VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Bhoomika M (1BM23CS068)

in partial fulfillment 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
Sep-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 **Bhoomika M (1BM23CS068),** who is bonafide student of **B.M.S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. 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.

Prof.Swathi Sridharan Assistant Professor Department of CSE, BMSCE	Dr.Kavitha Sooda Professor & HOD Department of CSE, BMSCE

Index

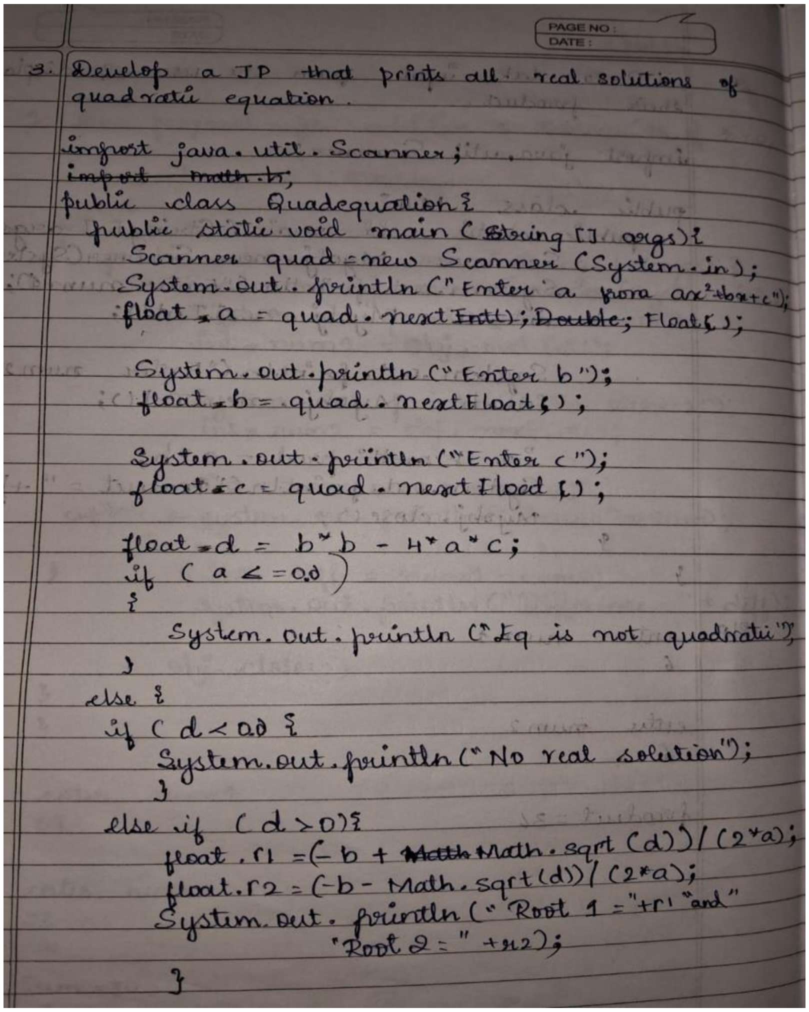
Github Link:
https://github.com/Bhoomika-M-CSE/ooj

Program 1

Implement Quadratic Equation

Code:

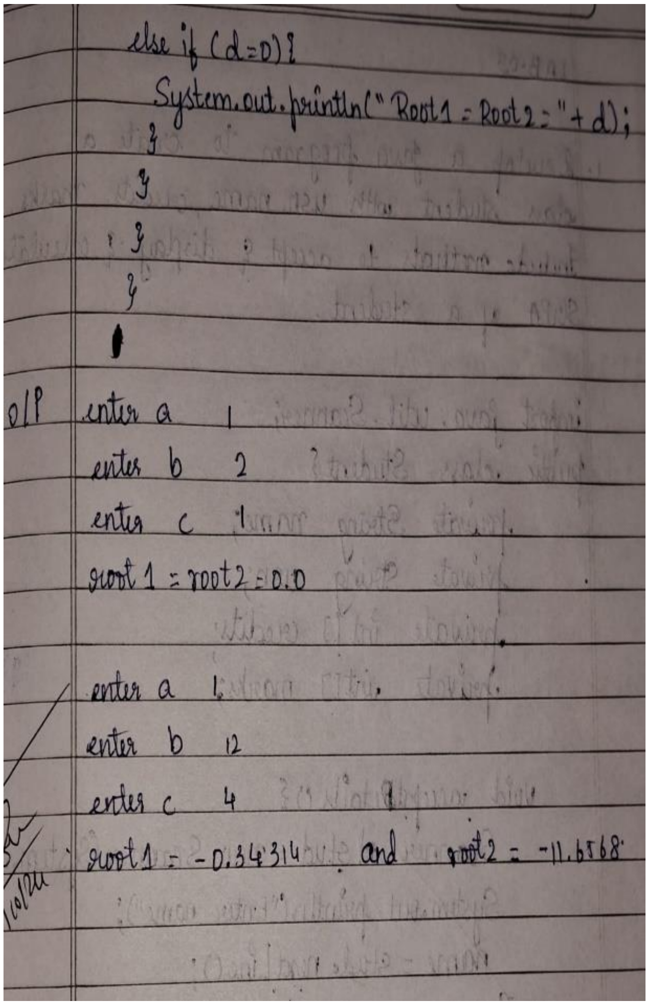
```
import java.util.Scanner;

public class QuadraticEquationSolver {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter the coefficients of the quadratic equation (a, b, c):");
        double a = scanner.nextDouble();
        double b = scanner.nextDouble();
        double c = scanner.nextDouble();
        if (a == 0) {
            System.out.println("This is not a quadratic equation.");
        } else {
            double discriminant = b * b - 4 * a * c;
            if (discriminant > 0) {
                // Two distinct real roots
                double root1 = (-b + Math.sqrt(discriminant)) / (2 * a);
                double root2 = (-b - Math.sqrt(discriminant)) / (2 * a);
                System.out.println("The roots are real and distinct.");
                System.out.println("Root 1: " + root1);
                System.out.println("Root 2: " + root2);
            }
          else if (discriminant == 0){
                double root = -b / (2 * a);
                System.out.println("The roots are real and equal.");
                System.out.println("Root: " + root);
            }
               else {
                double realPart = -b / (2 * a);
                double imaginaryPart = Math.sqrt(-discriminant) / (2 * a);
                System.out.println("The roots are complex and distinct.");
                System.out.println("Root 1: " + realPart + " + " + imaginaryPart + "i");
                System.out.println("Root 2: " + realPart + " - " + imaginaryPart + "i");
            }
        }
    }
  }
```



PAGE NO :
DATE :

3. Develop a JP that prints all real solutions of quadratic equation.

```
import java. util. Scanner;
~~import math.h~~;
public class Quadequation{
 public static void main (String [] args){
   Scanner quad = new Scanner (System.in);
   System.out.println(" Enter a from ax²+bx+c");
   float a = quad. next ~~Int()~~; ~~Double~~; Float();

   System.out.println(" Enter b");
   float b = quad. nextFloat();

   System.out.println("Enter c");
   float c = quad. next Flood ();

   float d = b*b - 4*a*c;
   if ( a <= 0.0 )
   {
      System.out.println("Eq is not quadratic");
   }
  else {
   if ( d < 0.0 {
      System.out.println(" No real solution");
      }
   else if (d > 0){
      float r1 =(-b + ~~Math~~ Math.sqrt (d))/(2*a);
      float r2 = (-b - Math.sqrt(d))/(2*a);
      System.out.println(" Root 1 ="+r1 "and"
                     "Root 2 = " +r2);
      }
```

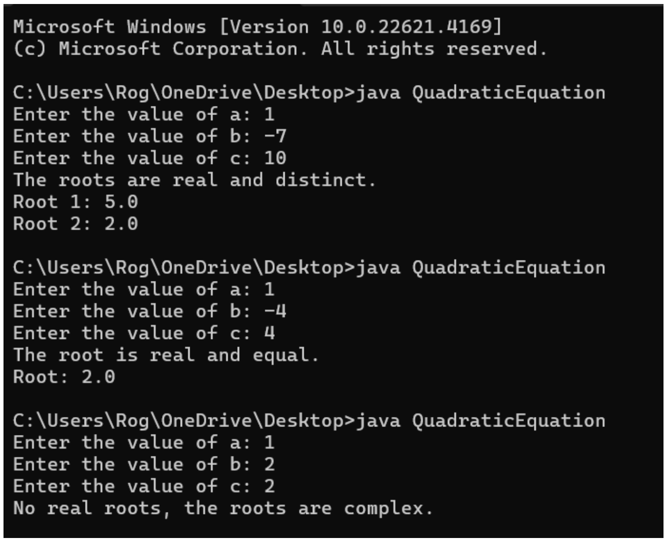


```
    else if (d=0){
       System.out.println(" Root1 = Root2 = "+d);
     }
     }
    }
   }
```

o/p

```
enter a   1
enter b   2
enter c   1
root 1 = root2 = 0.0

enter a   1
enter b   12
enter c   4
root1 = -0.34314 and root2 = -11.6568
```

Output:



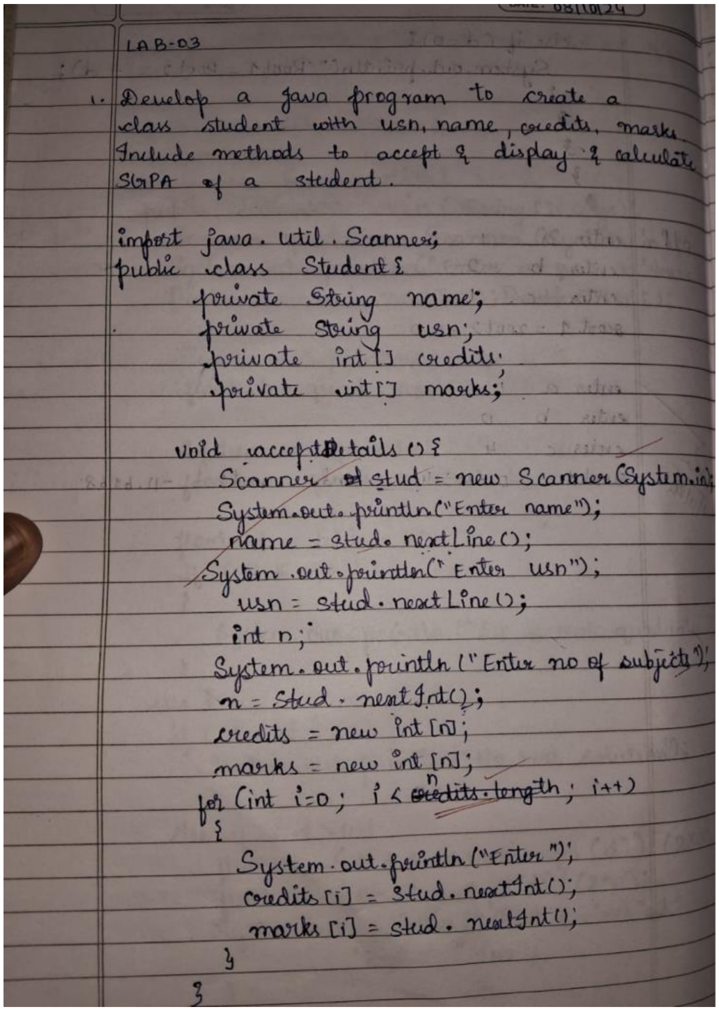
```
Microsoft Windows [Version 10.0.22621.4169]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Rog\OneDrive\Desktop>java QuadraticEquation
Enter the value of a: 1
Enter the value of b: -7
Enter the value of c: 10
The roots are real and distinct.
Root 1: 5.0
Root 2: 2.0

C:\Users\Rog\OneDrive\Desktop>java QuadraticEquation
Enter the value of a: 1
Enter the value of b: -4
Enter the value of c: 4
The root is real and equal.
Root: 2.0

C:\Users\Rog\OneDrive\Desktop>java QuadraticEquation
Enter the value of a: 1
Enter the value of b: 2
Enter the value of c: 2
No real roots, the roots are complex.
```

Program 2
SGPA of a Student

Code:

```
import java.util.Scanner;
class Subject {
  int subM;
  int cred;
  int grade;
  void setSubDet(int marks, int cred) {
    this.subM = marks;
    this.cred = cred;
if (subM >= 90) {
grade = 10; }
else if (subM >= 80) {
grade = 9; }
else if (subM >= 70) {
grade = 8; }
else if (subM >= 60) {
grade = 7; }
else if (subM >= 50) {
grade = 6; }
else if (subM >= 40) {
grade = 5; }
else {
grade = 0;}
}}

class Student {
String name;
String usn;
double SGPA;
Scanner s = new Scanner(System.in);
Subject[] subjects = new Subject[8];

Student() {
for (int i = 0; i < subjects.length; i++) {
subjects[i] = new Subject(); }
}
void getMarks() {
for (int i = 0; i < subjects.length; i++) {
System.out.print("Enter marks for subject " + (i + 1) + ": ");
int marks = s.nextInt();
System.out.print("Enter credit for subject " + (i + 1) + ": ");
int cred = s.nextInt();
subjects[i].setSubDet(marks, cred); }
}

double calSGPA() {
double Score = 0;
int totalCred = 0;
for (Subject subject : subjects) {
Score += (subject.grade * subject.cred);
totalCred += subject.cred; }
if (totalCred > 0) {
SGPA = Score / totalCred; }
else {
SGPA = 0; }
return SGPA;
}

public class StudentDetails {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
System.out.print("Enter number of semesters: ");
int   numSems = sc.nextInt();
Student[] students = new Student[  numSems];
double c=0.0;
String usn,name;
System.out.print("Enter USN: ");
usn = sc.next();
System.out.print("Enter Name: ");
name = sc.next();
for (int i = 0; i <   numSems; i++) {

System.out.println("Enter details for semester" + (i + 1));
students[i] = new Student();
students[i].getMarks();
double s=students[i].calSGPA();
c+=s;
}
c=c/numSems;
for (int i = 0; i <   numSems; i++) {
System.out.println("USN: " + usn);
System.out.println("Name: " + name);
System.out.println("SGPA for sem "+ (i+1)+":" + students[i].calSGPA());}
System.out.println("CGPA: " +c);
}
}
```

```
}
```

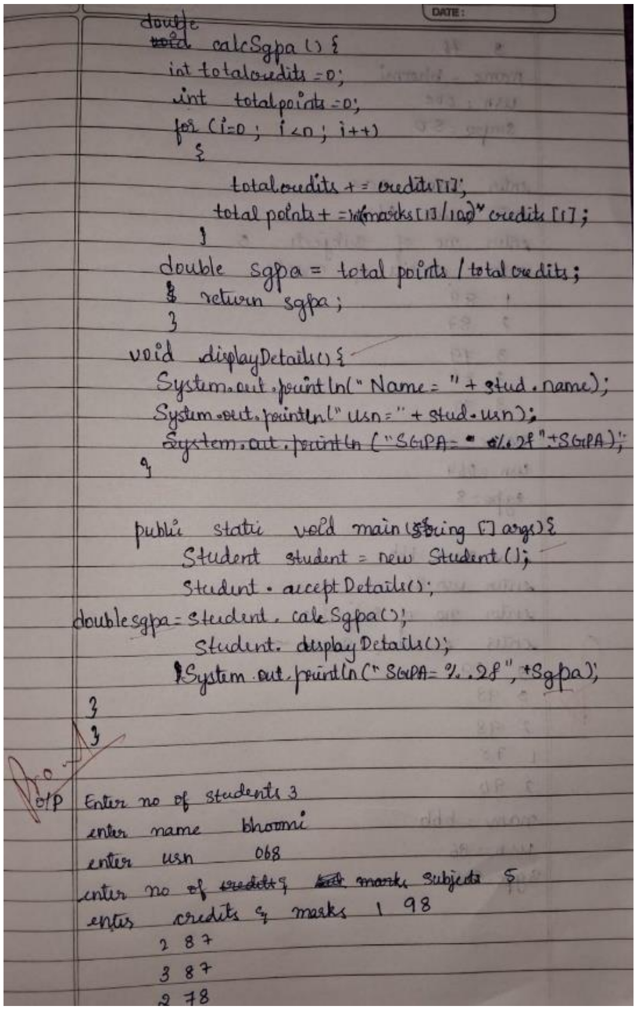


DATE: 08/10/24

LAB-03

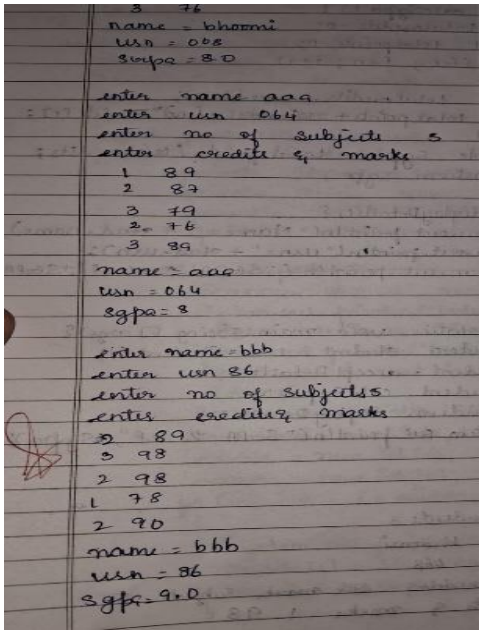
1. Develop a Java program to create a class student with usn, name, credits, marks. Include methods to accept & display & calculate SGPA of a student.

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

    void acceptDetails () {
       Scanner stud = new Scanner (System.in);
       System.out.println("Enter name");
        name = stud. nextLine();
      System.out.println(" Enter usn");
        usn = stud. next Line();
        int n;
       System.out.println ("Enter no of subjects");
       n = Stud. nextInt();
       credits = new int [n];
       marks = new int [n];
     for (int i=0; i < ~~credits.length~~ n; i++)
       {
         System.out.println ("Enter ");
         credits [i] = Stud. nextInt();
          marks [i] = Stud. nextInt();
       }
     }
```



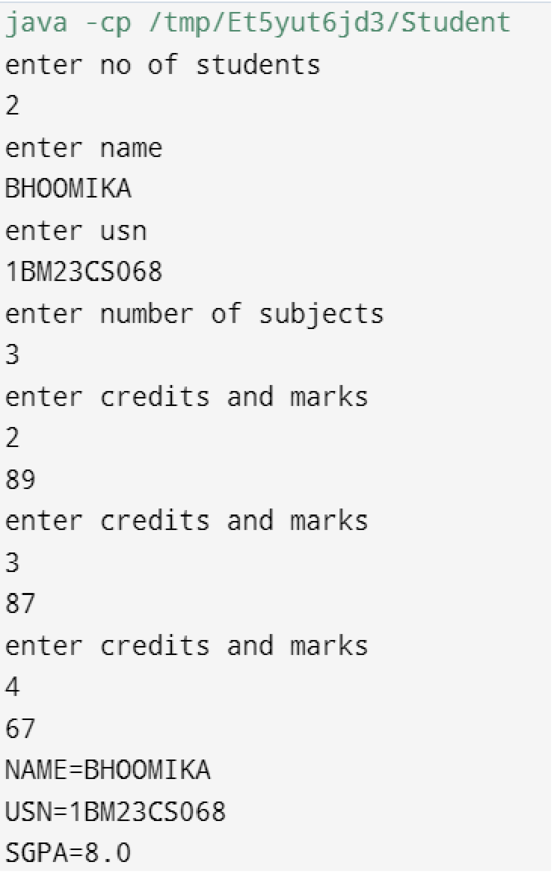
DATE:

```
  double
  ~~void~~ calcSgpa () {
    int totalcredits =0;
    int totalpoints =0;
    for (i=0; i<n; i++)
      {
          totalcredits += credits[i];
        total points + = (marks[i]/10)* credits [i];
      }
    double sgpa = total points / total credits;
    return sgpa;
    }
void displayDetails() {
   System.out.println(" Name = " + stud.name);
   System.out.println(" usn = " + stud.usn);
   ~~System.out.println ("SGPA= %.2f"+SGPA);~~
}


public static void main (String [] args) {
      Student student = new Student ();
      Student. acceptDetails();
doublesgpa = Student. calcSgpa();
       Student. displayDetails();
     System.out.println(" SGPA= %.2f", +Sgpa);
 }
 }

o/p  Enter no of students 3
     enter name      bhoomi
     enter usn       068
    enter no of ~~credits & marks~~ subjects  5
     enter  credits & marks  1 98
                 2 87
                 3 87
                 2 78
```

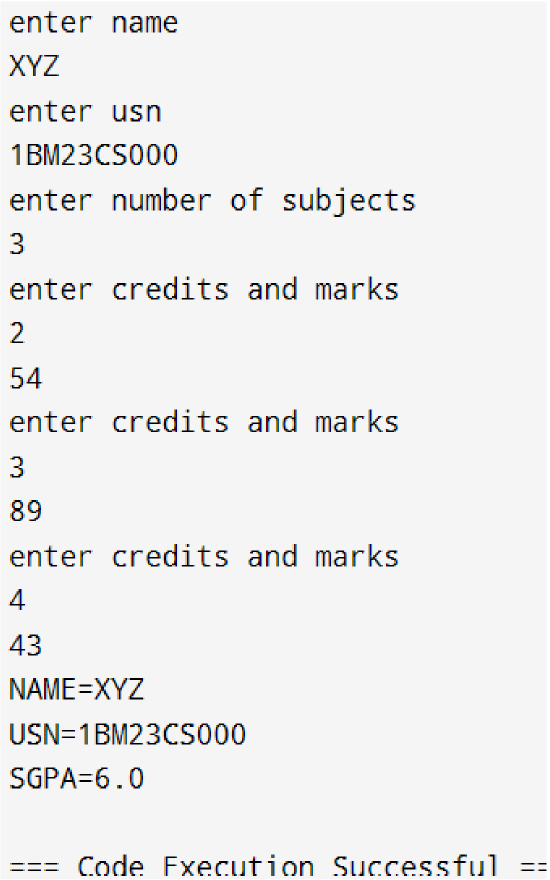


```
   3   76
 name = bhoomi
 usn = 068
 sgpa = 8.0

enter name aaa
enter usn   064
enter no of subjects   5
enter credits & marks
  1  89
  2  87
  3  79
  2  76
  3  89
name = aaa
usn = 064
 sgpa= 8

enter name=bbb
enter usn 86
enter no of subjects5
enter credits& marks
 2  89
 3  98
 2  98
1  78
2  90
name = bbb
usn = 86
sgpa= 9.0
```

Output:



```
java -cp /tmp/Et5yut6jd3/Student
enter no of students
2
enter name
BHOOMIKA
enter usn
1BM23CS068
enter number of subjects
3
enter credits and marks
2
89
enter credits and marks
3
87
enter credits and marks
4
67
NAME=BHOOMIKA
USN=1BM23CS068
SGPA=8.0
```

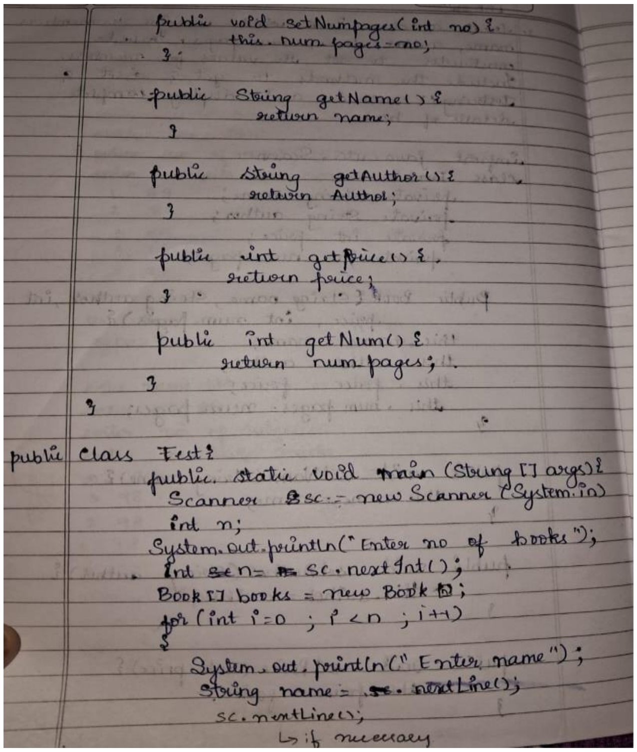


```
enter name
XYZ
enter usn
1BM23CS000
enter number of subjects
3
enter credits and marks
2
54
enter credits and marks
3
89
enter credits and marks
4
43
NAME=XYZ
USN=1BM23CS000
SGPA=6.0

=== Code Execution Successful ==
```

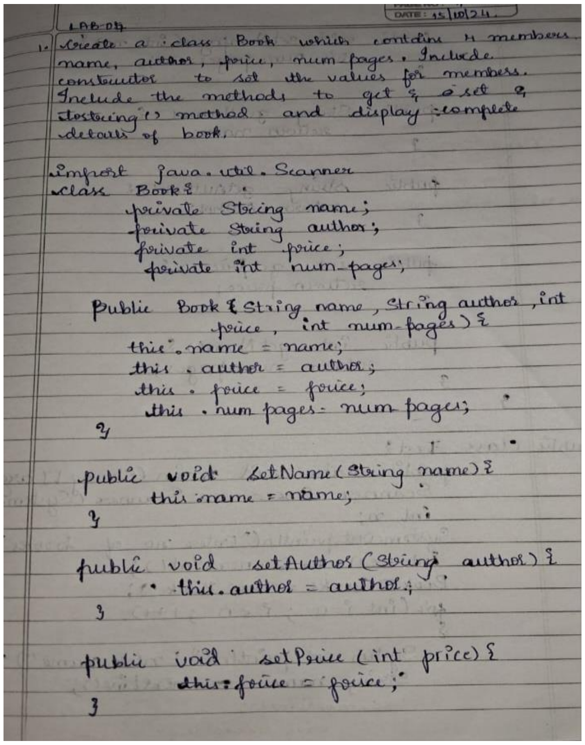
Program 3
Book Information

Code:

```
import java.util.Scanner;

public 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 + "\n";
    }
}

class Main {
    public static void main(String[] args) {
        Scanner ob = new Scanner(System.in);
        System.out.println("Enter number of books:");
        int n = ob.nextInt();
        ob.nextLine();
        Book[] books = new Book[n];
        for (int i = 0; i < n; i++) {
            System.out.println("Enter name of the book " + (i + 1) + ":");
            String name = ob.nextLine();

            System.out.println("Enter name of the author:");
            String author = ob.nextLine();

            System.out.println("Enter price of the book:");
            double price = ob.nextDouble();

            System.out.println("Enter number of pages:");
            int num_pages = ob.nextInt();
            ob.nextLine();

            books[i] = new Book(name, author, price, num_pages);

            System.out.println(books[i].toString());
        }
    }
}
```



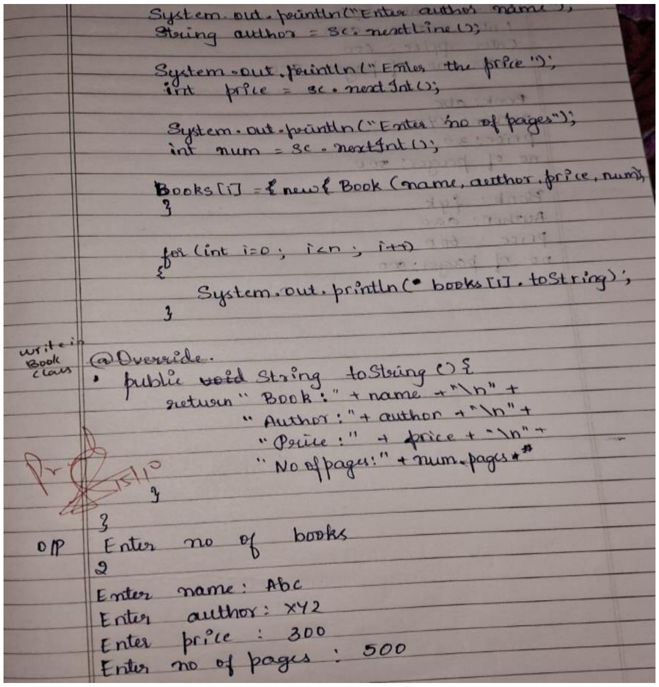
```
        public void setNumpages( int no) {
                this. num_pages = no;
        }

        public String getName() {
                return name;
        }

        public String getAuthor() {
                return Author;
        }

        public int getPrice() {
                return price;
        }

        public int getNum() {
                return num_pages;
        }
    }

public class Test {
        public static void main (String [] args) {
          Scanner sc = new Scanner (System.in)
          int n;
        System.out.println(" Enter no of books");
          int n= sc.nextInt();
        Book [] books = new Book [n];
        for (int i=0 ; i<n ; i++)
        {
            System.out.println(" Enter name");
            String name = . nextLine();
            sc.nextLine();
                  ↳ if necessary
```



LAB-07 DATE : 15/10/24.

1. Create a class Book which contains 4 members. name, author, price, num_pages. Include constructor to set the values for members. Include the methods to get & set & tostring() method and display complete details of book.

```
import java. util. Scanner
class Book {
        private String name;
        private String author;
        private int price;
         private int num_pages;

    Public Book ( String name , String author , int
                price , int num_pages) {
        this. name = name;
        this . auther = 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 (int price) {
         this. price = price;
   }
```



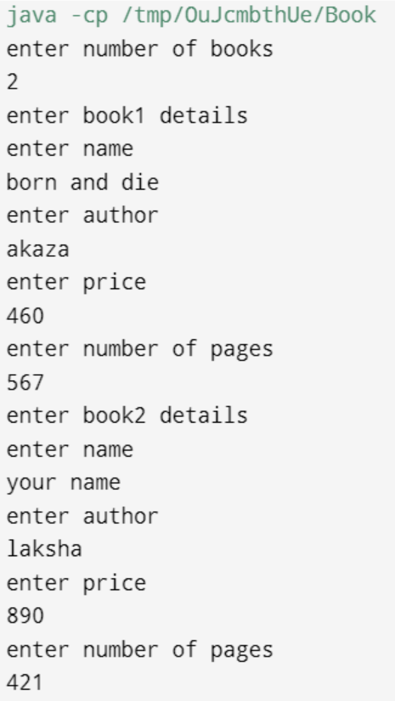
```
System.out.println("Enter author name");
String author = sc.nextLine();

System.out.println("Enter the price");
int   price = sc.nextInt();

System.out.println("Enter no of pages");
int  num = sc.nextInt();

Books[i] = { new{ Book (name, author, price, num);
}

for (int i=0 ; i<n ; i++)
{
    System.out.println(• books[i].toString);
}
```

write in Book class

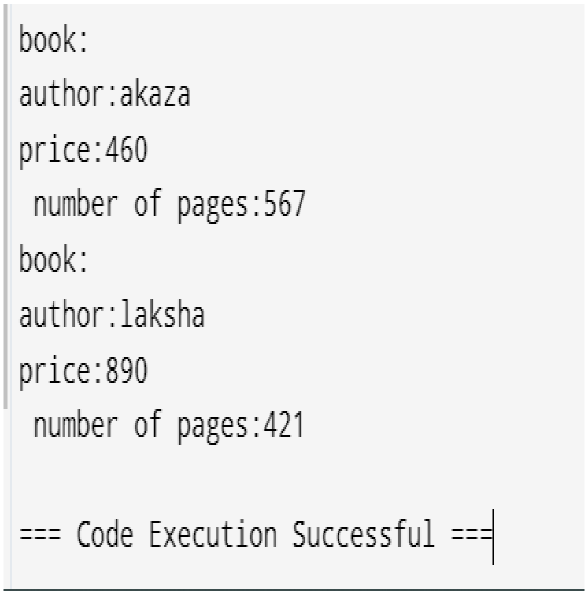
```
@Override.
• public ~~void~~ String  toString () {
    return " Book :" + name + "\n" +
           " Author :" + author + "\n" +
           " Price :"  + price + "\n" +
           " No of pages:" + num.pages + *
  }
}
```

```
O/P  Enter no of books
     2
     Enter name : Abc
     Enter author : XY2
     Enter price : 300
     Enter no of pages : 500
```

Output:



```
java -cp /tmp/OuJcmbthUe/Book
enter number of books
2
enter book1 details
enter name
born and die
enter author
akaza
enter price
460
enter number of pages
567
enter book2 details
enter name
your name
enter author
laksha
enter price
890
enter number of pages
421
```



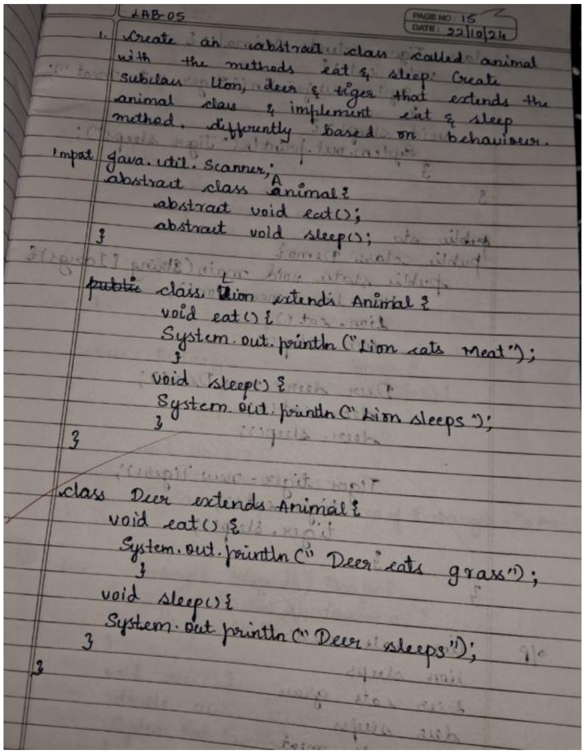
```
book:
author:akaza
price:460
 number of pages:567
book:
author:laksha
price:890
 number of pages:421

=== Code Execution Successful ===
```

Program 4
Abstract Class

Animal Class:

Code:

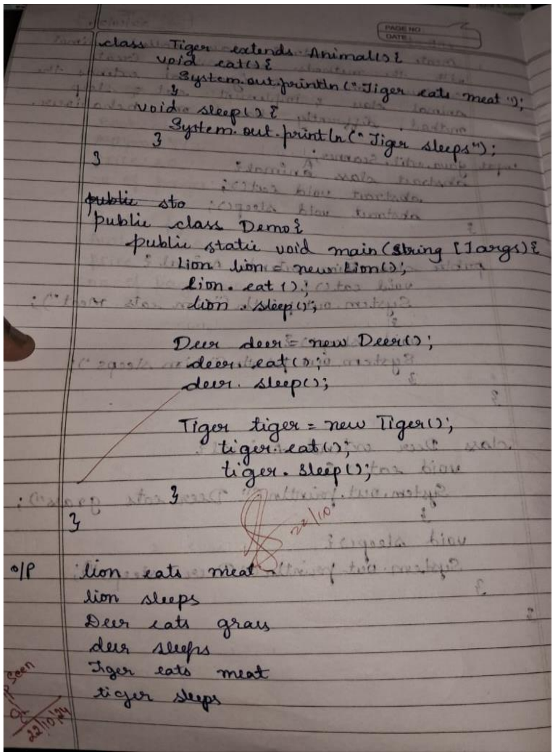
```
abstract class Animal {
   abstract void eatAndSleep();
}

class Lion extends Animal {
   void eatAndSleep() {
      System.out.println("Lion: Hunts for prey, eats meat, and sleeps in a den.");
   }
}

class Deer extends Animal {
   void eatAndSleep() {
      System.out.println("Deer: Grazes on grass, drinks water, and sleeps under trees.");
   }
}

class Tiger extends Animal {
   void eatAndSleep() {
      System.out.println("Tiger: Stalks its prey, eats meat, and rests in dense forests.");
   }
}

public class Main {
   public static void main(String[] args) {
      Animal lion = new Lion();
      Animal deer = new Deer();
      Animal tiger = new Tiger();
      System.out.println("Animal Behaviors:");
      lion.eatAndSleep();
      deer.eatAndSleep();
      tiger.eatAndSleep();
   }
}
```



LAB-05 PAGE NO: 15 DATE: 22|10|24

1. Create an abstract class called animal with the methods eat & sleep. Create subclass Lion, deer & tiger that extends the animal class & implement eat & sleep method. differently based on behaviour.

```
Impat java.util.Scanner;
abstract class Animal{
        abstract void eat();
        abstract void sleep();
}

~~public~~ class Lion extends Animal{
        void eat(){
        System.out.println("Lion eats Meat");
         }
        void sleep(){
        System.out.println("Lion sleeps");
        }
 }

class Deer extends Animal{
     void eat(){
      System.out.println(" Deer eats grass");
       }
    void sleep(){
    System.out.println("Deer sleeps");
   }
}
```



```
class Tiger extends Animal(){
      void eat(){
        System.out.println("Tiger eats meat");
        }
    void sleep(){
        System.out.println("Tiger sleeps");
      }
}

~~public~~ sto
public class Demo{
    public static void main(String []args){
        Lion lion = new Lion();
        lion.eat();
        lion.sleep();

        Deer deer = new Deer();
        deer.eat();
        deer.sleep();

        Tiger tiger = new Tiger();
          tiger.eat();
          tiger.sleep();
    }
}
```

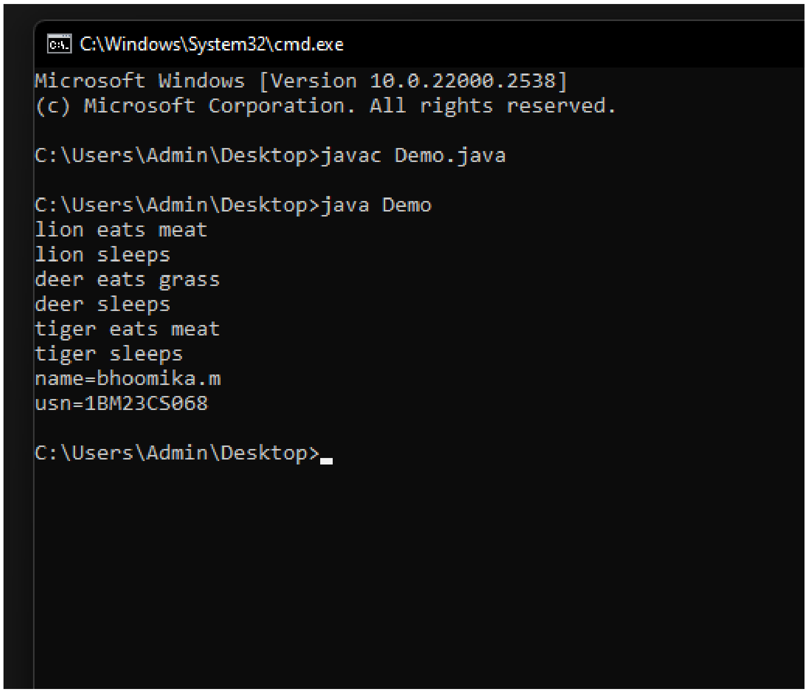
22/10

o/P

```
lion eats meat
lion sleeps
Deer eats grass
deer sleeps
Tiger eats meat
tiger sleeps
```

Seen 22/10/24

Output:



C:\Windows\System32\cmd.exe

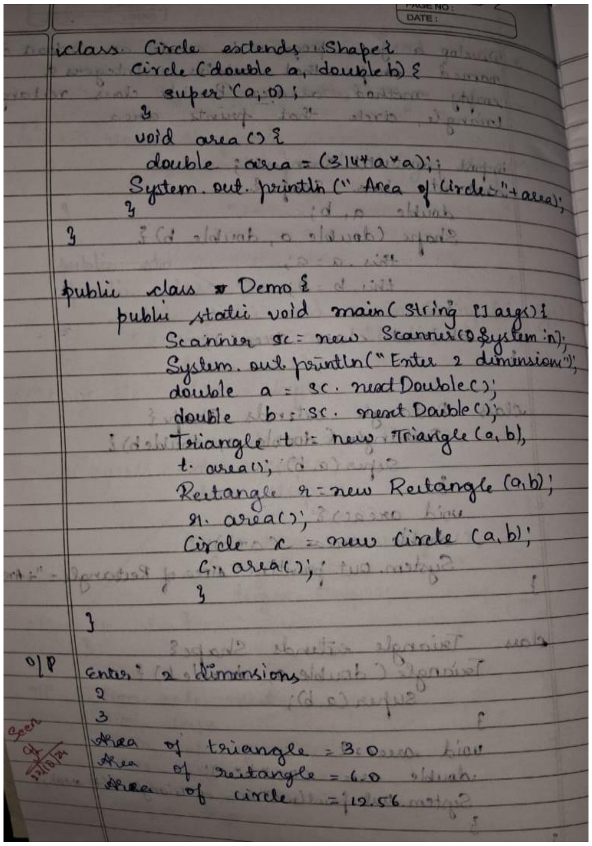
```
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop>javac Demo.java

C:\Users\Admin\Desktop>java Demo
lion eats meat
lion sleeps
deer eats grass
deer sleeps
tiger eats meat
tiger sleeps
name=bhoomika.m
usn=1BM23CS068

C:\Users\Admin\Desktop>
```

Shape class:

Code:

```
import java.util.Scanner;
```

```
interface Polygon{
  default double getPerimeter(){
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter number of sides:");
    int n=sc.nextInt();
    double p=0;
    for(int i=0;i<n;i++){
      System.out.println("Enter side:");
      double a=sc.nextDouble();
      p+=a;
   }
  return p;
}
  abstract double getArea();
}

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

public double getArea(){
  double area=length*breadth;
return area;
}
}

class Triangle implements Polygon{
  public double length1;
  public double breadth1;
 public Triangle(double length1,double breadth1){
  this.length1=length1;
  this.breadth1=breadth1;}

public double getArea(){
  double area=0.5*length1*breadth1;
return area;
}
}

public class Shape{
  public static void main(String []Args){
  Scanner sc=new Scanner(System.in);
  System.out.println("Bhoomika BG-1BM23CS067");
do{
  System.out.println("Choose:\n 1.Rectangle\n 2.Triangle");
  int x=sc.nextInt();
  switch(x){
  case(1):
    {
    System.out.println("Enter length:");
    double l=sc.nextDouble();
    System.out.println("Enter breadth:");
    double b=sc.nextDouble();
    Polygon rc=new Rectangle(l,b);
    double p=rc.getPerimeter();
    System.out.println("Perimeter of rectangle is:"+p);
    double a=rc.getArea();
    System.out.println("Area of rectangle is:"+a);
    break;
  }
  case(2):{
    System.out.println("Enter base:");
    double l=sc.nextDouble();
    System.out.println("Enter height:");
    double b=sc.nextDouble();
    Polygon tr=new Triangle(l,b);
    double p1=tr.getPerimeter();
    System.out.println("Perimeter of triangle is:"+p1);
    double a1=tr.getArea();
    System.out.println("Area of triangle is:"+a1);
    break;
   }
  default:
    System.exit(0);
}
}while(true);
}
}
```



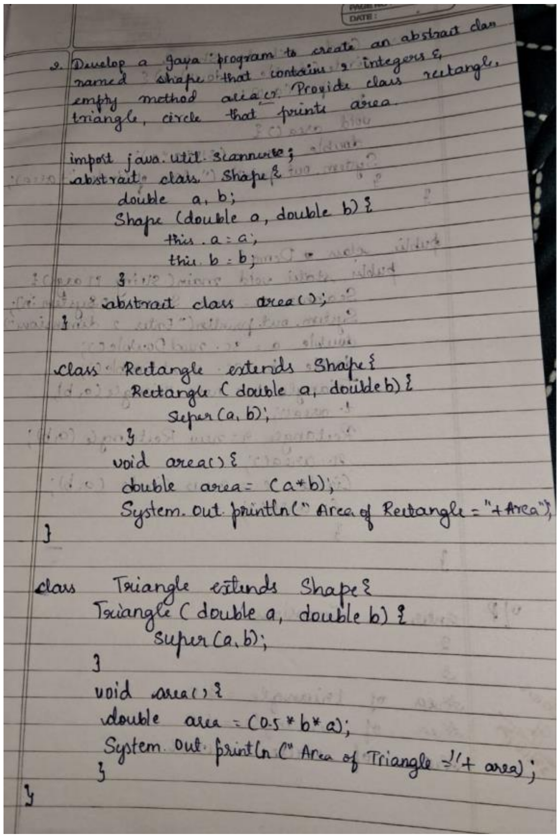
PAGE NO :
DATE :

```
class Circle extends Shape{
    Circle (double a, double b) {
        super (a, 0) ;
    }
    void area () {
     double area = (3.14* a * a);
    System. out. println (" Area of Circle="+area);
    }
}

public class Demo {
    public static void main( String [] args) {
        Scanner sc = new Scanner(System.in);
        System. out. println(" Enter 2 dimension");
        double a = sc. nextDouble();
        double b = sc. next Double();
        Triangle t = new Triangle (a, b),
         t. area();
         Rectangle r = new Rectangle (a,b);
          r. area();
         Circle c = new Circle (a, b);
          c. area();
          }
}
```

O/P

```
Enter 2 dimensions
 2
 3
Area of triangle = 3.0
Area of rectangle = 6.0
Area of circle = 12.56
```

Seen 22/10/24

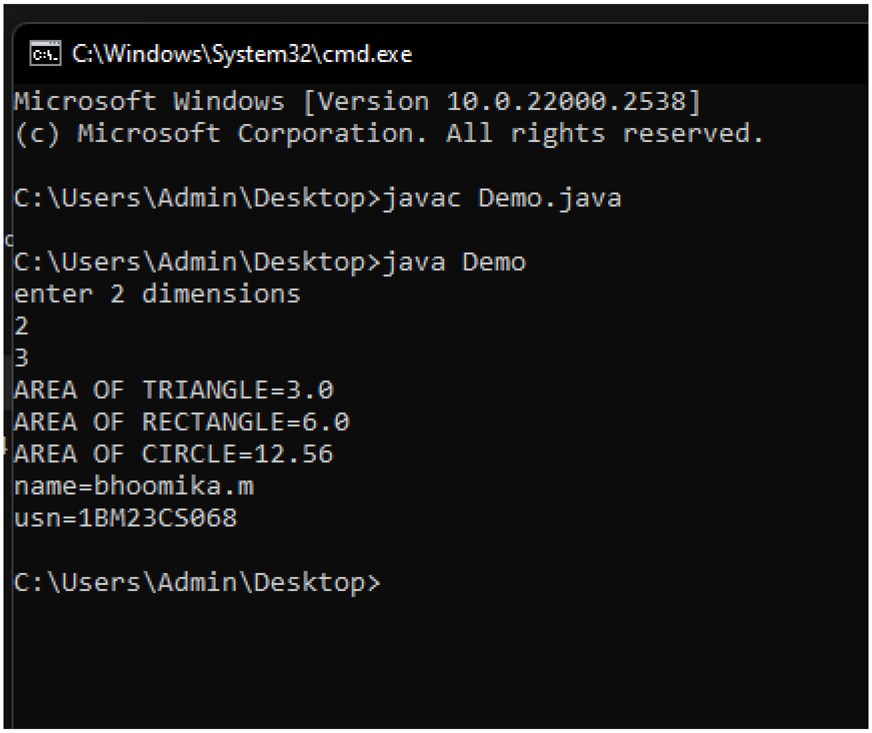


DATE :

2. Develop a Java program to create an abstract class named shape that contains 2 integers & empty method area(). Provide class rectangle, triangle, circle that prints area.

```
import java.util.scanner;
abstract class Shape{
        double a, b;
        Shape (double a, double b) {
             this . a = a;
             this. b = b;
        }
      abstract class area();
}

class Rectangle extends Shape{
        Rectangle (double a, double b) {
             Super (a, b);
        }
       void area() {
        double area= (a*b);
        System. out. println(" Area of Rectangle ="+ Area");
}

class Triangle extends Shape{
     Triangle (double a, double b) {
           super (a, b);
     }
     void area() {
     double area = (0.5 * b * a);
      System. out. print ln (" Area of Triangle ="+ area);
     }
}
```

Output:

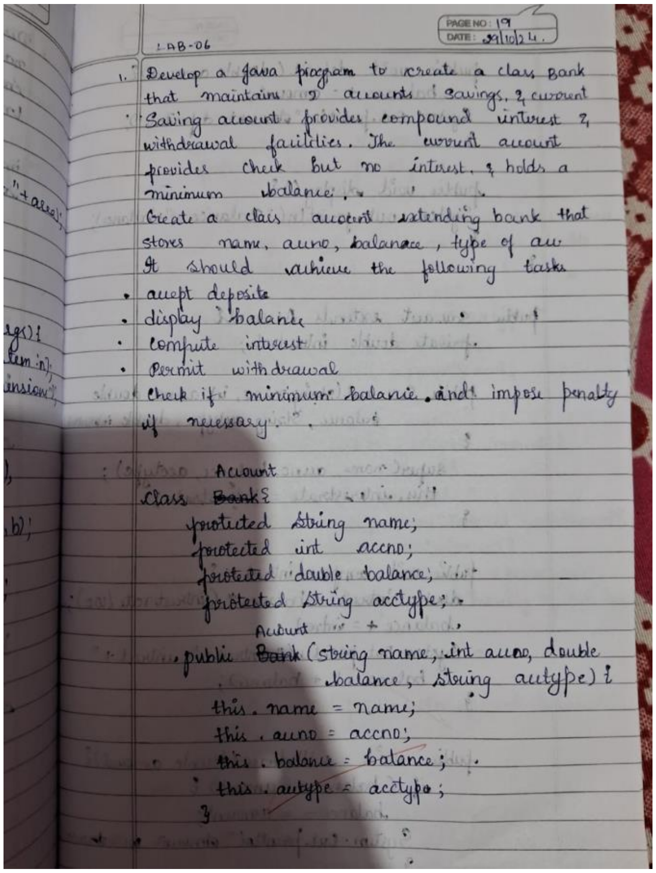


```
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop>javac Demo.java

C:\Users\Admin\Desktop>java Demo
enter 2 dimensions
2
3
AREA OF TRIANGLE=3.0
AREA OF RECTANGLE=6.0
AREA OF CIRCLE=12.56
name=bhoomika.m
usn=1BM23CS068

C:\Users\Admin\Desktop>
```

Program 5
Bank Class

Code:

```
import java.util.Scanner;

class Account {
   private String customerName;
   private int accountNumber;
   private double balance;

   public Account(String customer_name, int acc_no, double balance) {
      this.customer_name = customer_name;
      this.acc_no = acc_no;
      this.balance = balance;
   }

   public double getBalance() {
      return balance;
   }

   public void deposit(double amount) {
      if (amount > 0) {
         balance += amount;
         System.out.println("Deposited: " + amount);
         System.out.println("Your new balance is:"+balance);
      } else {
         System.out.println("Deposit amount must be positive.");
      }
   }

   public void displayBalance() {
      System.out.println("Current Balance: " + balance);
   }
}

class SavingsAccount extends Account {
   private double interestRate;

   public SavingsAccount(String customerName, int accountNumber, double initialBalance, double
interestRate) {
      super(customerName, accountNumber, initialBalance);
      this.interestRate = interestRate;
   }

   public void computeAndDepositInterest() {
      double interest = getBalance() * interestRate / 100;
      deposit(interest);
      System.out.println("Balance is: "+balance);
      System.out.println("Interest of " + interest + " has been credited.");
   }
   public void withdraw(double amount)
     {
       if(amount<=getBalance()){
          double new_balance=getBalance()-amount;
          System.out.println("withdrew:"+amount + " balance is:"+ new_balance);
          }
       else
         System.out.println("Insufficient funds!!");
     }
}

class CurrentAccount extends Account {
    private double minimumBalance;
    private double serviceCharge;

    public CurrentAccount(String customerName, int accountNumber, double initialBalance, double
minimumBalance, double serviceCharge) {
        super(customerName, accountNumber, initialBalance);
        this.minimumBalance = minimumBalance;
        this.serviceCharge = serviceCharge;
    }
    public void withdraw(double amount) {
        if (amount <= getBalance()) {
            double newBalance = getBalance() - amount;
            System.out.println("Withdrew: " + amount);
        } else {
            System.out.println("Insufficient balance.");
        }
    }

    private void checkMinimumBalance() {
        if (getBalance() < minimumBalance) {
            System.out.println("Balance is below minimum");
            balance-=serviceCharge;
            System.out.println("Deducted service charge:" +serviceCharge);
            System.out.println("Balance after deduction is":+balance);
        }
    }
}

public class Bank {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("enter customer name:");
        String name=sc.nextLine();
        System.out.println("enter accno:");
        int acc_no=sc.nextInt();
        System.out.println("enter initial balance:");
        double balance=sc.nextDouble();
        System.out.println("enter minimum balance:");
        double minimum_balance=sc.nextDouble();
        System.out.println("enter interest rate:");
    double interest_rate=sc.nextDouble();
    System.out.println("Enter choice:\n 1.Current acc\n 2.Savings acc");
    int ch=sc.nextInt();
    switch(ch){
      case(1):
        System.out.println("account is current type");
        CurrentAccount cu=new CurrentAccount(name,acc_no,balance,minimum_balance,interest_rate);
       do{ System.out.println("enter choice:\n 1.deposit\n 2.withdraw\n 3.display balance");
        int c=sc.nextInt();
        cu.checkMinimumBalance()
        if(c==1){
          System.out.println("enter amount to be deposited:");
          double amt=sc.nextDouble();
           cu.deposit(amt);}
        else if(c==2){
          System.out.println("enter amount to withdraw:");
          double amt=sc.nextDouble();
          cu.withdraw(amt);}
        else if(c==3){
          cu.displayBalance();}
        else
         System.out.println("invalid entry!!");
         exit(0);
         }while(true);

     case(2):
         System.out.println("account is savings type");
        SavingsAccount sa=new SavingsAccount(name,acc_no,balance,interest_rate);
        do{ System.out.println("enter choice:\n 1.deposit\n 2.withdraw\n 3.display balance");
        int c1=sc.nextInt();
        if(c1==1){
          System.out.println("enter amount to be deposited:");
          double amt=sc.nextDouble();
           sa.deposit(amt);}
        else if(c1==2){
          System.out.println("enter amount to withdraw:");
          double amt=sc.nextDouble();
          sa.withdraw(amt);}
        else if(c1==3){
          sa.displayBalance();}
        else{
         System.out.println("invalid entry!!");
         exit(0);}
         sa.computeAndDepositInterest();
        }while(true);

    sc.close();
  }
}
}
```



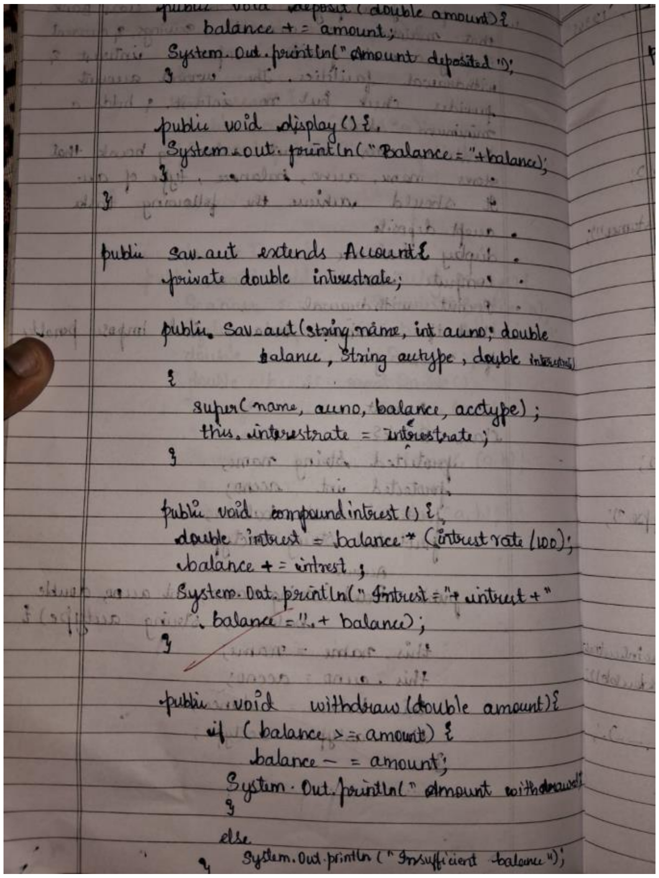
LAB-06

PAGE NO: 19
DATE: 29/10/24.

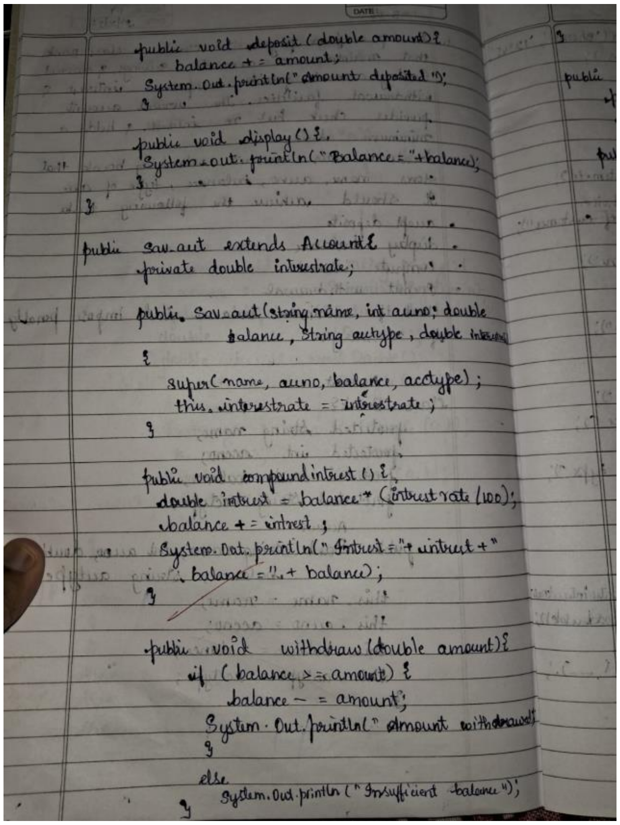
1. Develop a java program to create a class Bank that maintains 2 accounts savings. & current Saving account provides compound interest & withdrawal facilities. The current account provides check but no interest. & holds a minimum balance.
Create a class account extending bank that stores name, accno, balanace, type of acc It should achieve the following tasks
- accept deposite
- display balance
- compute interest
- Permit withdrawal

Check if minimum balance and impose penalty if necessary.

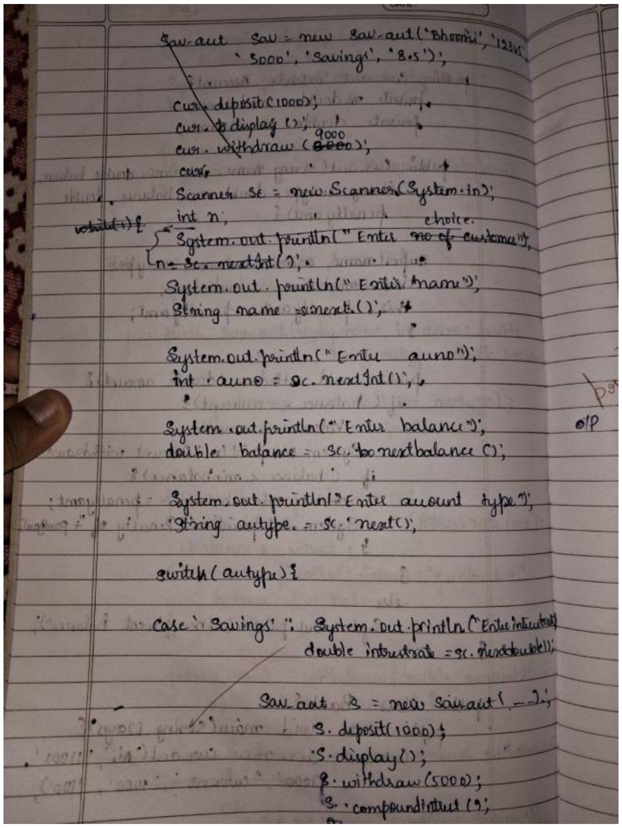
```
           Account
Class  ~~Bank~~{
      protected String name;
      protected int accno;
      protected double balance;
      protected String acctype;
                  Account
     public ~~Bank~~(string name, int accno, double
                    balance, string acctype){
       this. name = name;
       this . accno = accno;
       this . balance = balance;
       this . acctype = acctypo;
      }
```



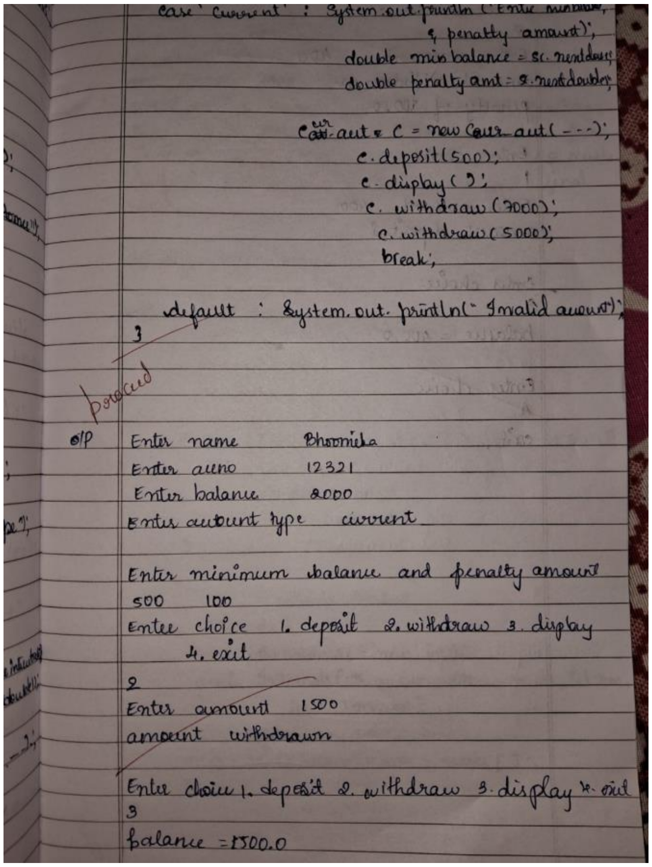
```
  public void deposit(double amount){
        balance + = amount;
     System. Out. println("Amount deposited ");
     }

  public void display(){
     System.out. println("Balance = "+balance);
   }
}

public  Sav-acct extends Account{
   private double interestrate;

   public Sav-acct(string name, int accno; double
               balance, String acctype, double interestrate)
   {
      super(name, accno, balance, acctype);
      this. interestrate = interestrate;
   }

   public void compoundintrest(){
     double intrest = balance * (intrest rate /100);
     balance + = intrest;
     System. Out. println(" Intrest = "+ intrest + "
        balance = ". + balance);
   }

   public void  withdraw(double amount){
       if (balance >= amount) {
            balance - = amount;
         System. Out. println(" Amount withdrawal
         }
         else
     }    System.Out.println("Insufficient balance");
```



```
  public void deposit(double amount){
        balance + = amount;
     System. Out. println("Amount deposited ");
     }

  public void display(){
     System.out. println("Balance = "+balance);
   }
}

public  Sav-acct extends Account{
   private double interestrate;

   public Sav-acct(string name, int accno; double
               balance, String acctype, double interestrate)
   {
      super(name, accno, balance, acctype);
      this. interestrate = interestrate;
   }

   public void compoundintrest(){
     double intrest = balance * (intrest rate /100);
     balance + = intrest;
     System. Out. println(" Intrest = "+ intrest + "
        balance = ". + balance);
   }

   public void  withdraw(double amount){
       if (balance >= amount) {
            balance - = amount;
         System. Out. println(" Amount withdrawal
         }
         else
     }    System.Out.println("Insufficient balance");
```



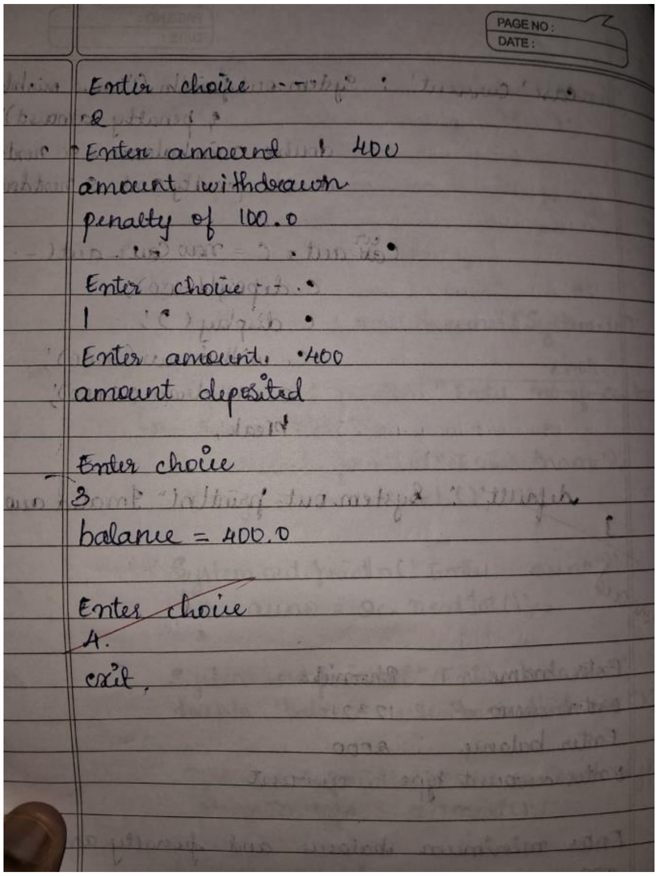
```
Sav-acct  Sav = new  Sav-acct('Bhoomi', '1234',
              '5000', 'Savings', '8.5');

   cur. deposit(1000);
   cur. display();
                 9000
   cur. withdraw(~~8000~~);
   cur.
   Scanner sc = new Scanner(System.in);
   int n;                        choice.
~~while(1){~~ System.out.println("Enter ~~no of customer~~");
   ~~n = sc. nextInt();~~
   System. out. println("Enter name");
   String name = sc.next();

   System.out.println("Enter accno");
   int accno = sc. nextInt();

   System.out.println("Enter balance");               o/p
   double balance = sc.nextbalance();

   System.out.println("Enter account type");
   String acctype. = sc.next();

  switch(acctype){

  case 'Savings' : System.out.println("Enter interestrate
                    double intrestrate = sc. nextdouble();

                Sav-acct  s = new savacct(_.._);
                      s. deposit(1000);
                      s. display();
                      s. withdraw(5000);
                      s. compoundintrest();
```



```
case 'current' : System.out.println("Enter minbalance
                         & penalty amount");
                 double minbalance = sc.nextdouble;
                 double penaltyamt = s.nextdouble;

               cur-acct c = new Curr-acct(---);
                    c.deposit(500);
                    c.display();
                    c.withdraw(7000);
                    c.withdraw(5000);
                    break;

     default : System.out.println("Invalid account");
}
```

o/p

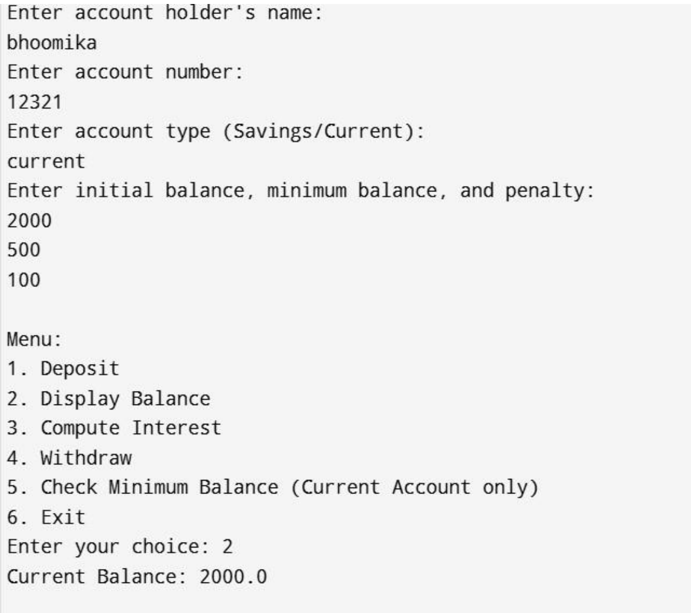
```
Enter name        Bhoomika
Enter accno       12321
Enter balance     2000
Enter account type    current

Enter minimum balance and penalty amount
500    100
Enter choice   1. deposit  2. withdraw  3. display
      4. exit
2
Enter amount   1500
amount withdrawn

Enter choice 1. deposit 2. withdraw 3. display 4. exit
3
balance = 1500.0
```



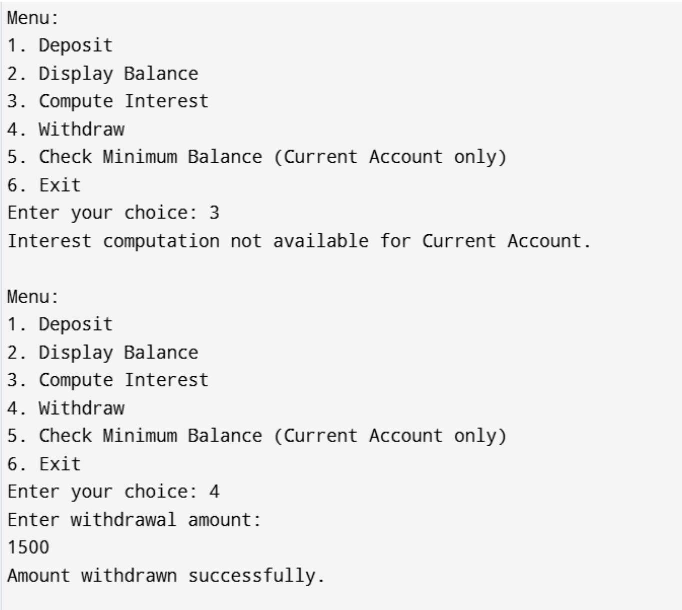
PAGE NO :
DATE :

```
Enter choice
2
Enter amount   400
amount withdrawn
penalty of 100.0

Enter choice
1
Enter amount  400
amount deposited

Enter choice
3
balance = 400.0

Enter choice
4.
exit.
```

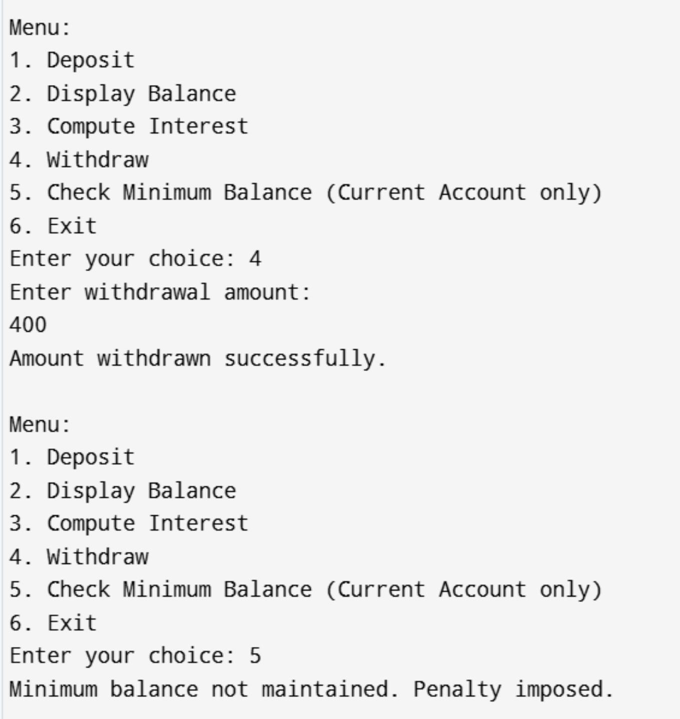
Output:



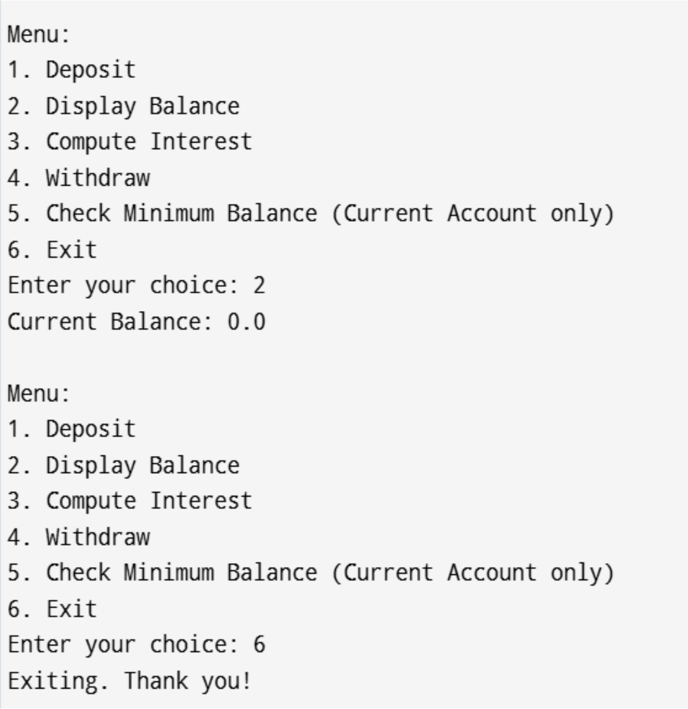
```
Enter account holder's name:
bhoomika
Enter account number:
12321
Enter account type (Savings/Current):
current
Enter initial balance, minimum balance, and penalty:
2000
500
100

Menu:
1. Deposit
2. Display Balance
3. Compute Interest
4. Withdraw
5. Check Minimum Balance (Current Account only)
6. Exit
Enter your choice: 2
Current Balance: 2000.0
```



```
Menu:
1. Deposit
2. Display Balance
3. Compute Interest
4. Withdraw
5. Check Minimum Balance (Current Account only)
6. Exit
Enter your choice: 3
Interest computation not available for Current Account.

Menu:
1. Deposit
2. Display Balance
3. Compute Interest
4. Withdraw
5. Check Minimum Balance (Current Account only)
6. Exit
Enter your choice: 4
Enter withdrawal amount:
1500
Amount withdrawn successfully.
```



```
Menu:
1. Deposit
2. Display Balance
3. Compute Interest
4. Withdraw
5. Check Minimum Balance (Current Account only)
6. Exit
Enter your choice: 4
Enter withdrawal amount:
400
Amount withdrawn successfully.

Menu:
1. Deposit
2. Display Balance
3. Compute Interest
4. Withdraw
5. Check Minimum Balance (Current Account only)
6. Exit
Enter your choice: 5
Minimum balance not maintained. Penalty imposed.
```

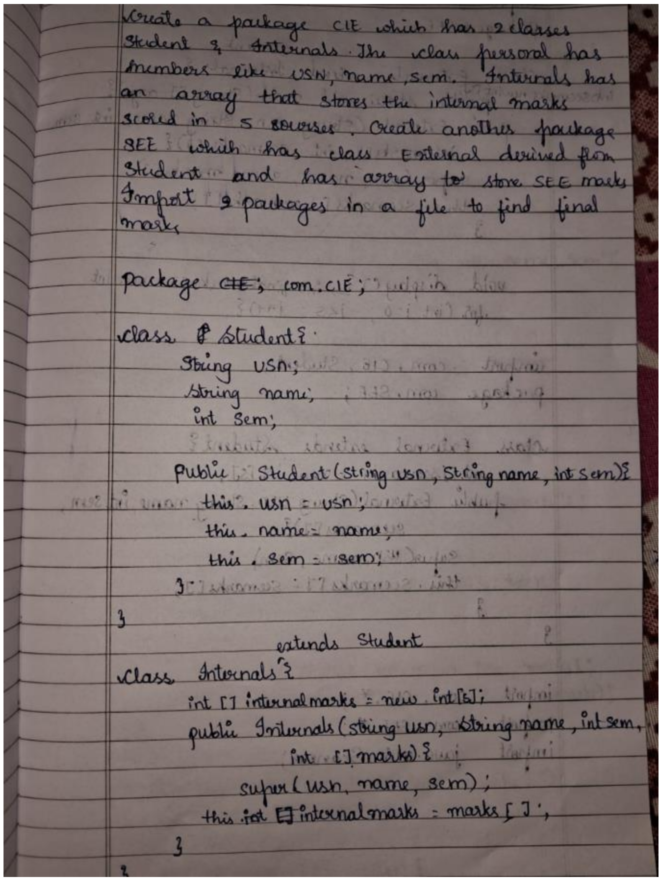


```
Menu:
1. Deposit
2. Display Balance
3. Compute Interest
4. Withdraw
5. Check Minimum Balance (Current Account only)
6. Exit
Enter your choice: 2
Current Balance: 0.0

Menu:
1. Deposit
2. Display Balance
3. Compute Interest
4. Withdraw
5. Check Minimum Balance (Current Account only)
6. Exit
Enter your choice: 6
Exiting. Thank you!
```

Program 6
Packages

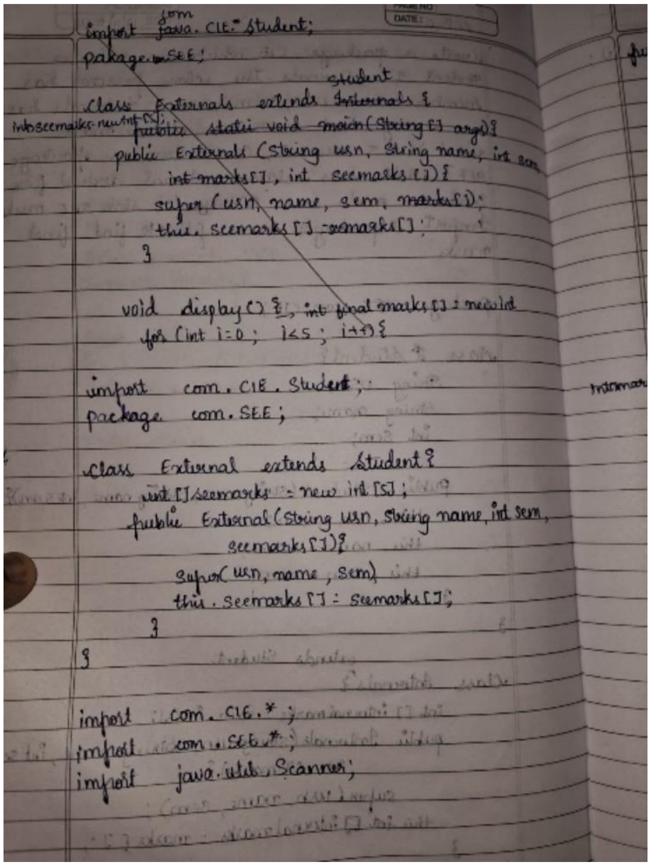
Code:

```
package CIE;

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

    public void displayDetails() {
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Semester: " + sem);
    }
}

public class Internals {
    public int[] internalMarks;
    public Internals(int[] marks) {
        if (marks.length != 5) {
            System.out.println("Error: Enter 5 marks!");
            return;
        }
        this.internalMarks = marks;
    }


    public void displayIMarks() {
        System.out.println("Internal Marks: ");
        for (int mark : internalMarks) {
            System.out.print(mark + " ");
        }
        System.out.println();
    }
}
package SEE;
import CIE.Student;
import CIE.Internals;

public class Externals extends Internals {
    public int[] externalMarks;
    public Externals(String name, String usn, int sem, int[] marks) {
        super(marks);
        if (marks.length != 5) {
            System.out.println("Error: Enter 5 marks!");
            return;
        }
        this.externalMarks = marks;
    }

    public void displayEMarks() {
        System.out.println("SEE Marks: ");
        for (int mark : externalMarks) {
            System.out.print(mark + " "); }
        System.out.println();
    }
}
import CIE.*;
import SEE.*;
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of students: ");
        int n = sc.nextInt();
        sc.nextLine();
        Externals[] students = new Externals[n];
        Internals[] intMarks = new Internals[n];
        for (int i = 0; i < n; i++) {
            System.out.println("Enter details for student " + (i + 1) + ":");
            System.out.print("Enter USN: ");
            String usn = sc.nextLine();
            System.out.print("Enter Name: ");
            String name = sc.nextLine();
            System.out.print("Enter Semester: ");
            int sem = sc.nextInt();
            System.out.println("Enter internal marks for 5 subjects:");
            int[] iMarks = new int[5];
            for (int j = 0; j < 5; j++) {
                iMarks[j] = sc.nextInt();
            }
            System.out.println("Enter external marks for 5 subjects:");
            int[] eMarks = new int[5];
            for (int j = 0; j < 5; j++) {
                eMarks[j] = sc.nextInt();
            }
            sc.nextLine();
            students[i] = new Externals(name, usn, sem, eMarks);
            intMarks[i] = new Internals(iMarks);
        }
        System.out.println("\nStudent Details:");
        for (int i = 0; i < n; i++) {
            students[i].displayDetails();
            intMarks[i].displayIMarks();
            students[i].displayEMarks();
        }
        sc.close();
    }
}
```

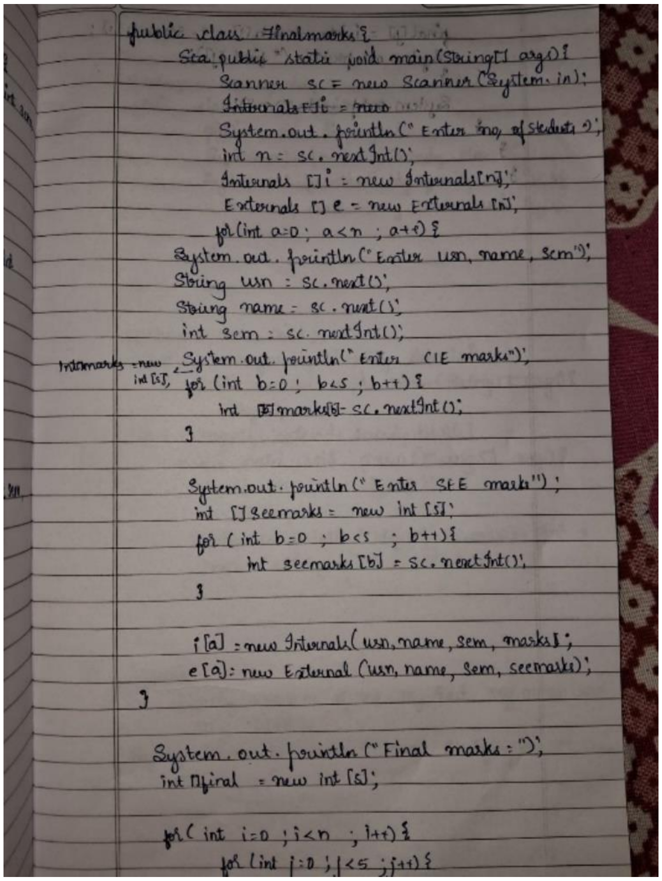


Create a package CIE which has 2 classes Student & Internals. The class personal has members like USN, name, sem. Internals has an array that stores the internal marks scored in 5 sources. Create another package SEE which has class External derived from Student and has array to store SEE marks Import 2 packages in a file to find final marks

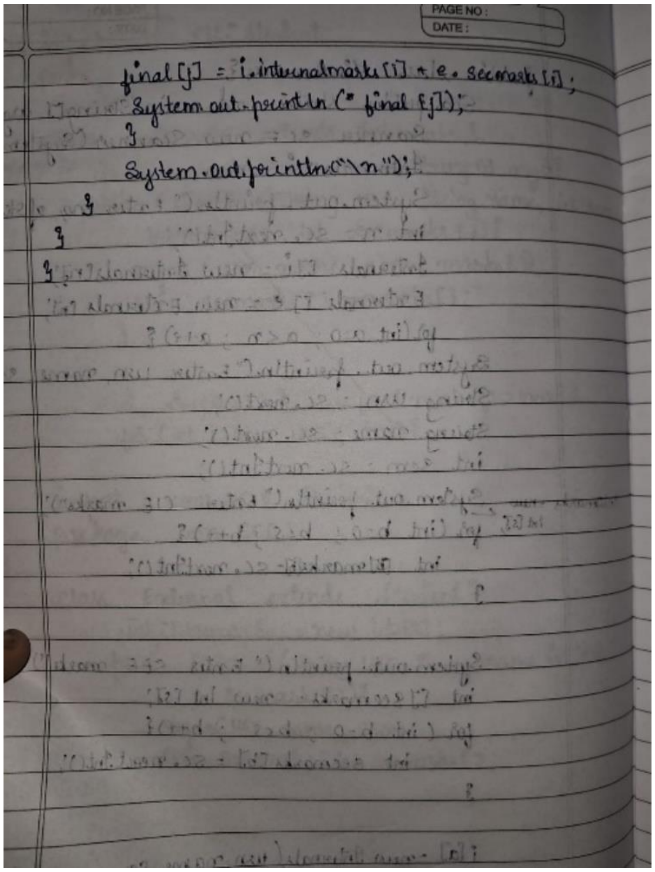
```
package ~~CIE~~; com.CIE;

class ~~P~~ Student{
      String usn;
      String name;
      int sem;

     Public Student(string usn, String name, int sem){
        this. usn = usn;
        this. name = name;
        this . sem = sem;
     }
}

class Internals extends Student{
      int [] internalmarks = new int[5];
      public Internals(string usn, String name, int sem,
                 int [] marks) {
           super(usn, name, sem);
        this ~~int []~~ internalmarks = marks [ ];
     }
}
```



```
~~import com java. CIE.* Student;~~
~~pakage SEE;~~

~~Class Externals extends Student Internals {~~
~~int seemarks = new int [5];~~
~~public static void main(String [] args){~~
~~public Externals (String usn, String name, int sem,~~
~~int marks[], int seemarks []){~~
~~super (usn, name, sem, marks[]);~~
~~this. seemarks [] = seemarks[];~~
~~}~~

~~void display() {, int final marks [] = new int~~
~~for (int i=0; i<5; i++){~~

import com. CIE. Student;
package com. SEE;

class External extends Student{
      int [] seemarks = new int [5];
    public External(String usn, String name, int sem,
              seemarks[]){
         super( usn, name, sem)
         this . seemarks [] = seemarks [];
    }
}

import com. CIE.*;
import com . SEE.*;
import java.util.Scanner;
```



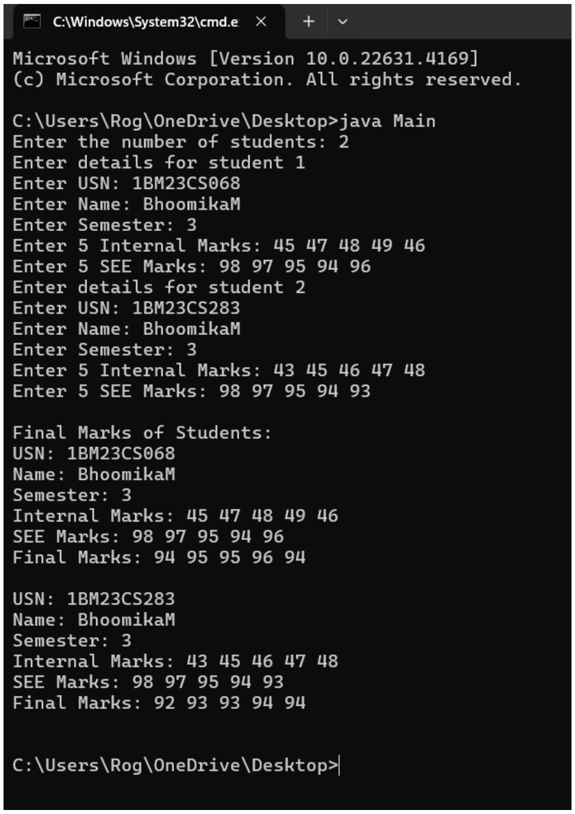
```
public class Finalmarks{
    ~~Sta~~ public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        ~~Internals [] i = new~~
        System.out.println(" Enter no. of students ");
        int n = sc.nextInt();
        Internals [] i = new Internals[n];
        Externals [] e = new Externals [n];
      for(int a=0; a<n ; a++) {
    System.out.println(" Enter usn, name, sem");
    String usn = sc.next();
    String name = sc.next();
    int sem = sc.nextInt();
    intmarks = new int [5];
    System.out.println(" Enter CIE marks");
    for (int b=0; b<5 ; b++) {
        int ~~[b]~~ marks[b] = sc.nextInt();
    }

    System.out.println(" Enter SEE marks");
    int [] seemarks = new int [5];
    for ( int b=0 ; b<5 ; b++){
          int seemarks [b] = sc.nextInt();
    }

    i [a] = new Internals( usn, name, sem, marks);
    e [a] = new External (usn, name, sem, seemarks);
  }

  System.out.println("Final marks:");
  int []final = new int [5];

  for( int i=0 ;i<n ; i++){
       for (int j=0 ; j<5 ; j++){
```



PAGE NO :
DATE :

```
        final [j] = i.internalmarks [i] + e. seemarks [i];
        System.out.println (" final [j]);
        }
        System.out.println("\n");
    }
  }
}
```

Output:

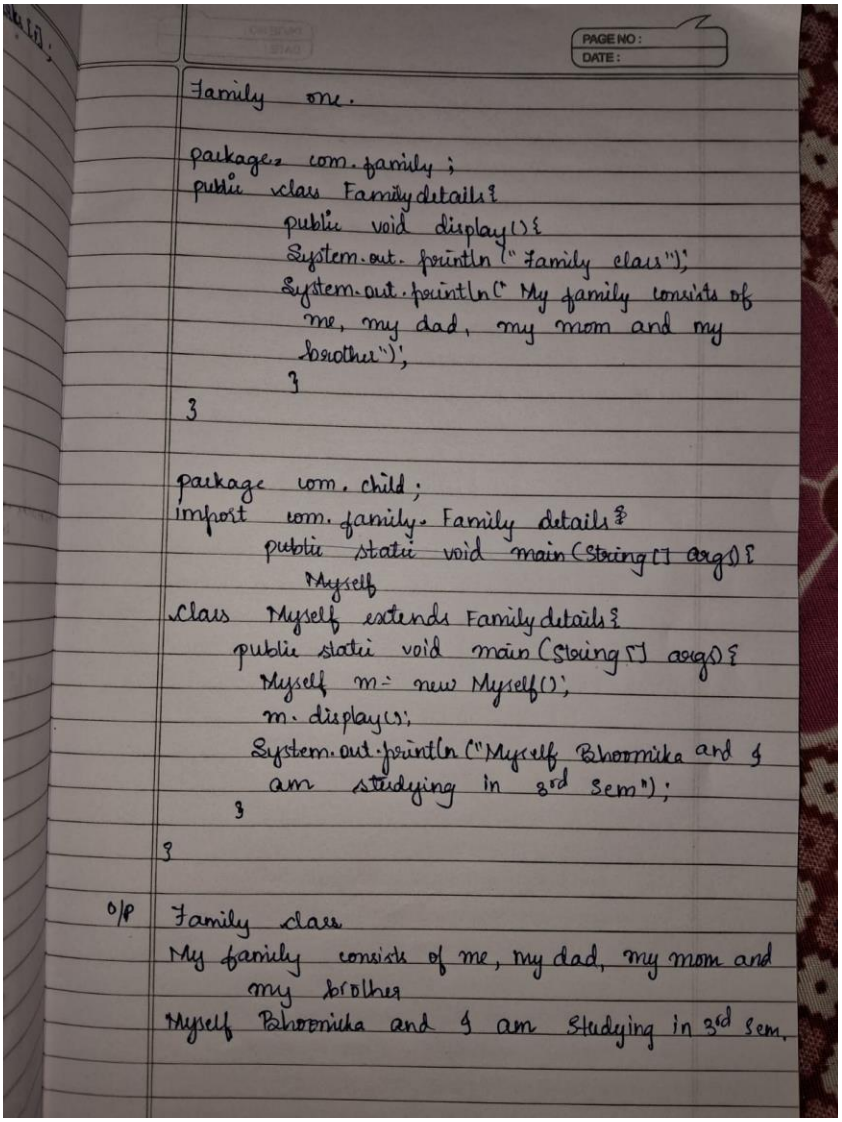


```
Microsoft Windows [Version 10.0.22631.4169]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Rog\OneDrive\Desktop>java Main
Enter the number of students: 2
Enter details for student 1
Enter USN: 1BM23CS068
Enter Name: BhoomikaM
Enter Semester: 3
Enter 5 Internal Marks: 45 47 48 49 46
Enter 5 SEE Marks: 98 97 95 94 96
Enter details for student 2
Enter USN: 1BM23CS283
Enter Name: BhoomikaM
Enter Semester: 3
Enter 5 Internal Marks: 43 45 46 47 48
Enter 5 SEE Marks: 98 97 95 94 93

Final Marks of Students:
USN: 1BM23CS068
Name: BhoomikaM
Semester: 3
Internal Marks: 45 47 48 49 46
SEE Marks: 98 97 95 94 96
Final Marks: 94 95 95 96 94

USN: 1BM23CS283
Name: BhoomikaM
Semester: 3
Internal Marks: 43 45 46 47 48
SEE Marks: 98 97 95 94 93
Final Marks: 92 93 93 94 94


C:\Users\Rog\OneDrive\Desktop>
```

2.Family program:

Code:

```
package com.example.me;
public class Myself{
  public void bhoomika(){
    System.out.println(“I am Bhoomika BG\n My age is 19”);}
}

package com.example.family;
import com.example.me.Myself;
public class Family{
  public static void main(String[] args){
    Myself me=new Myself();
    Family fam=new Family();
    System.out.println(“MY Family has 3 members”;
    me.bhoomika();
  }
}
```



PAGE NO :
DATE :

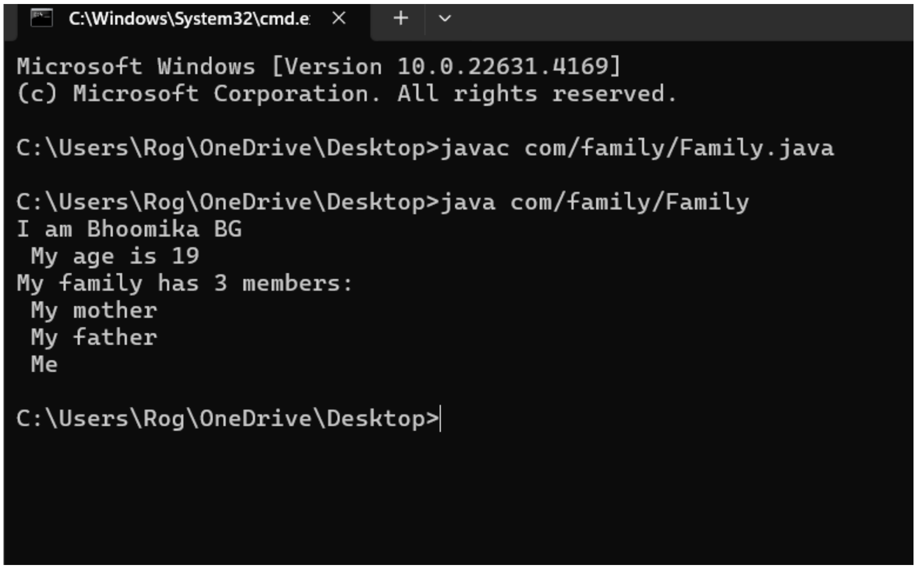
Family one.

```
package, com.family ;
public class Familydetails{
        public void display(){
        System.out. println ("Family class");
        System.out.println(" My family consists of
          me, my dad, my mom and my
          brother");
        }
}

package com. child ;
import com. family. Family details;
        public static void main(String[] args){
           Myself
class  Myself extends Familydetails{
      public static void main (String[] args){
        Myself m= new Myself();
        m. display();
        System.out.println ("Myself Bhoomika and I
         am studying in 3rd sem");
      }
}
```

o/p

```
Family class
My family consists of me, my dad, my mom and
      my brother
Myself Bhoomika and I am Studying in 3rd Sem.
```

Output:

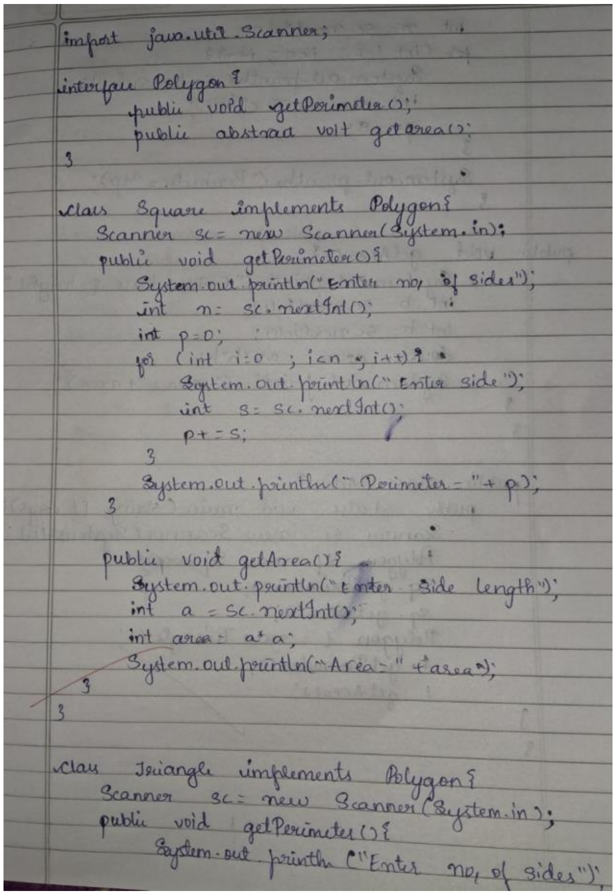


```
Microsoft Windows [Version 10.0.22631.4169]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Rog\OneDrive\Desktop>javac com/family/Family.java

C:\Users\Rog\OneDrive\Desktop>java com/family/Family
I am Bhoomika BG
 My age is 19
My family has 3 members:
 My mother
 My father
 Me

C:\Users\Rog\OneDrive\Desktop>
```

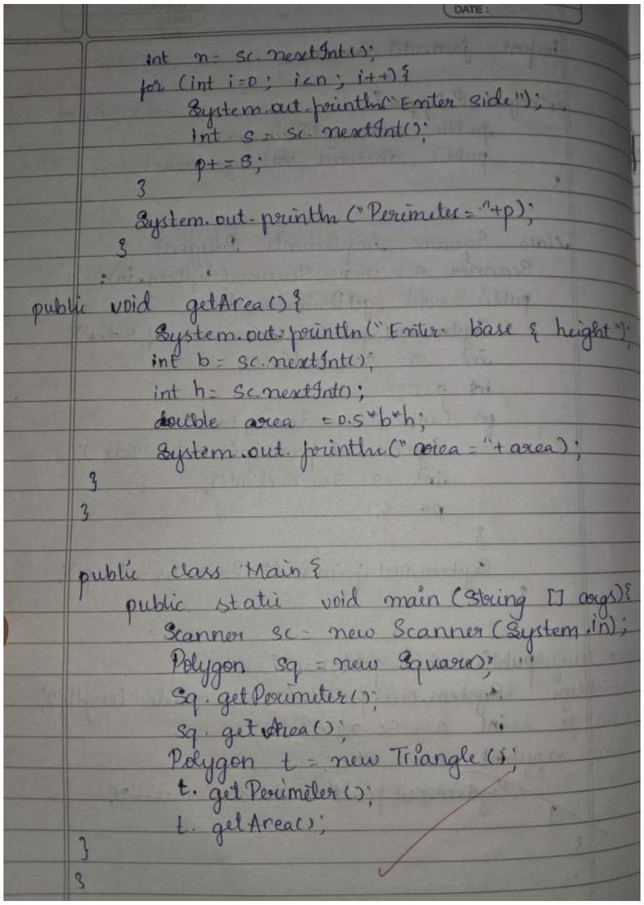
Program 7
Interfaces

Code:

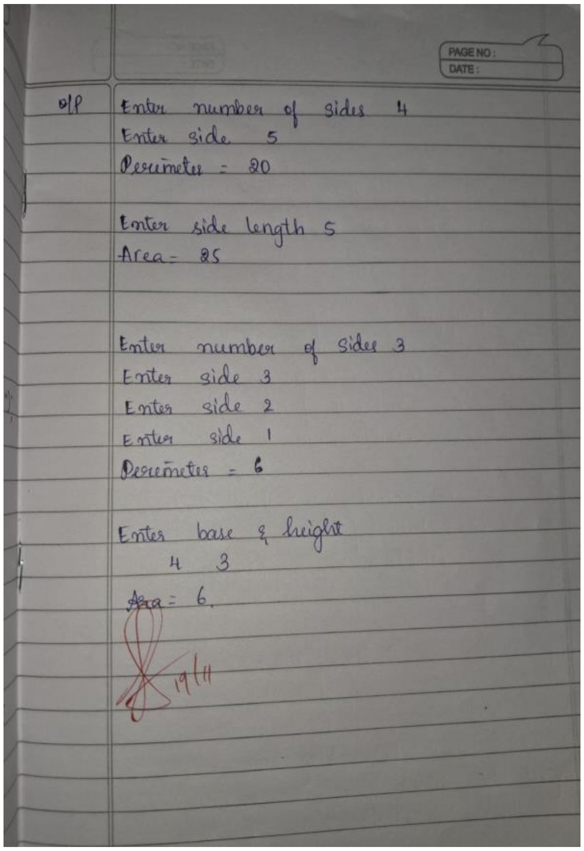
```
interface Polygon {
    default double getPerimeter(sides) {
        double perimeter = 0;
        for (double side : sides) {
            perimeter += side;
        }
        return perimeter;
    }
    double getArea();
}

class Rectangle implements Polygon {
    private double length;
    private double breadth;

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

    public double getArea() {
        return length * breadth;
    }
    public double calculatePerimeter() {
        return getPerimeter(length, breadth, length, breadth);
    }
}

class Triangle implements Polygon {
    private double base;
    private double height;
    private double side1, side2, side3;

    Triangle(double base, double height, double side1, double side2, double side3) {
        this.base = base;
        this.height = height;
        this.side1 = side1;
        this.side2 = side2;
        this.side3 = side3;
    }

    public double getArea() {
        return 0.5 * base * height;
    }

    public double calculatePerimeter() {
        return getPerimeter(side1, side2, side3);
    }
}

// Main Class
public class Main {
    public static void main(String[] args) {
        // Rectangle instance
        Rectangle rectangle = new Rectangle(5.0, 3.0);
        System.out.println("Rectangle:");
        System.out.println("Area: " + rectangle.getArea());
        System.out.println("Perimeter: " + rectangle.calculatePerimeter());

        // Triangle instance
        Triangle triangle = new Triangle(5.0, 4.0, 3.0, 4.0, 5.0);
        System.out.println("\nTriangle:");
        System.out.println("Area: " + triangle.getArea());
        System.out.println("Perimeter: " + triangle.calculatePerimeter());
    }
}
```



```
import java.util.Scanner;

interface Polygon{
        public void getPerimeter();
        public abstract voit getarea();
}

class Square implements Polygon{
    Scanner sc= new Scanner(System.in);
    public void getPerimeter(){
        System.out.println("Enter no, of sides");
        int n= sc.nextInt();
        int p=0;
        for (int i=0 ; i<n ; i++){
            System.out.println(" Enter side");
            int s= sc.nextInt();
            p+=s;
        }
        System.out.println(" Perimeter = "+ p);
    }

    public void getArea(){
        System.out.println(" Enter Side length");
        int a = sc.nextInt();
        int area= a* a;
        System.out.println(" Area = " +area);
    }
}

class Triangle implements Polygon{
    Scanner sc= new Scanner(System.in);
    public void getPerimeter(){
        System.out.println ("Enter no, of sides");
```



```
        int n= sc.nextInt();
        for (int i=0; i<n; i++){
            System.out.println(" Enter side");
            Int s = sc.nextInt();
            p+=s;
        }
        System.out.println ("Perimeter = "+p);
    }

public void getArea(){
        System.out.println(" Enter base & height");
        int b= sc.nextInt();
        int h= sc.nextInt();
        double area =0.5*b*h;
        System.out.println(" area = "+ area);
    }
}

public class Main{
    public static void main (String [] args){
        Scanner sc= new Scanner (System.in);
        Polygon sq = new Square();
        sq.getPerimeter();
        sq.getArea();
        Polygon t = new Triangle();
        t. getPerimeter();
        t. getArea();
}
}
```



PAGE NO :
DATE :

o/p Enter number of sides 4
Enter side 5
Perimeter = 20

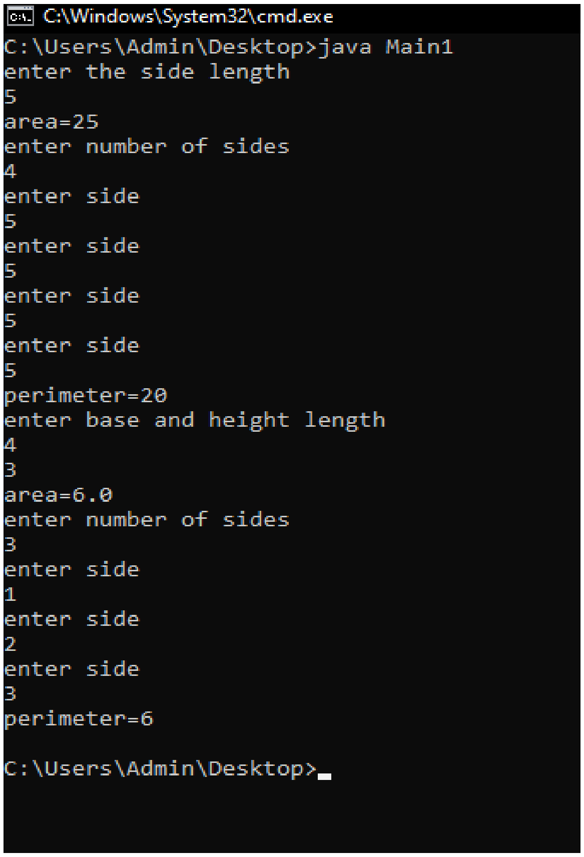
Enter side length 5
Area = 25

Enter number of sides 3
Enter side 3
Enter side 2
Enter side 1
Perimeter = 6

Enter base & height
4 3
Area = 6.

19/11

Output:



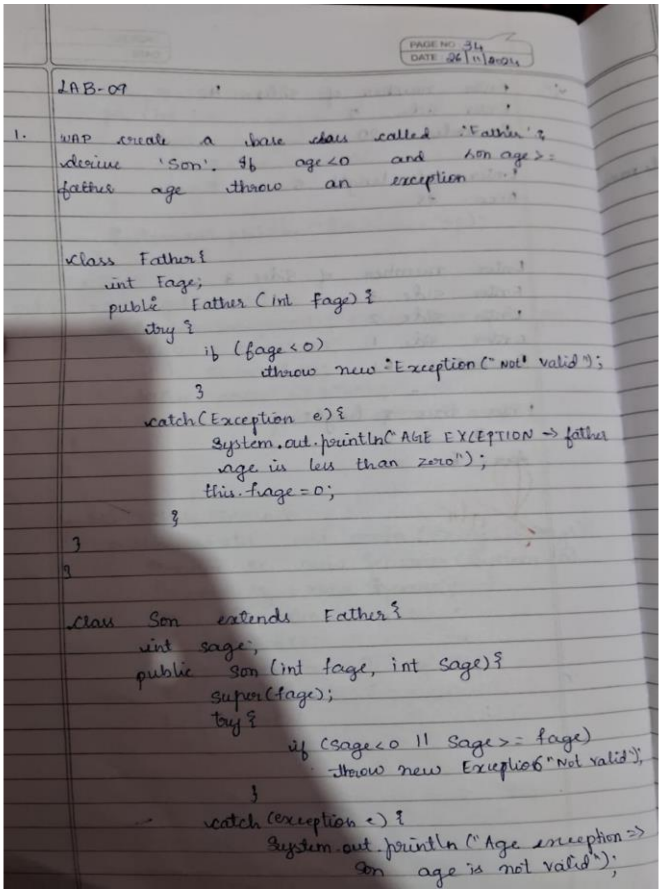
C:\Windows\System32\cmd.exe

```
C:\Users\Admin\Desktop>java Main1
enter the side length
5
area=25
enter number of sides
4
enter side
5
enter side
5
enter side
5
enter side
5
perimeter=20
enter base and height length
4
3
area=6.0
enter number of sides
3
enter side
1
enter side
2
enter side
3
perimeter=6

C:\Users\Admin\Desktop>
```

Program 8
Exception Handling

Code:

```
class WrongAgeException extends Exception {
    public WrongAgeException(String message) {
        super(message);
    }
}
class Father {
    protected int age;
    public Father(int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException("Father's age cannot be less than zero.");
        }
        this.age = age;
    }
}

class Son extends Father {
    private int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAgeException {
        super(fatherAge);
        if (sonAge < 0) {
            throw new WrongAgeException("Son's age cannot be less than zero.");
        }
        if (sonAge >= fatherAge) {
            throw new WrongAgeException("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: " + age);
        System.out.println("Son's Age: " + sonAge);
    }
}

public class ExceptionHandlingInheritance {
    public static void main(String[] args) {
        try {
            Son son1 = new Son(40, 15);
            son1.displayAges();
            Father fatherInvalid = new Father(-5);
            Son sonInvalid = new Son(35, 40);
        } catch (WrongAgeException e) {
            System.out.println("Exception occurred: " + e.getMessage());
        }
    }
}
```

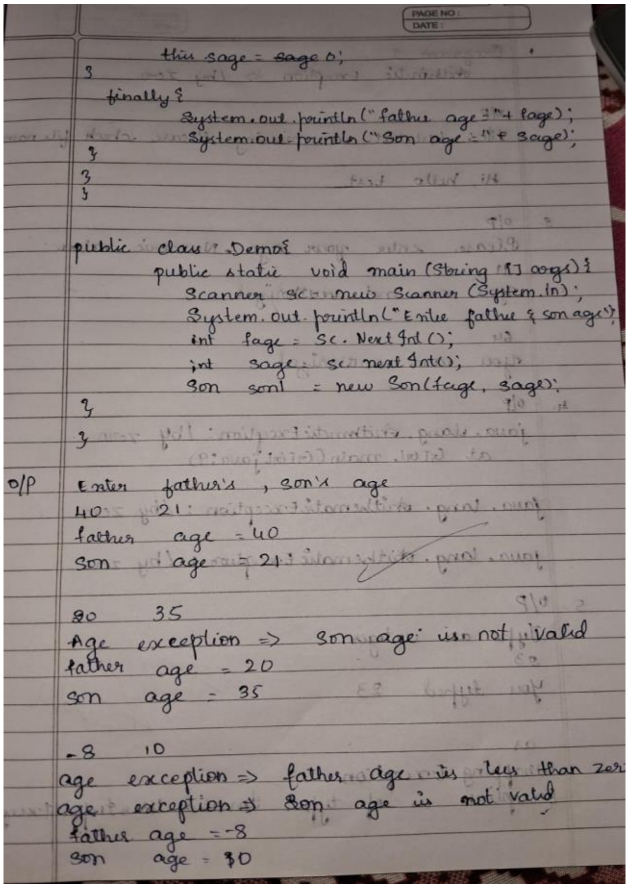


PAGE NO 34
DATE 26|11|2024

LAB-07

1. WAP create a base class called 'Father' & derive 'Son'. If age<0 and son age >= father age throw an exception

```
class Father{
    int Fage;
    public Father (int Fage) {
        try {
            if (fage<0)
                throw new Exception ("Not valid");
        }
        catch(Exception e){
            System.out.println("AGE EXCEPTION -> father
            age is less than zero");
            this.fage=0;
        }
    }
}

class Son extends Father{
    int sage;
    public Son (int fage, int sage){
        super(fage);
        try{
            if (sage<0 || sage>= fage)
                throw new Exception("Not valid");
        }
        catch (exception e) {
            System.out.println ("Age exception =>
                son age is not valid");
```

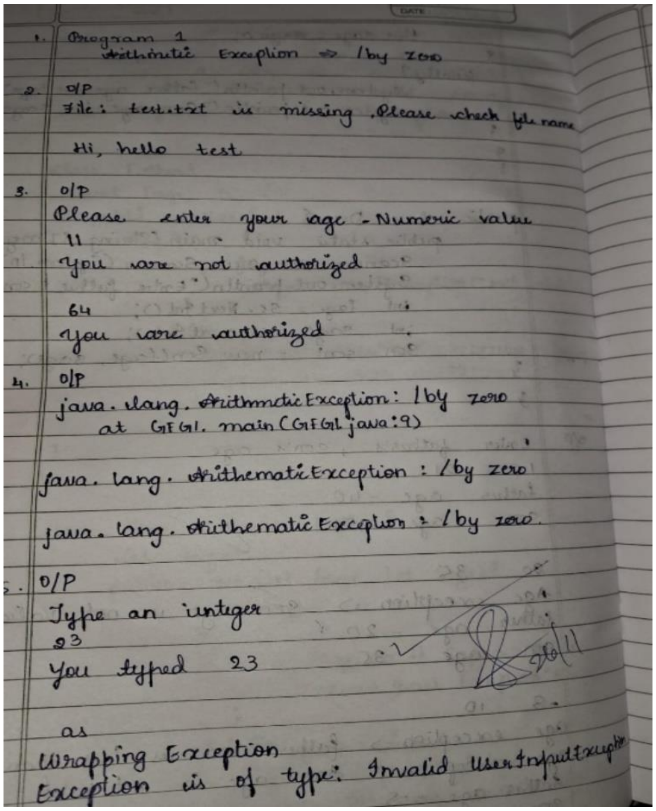


PAGE NO:
DATE:

```
            this sage = ~~sage~~ 0;
        }
        finally{
            System.out.println("father age ="+ fage);
            System.out.println("Son age ="+ sage);
        }
    }
}

public class Demo{
    public static void main (String [] args){
        Scanner sc = new Scanner (System.in);
        System.out.println("Enter father & son age");
        int fage = sc. NextInt ();
        int sage = sc.nextInt();
        Son son1 = new Son(fage, sage);
    }
}
```

O/P

```
Enter father's , son's age
40 21
father age =40
son age = 21

20 35
Age exception => son age is not valid
father age = 20
son age = 35

-8 10
age exception => father age is less than zer
age exception => son age is not valid
father age =-8
son age = 10
```



DATE

1. Program 1
Arithmetic Exception => /by zero

2. O/P

```
File: test.txt is missing .Please check file name

Hi, hello test
```

3. O/P

```
Please enter your age - Numeric value
11
You are not authorized

64
You are authorized
```

4. O/P

```
java.lang.ArithmeticException: /by zero
    at GFG1.main(GFG1.java:9)

java.lang.ArithematicException : /by zero

java.lang.ArithematicException : /by zero.
```

5. O/P

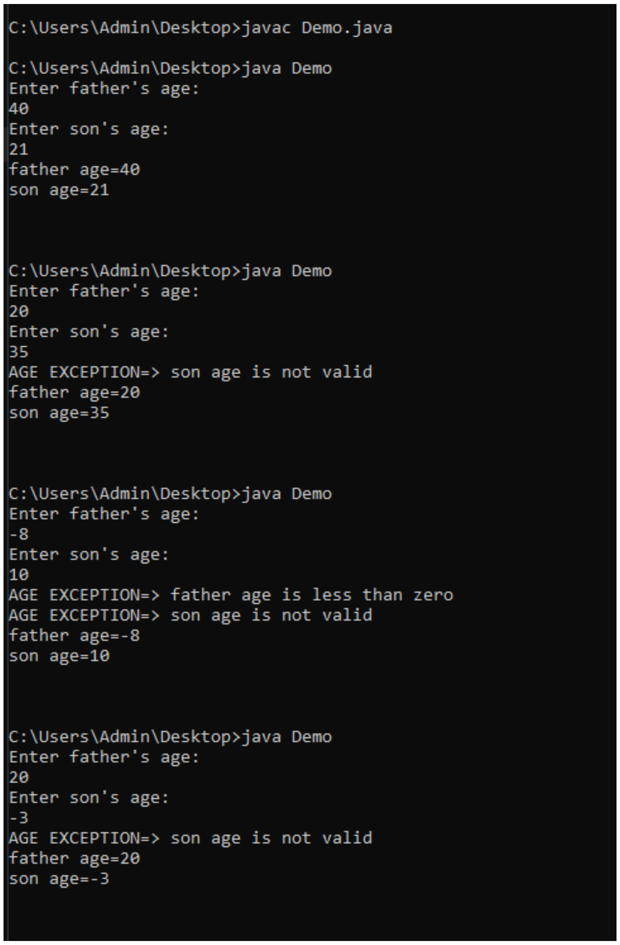
```
Type an integer
23
You typed 23
```

26/11

as

```
Wrapping Exception
Exception is of type: Invalid UserInputException
```

Output:

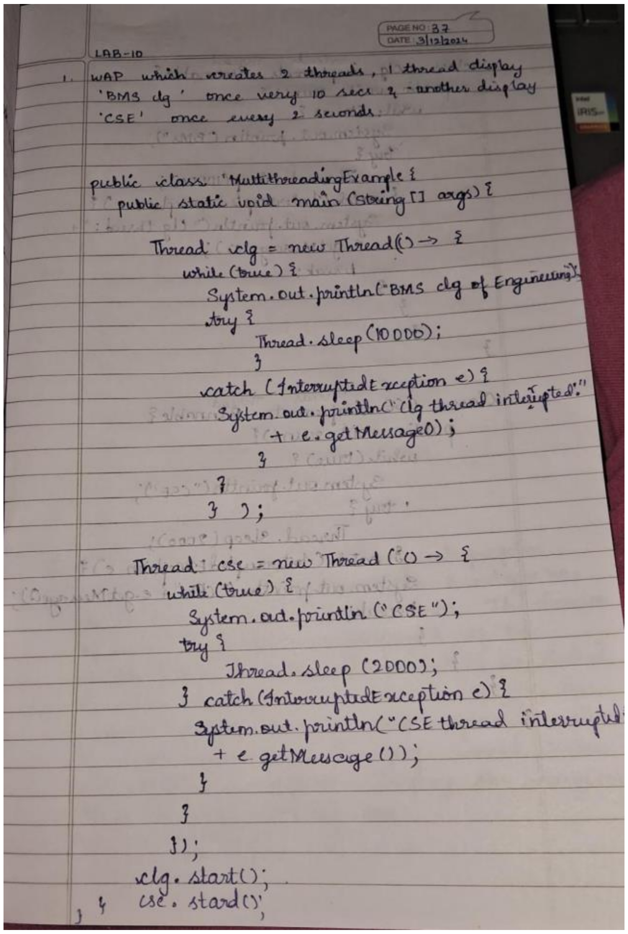


```
C:\Users\Admin\Desktop>javac Demo.java

C:\Users\Admin\Desktop>java Demo
Enter father's age:
40
Enter son's age:
21
father age=40
son age=21



C:\Users\Admin\Desktop>java Demo
Enter father's age:
20
Enter son's age:
35
AGE EXCEPTION=> son age is not valid
father age=20
son age=35



C:\Users\Admin\Desktop>java Demo
Enter father's age:
-8
Enter son's age:
10
AGE EXCEPTION=> father age is less than zero
AGE EXCEPTION=> son age is not valid
father age=-8
son age=10



C:\Users\Admin\Desktop>java Demo
Enter father's age:
20
Enter son's age:
-3
AGE EXCEPTION=> son age is not valid
father age=20
son age=-3
```

Program 9
Thread Programming

Code:

```
class Thread1 extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000);
            }
        } catch (InterruptedException e) {
            System.out.println("Thread1 interrupted: " + e.getMessage());
        }
    }
}

class Thread2 extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("CSE");
                Thread.sleep(2000);
            }
        } catch (InterruptedException e) {
            System.out.println("Thread2 interrupted: " + e.getMessage());
        }
    }
}

public class MultiThreadDisplay {
    public static void main(String[] args) {
        Thread1 t1 = new Thread1();
        t1.start();
        Thread2 t2 = new Thread2();
        t2.start();
    }
}
```

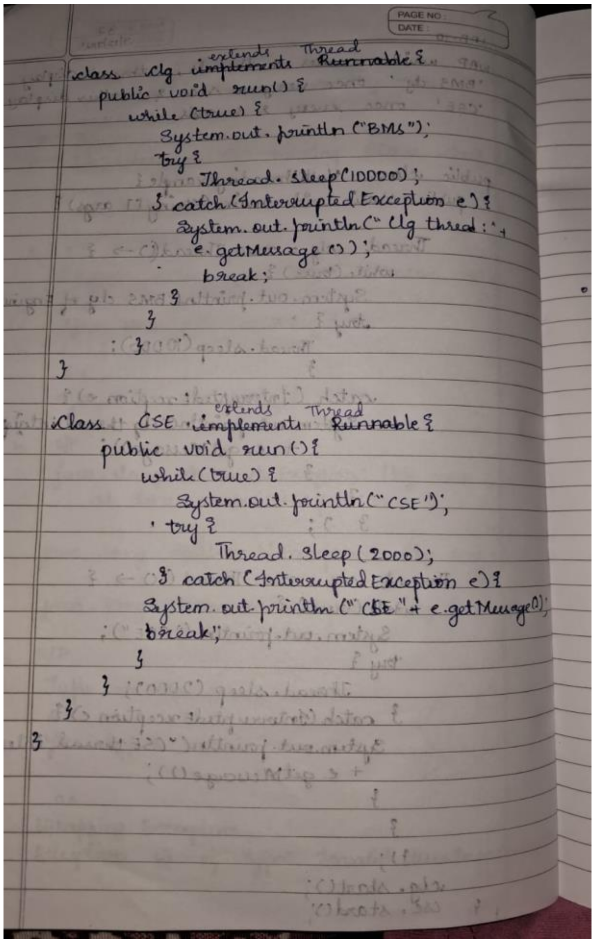


PAGE NO: 37
DATE: 3/12/2024

LAB-10

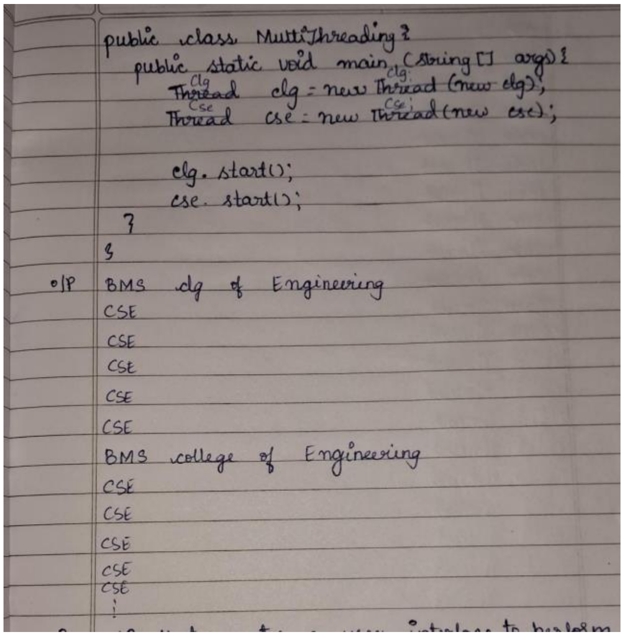
1. WAP which creates 2 threads, 1 thread display 'BMS clg' once very 10 secs & another display 'CSE' once every 2 seconds.

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

      Thread clg = new Thread(() -> {
        while (true) {
          System.out.println("BMS clg of Engineering");
          try {
              Thread.sleep (10000);
              }
          catch (InterruptedException e) {
            System.out.println("clg thread interupted:"
                + e.getMessage());
              }
          }
          } );

    Thread cse = new Thread (() -> {
      while (true) {
        System.out.println ("CSE");
       try {
            Thread.sleep (2000);
       } catch (InterruptedException e) {
         System.out.println("CSE thread interrupted
           + e getMessage());
         }
       }
      });
    clg.start();
  } cse.start();
}
```



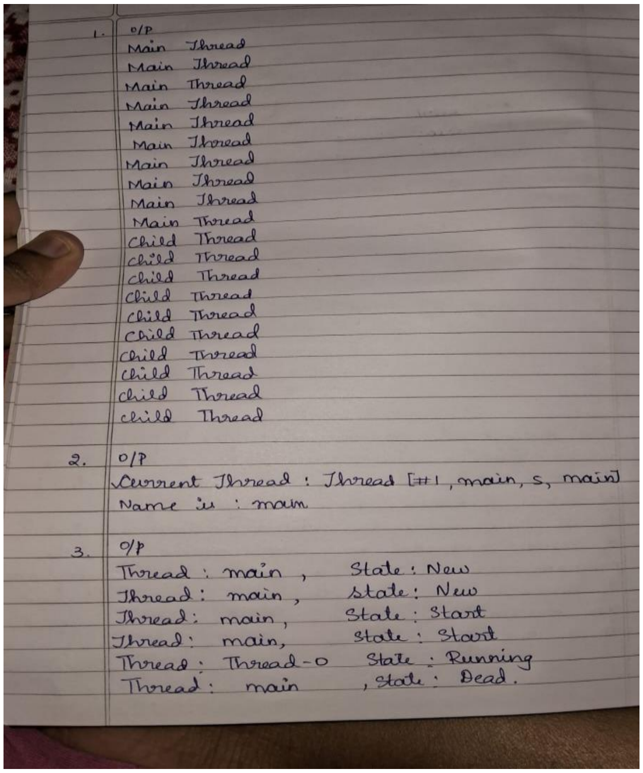
PAGE NO:
DATE:

```
                  extends    Thread
class clg implements Runnable {
   public void run() {
      while (true) {
         System.out.println ("BMS");
         try {
            Thread.sleep(10000);
         } catch (InterruptedException e) {
           System.out.println(" Clg thread: " +
           e.getMessage ());
            break;
         }
       }
    }
}

                  extends   Thread
Class CSE implements Runnable {
    public void run() {
       while (true) {
          System.out.println("CSE");
        try {
             Thread. sleep (2000);
         } catch (InterruptedException e) {
        System.out.println (" CSE " + e.getMessage());
        break;
       }
     }
   }
}
```

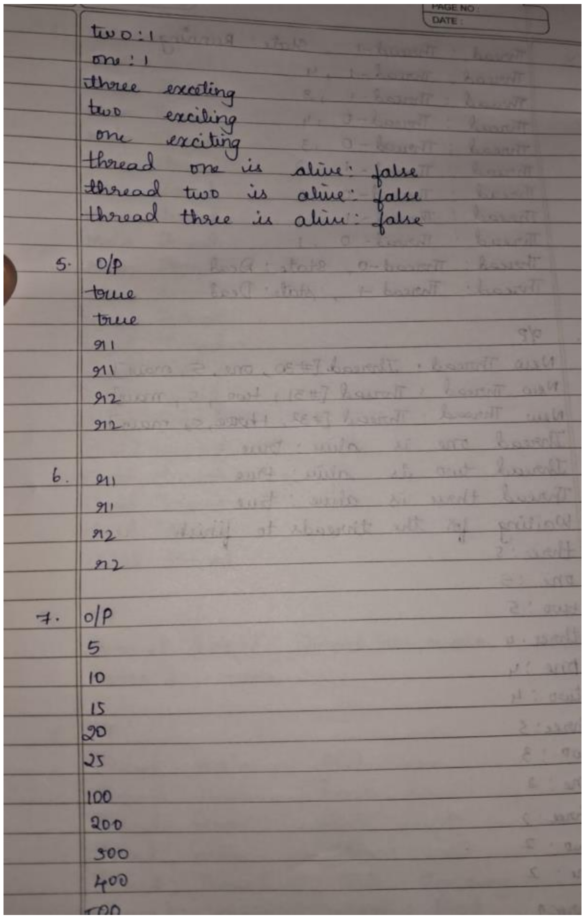


```
public class MultiThreading {
    public static void main (String [] args) {
        Clg                   clg
        Thread  clg = new Thread (new clg);
        Cse                   Cse
        Thread  cse = new Thread(new cse);

        clg. start();
        cse. start();
    }
}
```

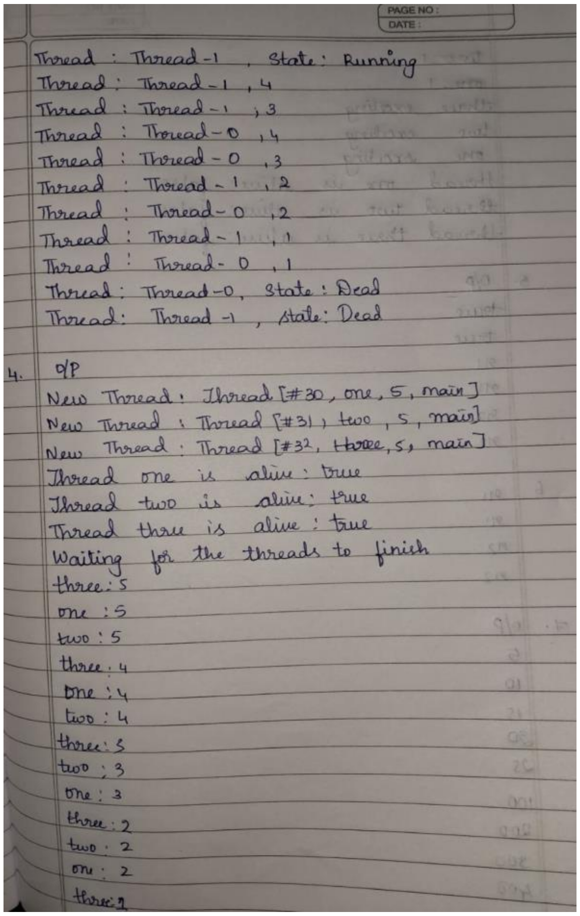
```
o/P  BMS clg of Engineering
     CSE
     CSE
     CSE
     CSE
     CSE
     BMS college of Engineering
     CSE
     CSE
     CSE
     CSE
     CSE
     :
```



```
1.  o/P
    Main Thread
    Main Thread
    Main Thread
    Main Thread
    Main Thread
    Main Thread
    Main Thread
    Main Thread
    Main Thread
    Main Thread
    Child Thread
    Child Thread
    Child Thread
    Child Thread
    Child Thread
    Child Thread
    Child Thread
    Child Thread
    Child Thread
    Child Thread

2.  o/P
    Current Thread : Thread [#1, main, 5, main]
    Name is : main

3.  o/P
    Thread : main ,       State : New
    Thread : main ,       State : New
    Thread : main,        State : Start
    Thread : main,        State : Start
    Thread : Thread-0     State : Running
    Thread : main       , State : Dead.
```

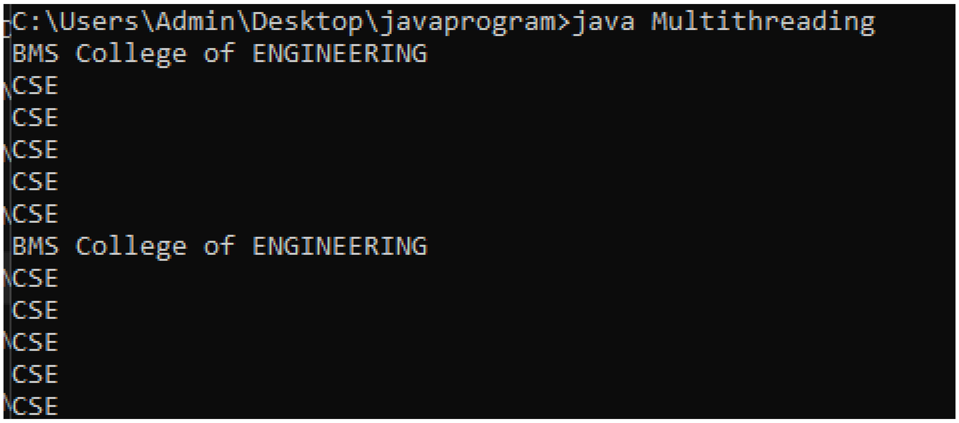


```
    two : 1
    one : 1
    three exciting
    two exciting
    one exciting
    thread one is alive: false
    thread two is alive: false
    thread three is alive: false

5.  O/P
    true
    true
    91
    91
    92
    92

6.  91
    91
    92
    92

7.  o/P
    5
    10
    15
    20
    25
    100
    200
    300
    400
    500
```



```
    Thread : Thread-1 , State: Running
    Thread : Thread-1 , 4
    Thread : Thread-1 , 3
    Thread : Thread-0 , 4
    Thread : Thread-0 , 3
    Thread : Thread-1 , 2
    Thread : Thread-0 , 2
    Thread : Thread-1 , 1
    Thread : Thread-0 , 1
    Thread : Thread-0, State : Dead
    Thread : Thread-1 , State: Dead

4.  O/P
    New Thread : Thread [#30, one, 5, main]
    New Thread : Thread [#31, two , 5, main]
    New Thread : Thread [#32, three, 5, main]
    Thread one is alive : true
    Thread two is alive : true
    Thread three is alive : true
    Waiting for the threads to finish
    three : 5
    one : 5
    two : 5
    three : 4
    one : 4
    two : 4
    three : 3
    two : 3
    one : 3
    three : 2
    two : 2
    one : 2
    three : 1
```

Output:



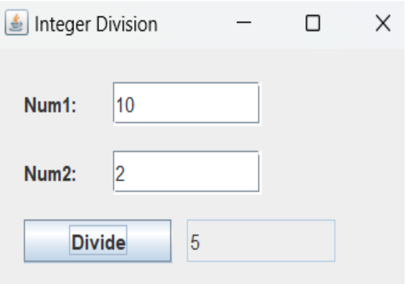
```
C:\Users\Admin\Desktop\javaprogram>java Multithreading
BMS College of ENGINEERING
CSE
CSE
CSE
CSE
CSE
BMS College of ENGINEERING
CSE
CSE
CSE
CSE
CSE
```

Program 10
Open Ended Exercise

Code:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class DivisionApp {
    public static void main(String[] args)
        JFrame frame = new JFrame("Division Calculator");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400, 200);
        frame.setLayout(new GridLayout(4, 2, 10, 10));
        JLabel labelNum1 = new JLabel("Enter number 1:");
        JTextField textNum1 = new JTextField();
        JLabel labelNum2 = new JLabel("Enter number 2:");
        JTextField textNum2 = new JTextField();
        JLabel labelResult = new JLabel("Result:");
        JTextField textResult = new JTextField();
        textResult.setEditable(false);

        JButton buttonDivide = new JButton("Divide");
        frame.add(labelNum1);
        frame.add(textNum1);
        frame.add(labelNum2);
        frame.add(textNum2);
        frame.add(labelResult);
        frame.add(textResult);
        frame.add(new JLabel());
        frame.add(buttonDivide);
        buttonDivide.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                try {
                    int num1 = Integer.parseInt(textNum1.getText());
                    int num2 = Integer.parseInt(textNum2.getText());
                    if (num2 == 0) {
                        throw new ArithmeticException("Cannot divide by zero.");
                    }
                    int result = num1 / num2;
                    textResult.setText(String.valueOf(result));
                } catch (NumberFormatException ex) {
                    JOptionPane.showMessageDialog(frame,
                        "Invalid input! Please enter integers only.",
                        "Number Format Error",
                        JOptionPane.ERROR_MESSAGE);
                } catch (ArithmeticException ex) {
                    JOptionPane.showMessageDialog(frame,
                        ex.getMessage(),
                        "Arithmetic Error",
                        JOptionPane.ERROR_MESSAGE);
                }
            }
        });
        frame.setVisible(true);
    }
}
```

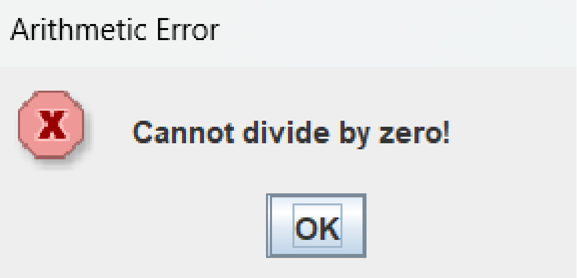
Output:



Num1: 10

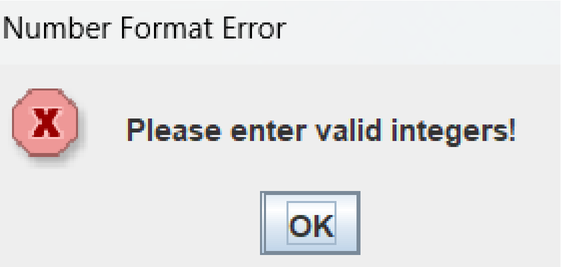
Num2: 2

5



Arithmetic Error

Cannot divide by zero!



Number Format Error

Please enter valid integers!