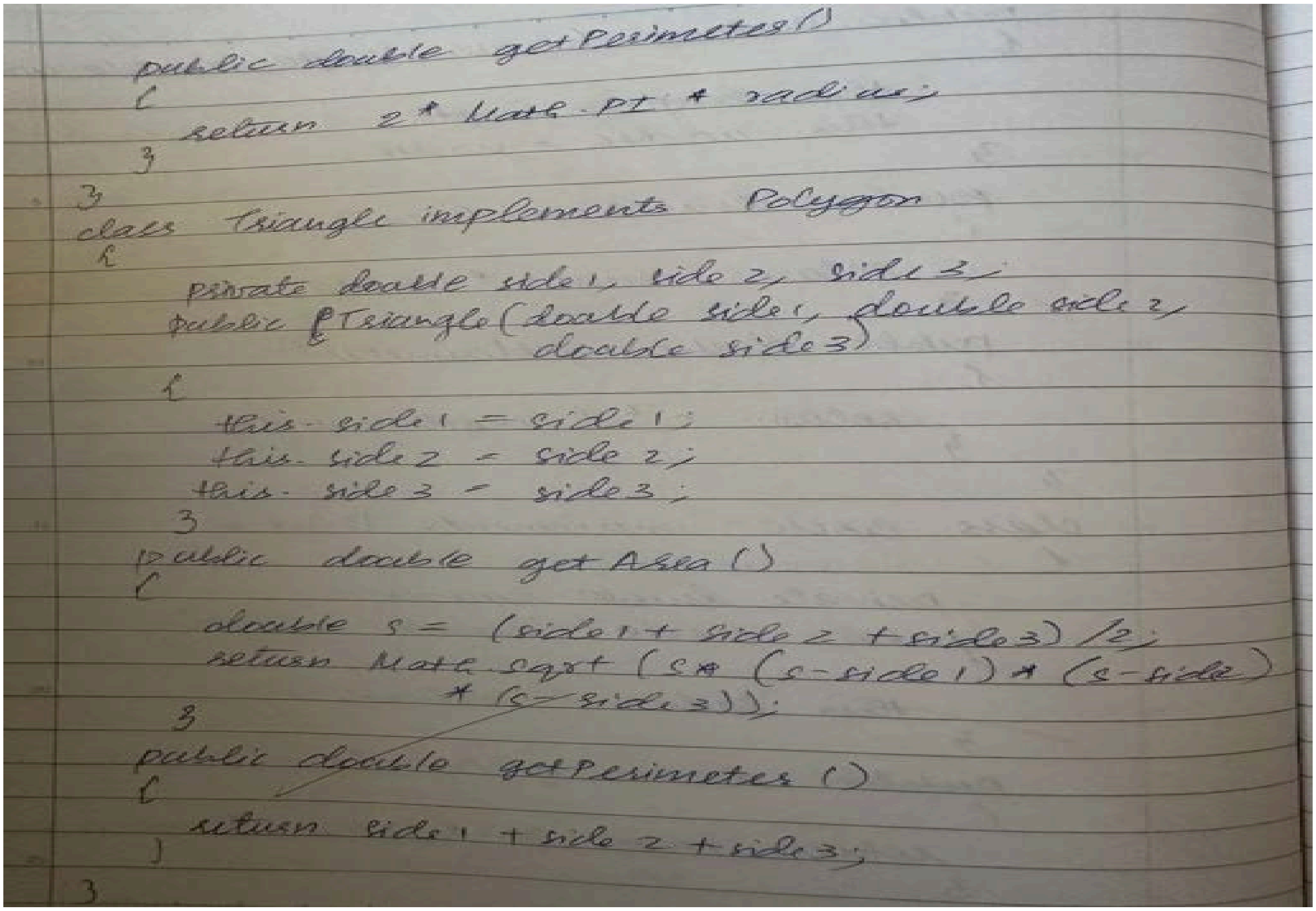
Program

5.
```
interface Polygon
{
    default double getPerimeter ()
    {
        return 0.0;
    }
    double getArea ();
}
class Rectangle implements Polygon
{
        private double length;
        private double width;
```

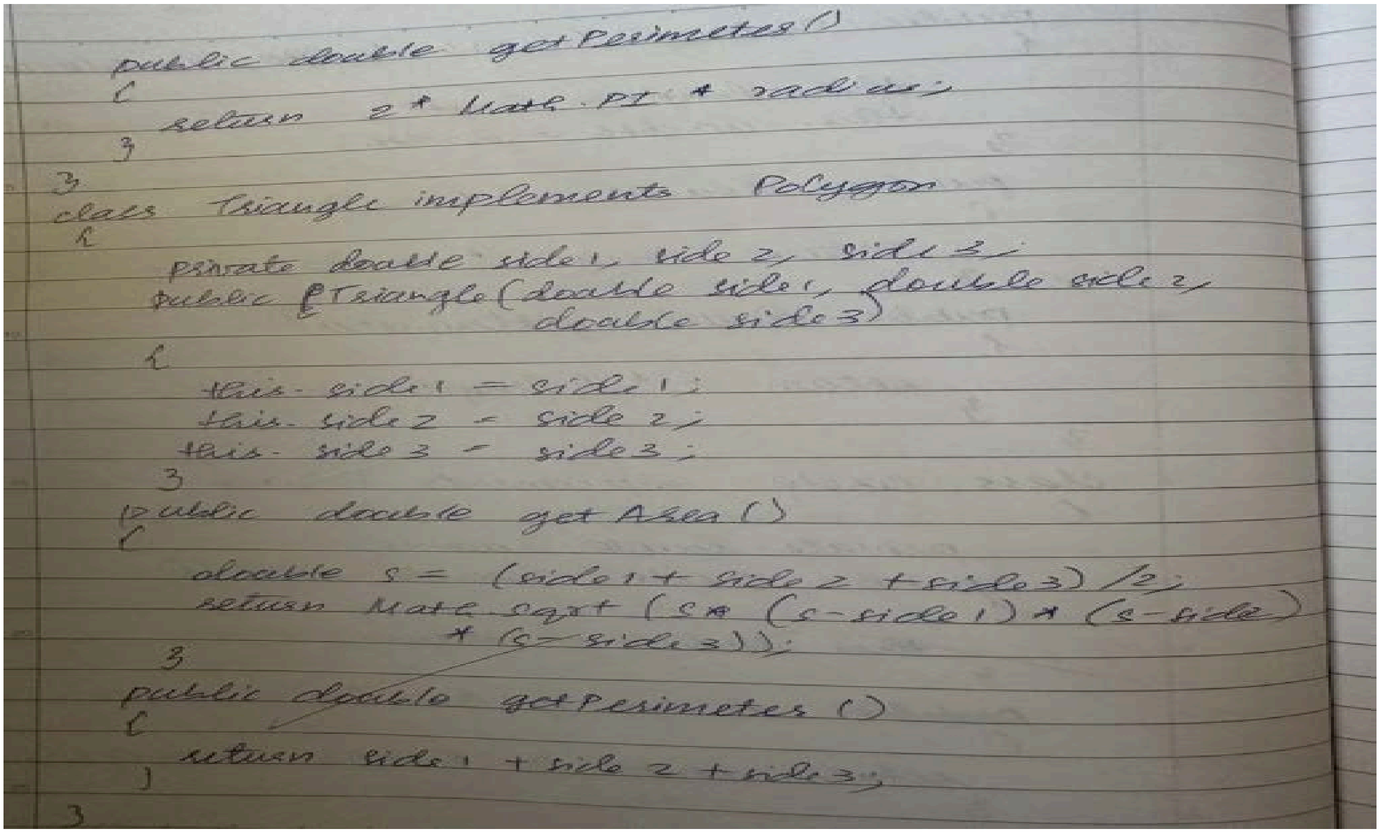


Date :

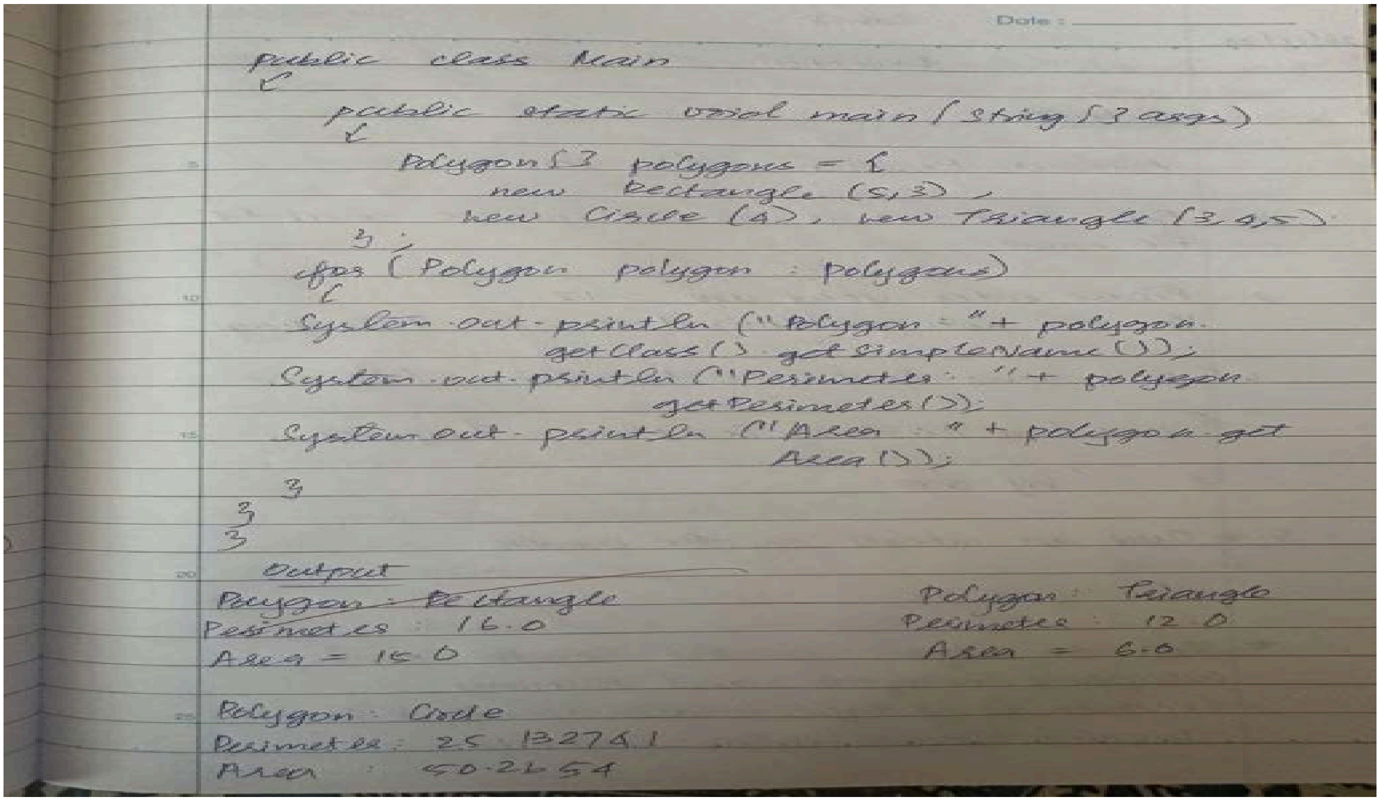
```
    public Rectangle (double length, double width)
    {
        this.length = length;
        this.width = width;
    }
    public double getArea()
    {
        return length * width;
    }
    public double getPerimeter ()
    {
        return 2 * (length + width);
    }
}
class circle implements Polygon
{
    private double radius;
    public circle ( double radius)
    {
        this.radius = radius;
    }
    public double getArea ()
    {
        return Math.PI * radius * radius;
    }
```



```
    public double getPerimeter()
    {
      return 2* Math.PI * radius;
    }
  }
class Triangle implements Polygon
 {
     private double side1, side2, side3;
     public Triangle(double side1, double side2,
                       double side3)
     {
       this.side1 = side1;
       this.side2 = side2;
       this.side3 = side3;
     }
    public double getArea()
    {
       double s = (side1 + side2 + side3) /2;
       return Math.sqrt (s* (s-side1) * (s-side)
                  * (s-side3));
     }
    public double getPerimeter ()
    {
      return side1 + side2 + side3;
    }
 }
```



```
    public double getPerimeter()
    {
      return 2* Math.PI * radius;
    }
  }
class Triangle implements Polygon
 {
     private double side1, side2, side3;
     public Triangle(double side1, double side2,
                       double side3)
     {
       this.side1 = side1;
       this.side2 = side2;
       this.side3 = side3;
     }
    public double getArea()
    {
       double s = (side1 + side2 + side3) /2;
       return Math.sqrt (s* (s-side1) * (s-side)
                  * (s-side3));
     }
    public double getPerimeter ()
    {
      return side1 + side2 + side3;
    }
 }
```

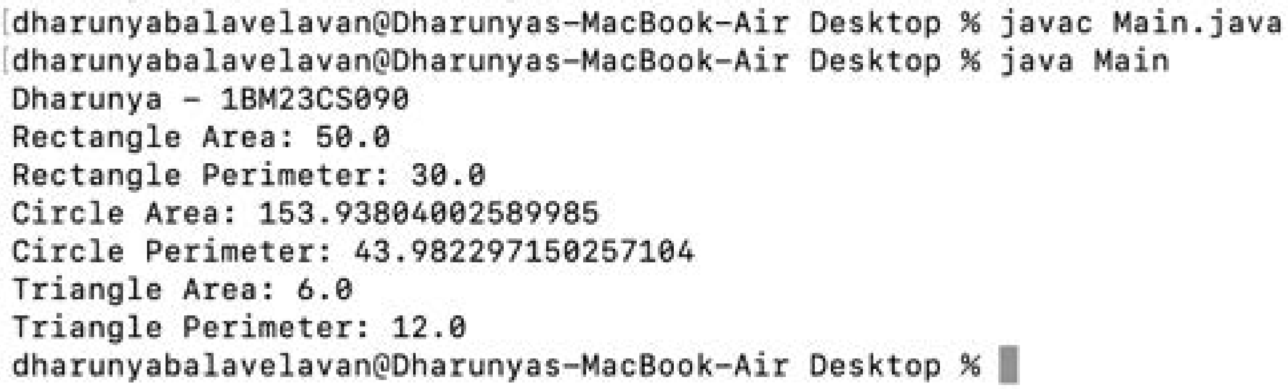


Date :

```
public class Main
{
    public static void main(String[] args)
    {
        Polygon[] polygons = {
            new Rectangle(5,3),
            new Circle(4), new Triangle(3,4,5)
        };
    for (Polygon polygon : polygons)
    {
    System.out.println("Polygon: "+ polygon.
                getClass().getSimpleName());
    System.out.println("Perimeter: "+ polygon
                getPerimeter());
    System.out.println("Area: " + polygon.get
                Area());
    }
  }
}
```

Output

```
Polygon: Rectangle                      Polygon: Triangle
Perimeter: 16.0                         Perimeter: 12.0
Area = 15.0                             Area = 6.0

Polygon: Circle
Perimeter: 25.132741
Area: 50.2654
```



```
dharunyabalavelavan@Dharunyas-MacBook-Air Desktop % javac Main.java
dharunyabalavelavan@Dharunyas-MacBook-Air Desktop % java Main
Dharunya - 1BM23CS090
Rectangle Area: 50.0
Rectangle Perimeter: 30.0
Circle Area: 153.93804002589985
Circle Perimeter: 43.982297150257104
Triangle Area: 6.0
Triangle Perimeter: 12.0
dharunyabalavelavan@Dharunyas-MacBook-Air Desktop %
```