VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Daivya Priyankkumar Shah (1BM23CS084)

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 **Daivya Priyankkumar Shah (1BM23CS084),** 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.

Geetha N Assistant Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE

Index

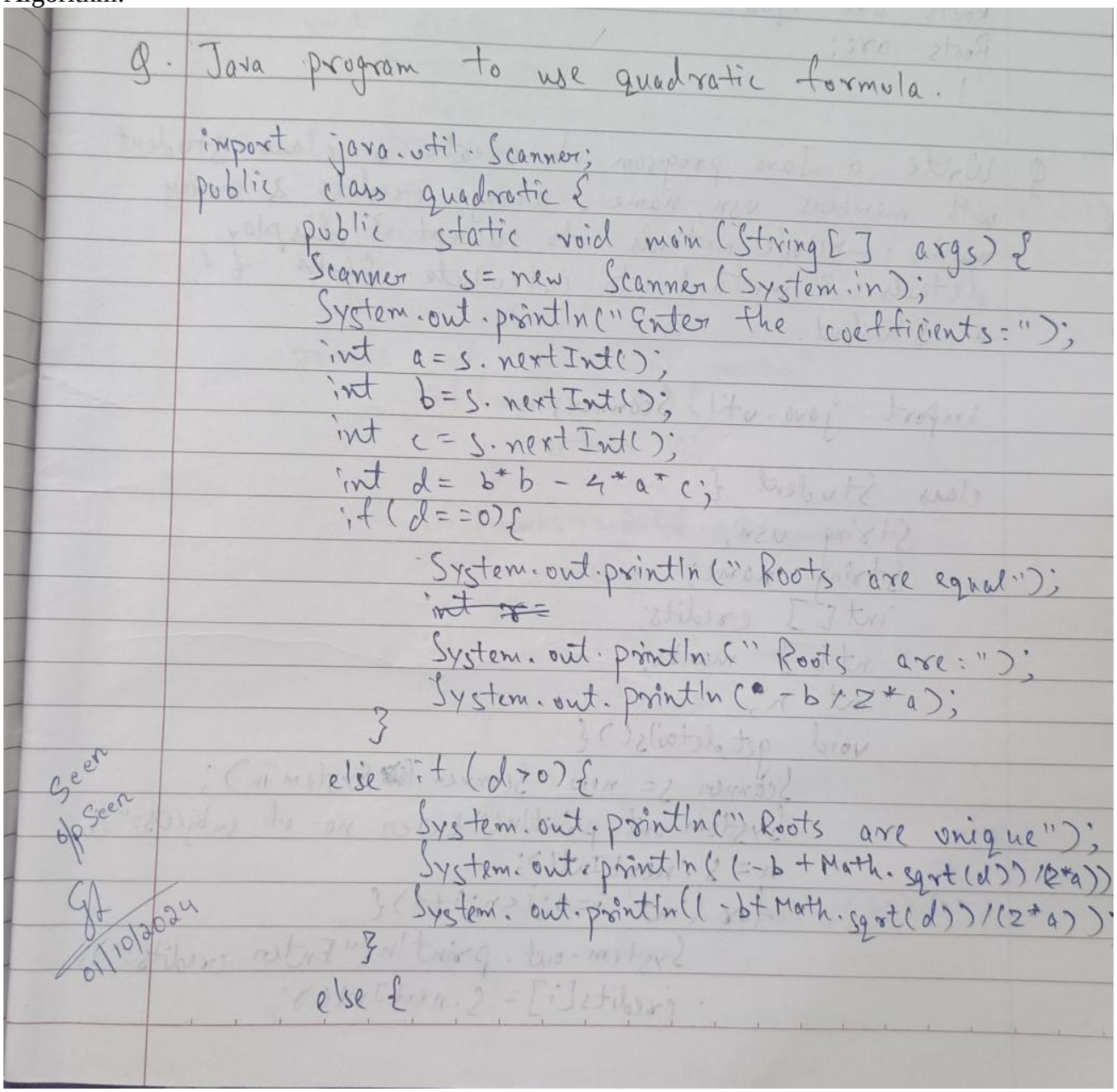
Github Link:

https://github.com/daivya17/OOJ_Lab

Program 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 discriminate b2 -4ac is negative, display a message stating that there are no real solutions

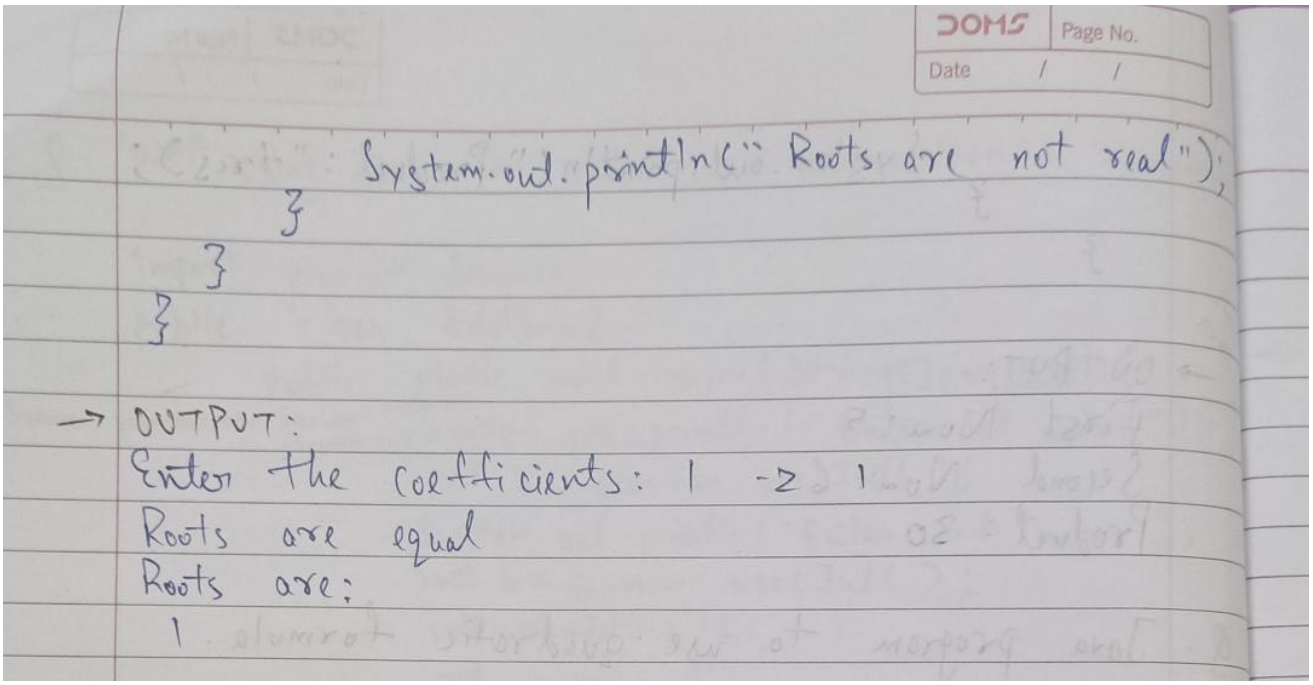
Algorithm:



Q. Java program to use quadratic formula.

```
import java.util.Scanner;
public class quadratic {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the coefficients:");
        int a = s.nextInt();
        int b = s.nextInt();
        int c = s.nextInt();
        int d = b*b - 4*a*c;
        if(d==0){
            System.out.println("Roots are equal");
            int x=
            System.out.println("Roots are:");
            System.out.println(-b/2*a);
        }
        else if(d>0){
            System.out.println("Roots are unique");
            System.out.println((-b+Math.sqrt(d))/(2*a));
            System.out.println((-b+Math.sqrt(d))/(2*a));
        }
        else{
```

seen
op seen
01/10/2024



Code:

```
import java.util.Scanner;
public class quad{
	public static void main(String[] args){
		Scanner s=new Scanner(System.in);
		System.out.println("Enter the coefficients:");
		int a=s.nextInt();
		int b=s.nextInt();
		int c=s.nextInt();
		int d=b*b-4*a*c;
		if(d==0){
			System.out.println("Roots are equal");
			System.out.println("Roots are:");
			System.out.println(-b/2*a);
}
		else if(d>0){
			System.out.println("Roots are unique");
			System.out.println((-b+Math.sqrt(d))/(2*a));
			System.out.println((-b-Math.sqrt(d))/(2*a));

}
		else{
			System.out.println("No real roots");
}
		System.out.println("Daivya Priyankkumar Shah");
		System.out.println("1BM23CS084");
	}
}
```

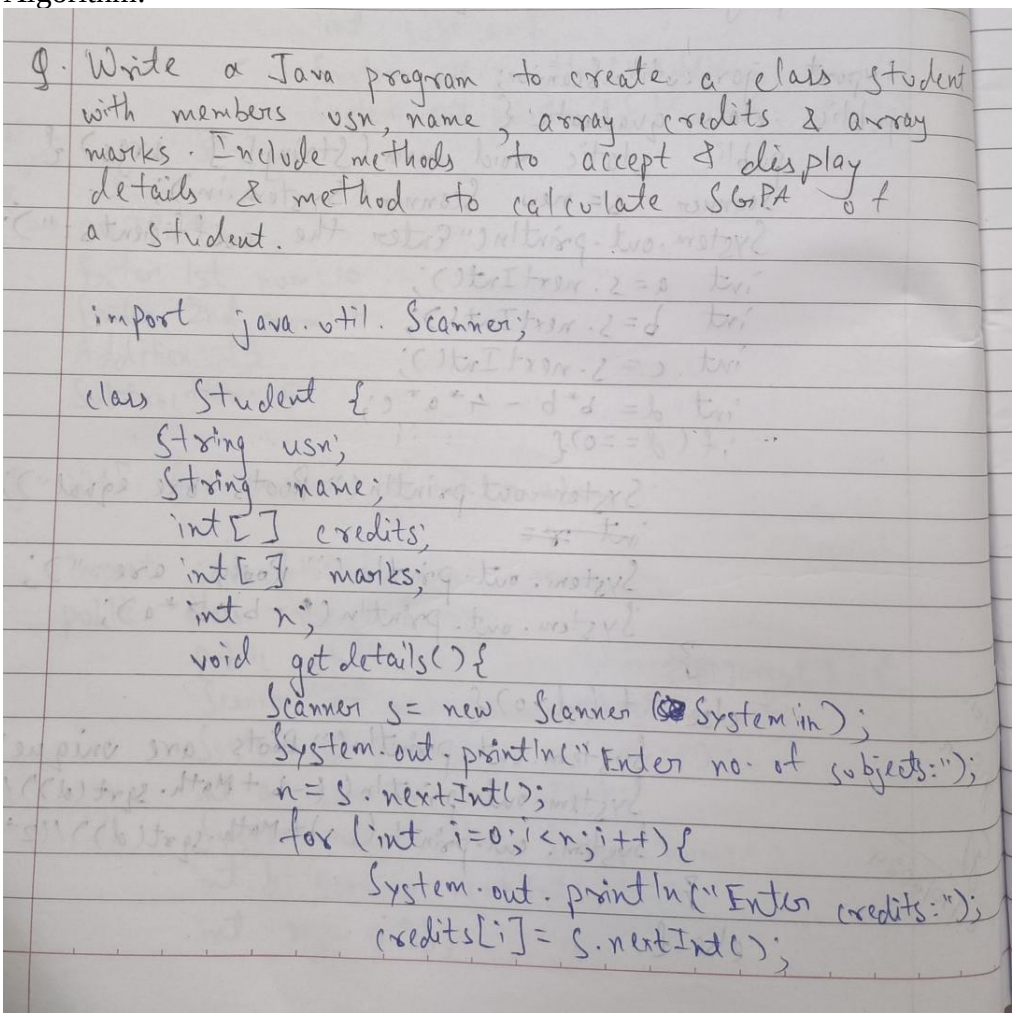
OUTPUT:

```
PS C:\Users\daivy\IdeaProjects\JavaPractice\src> java Quadratic.java
enter the coefficients:
1 -7 10
real and unequal roots
roots:5.0 and 2.0
PS C:\Users\daivy\IdeaProjects\JavaPractice\src> java Quadratic.java
enter the coefficients:
2 5 1
real and unequal roots
roots:-0.21922359359558485 and -2.2807764064044154
PS C:\Users\daivy\IdeaProjects\JavaPractice\src> java Quadratic.java
enter the coefficients:
1 -2 1
real and equal roots
roots:1.0
```

Program 2

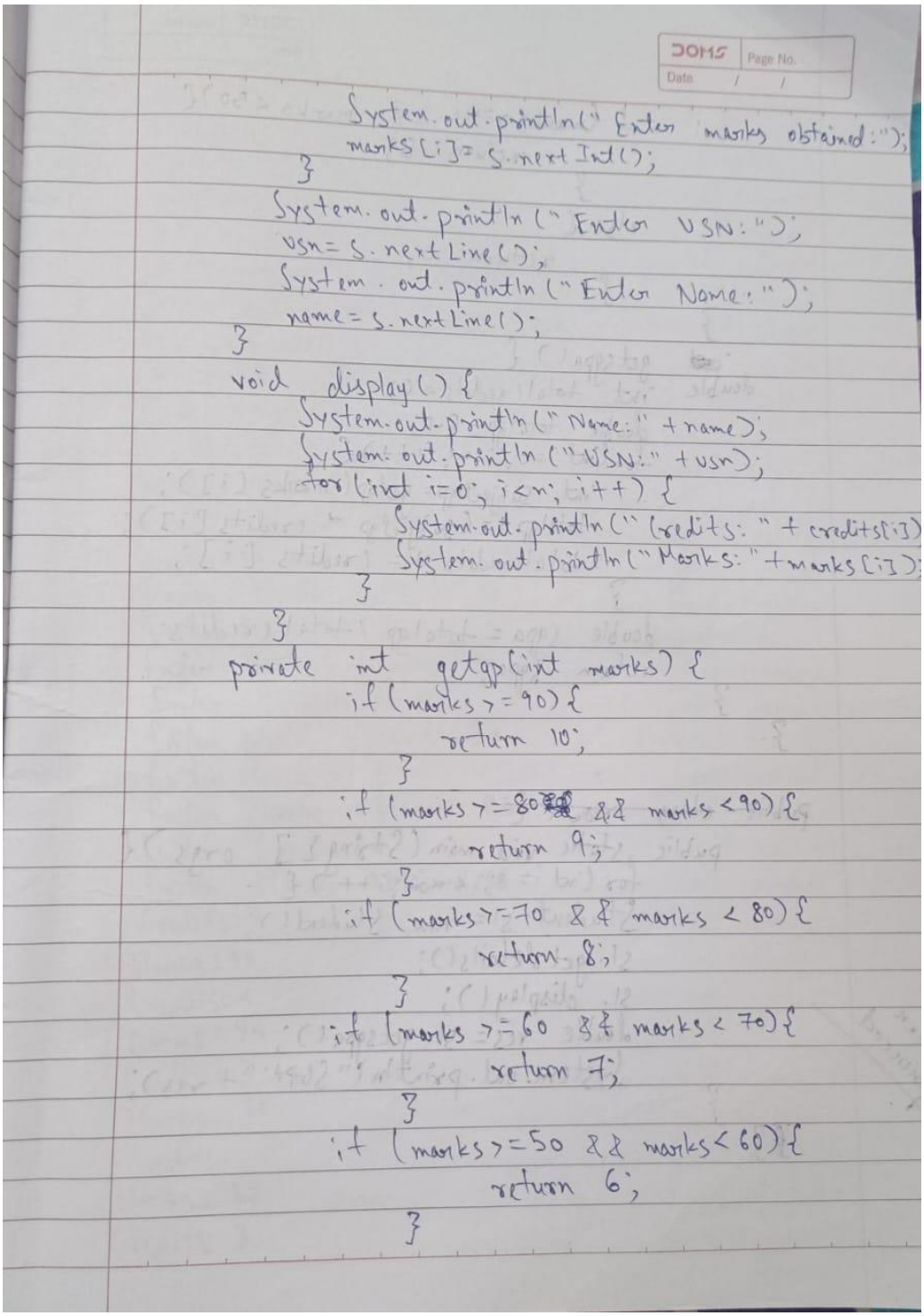
Develop a Java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student

Algorithm:



Q. Write a Java program to create a class student with members usn, name, array credits & array marks. Include methods to accept & display details & method to calculate SGPA of a student.

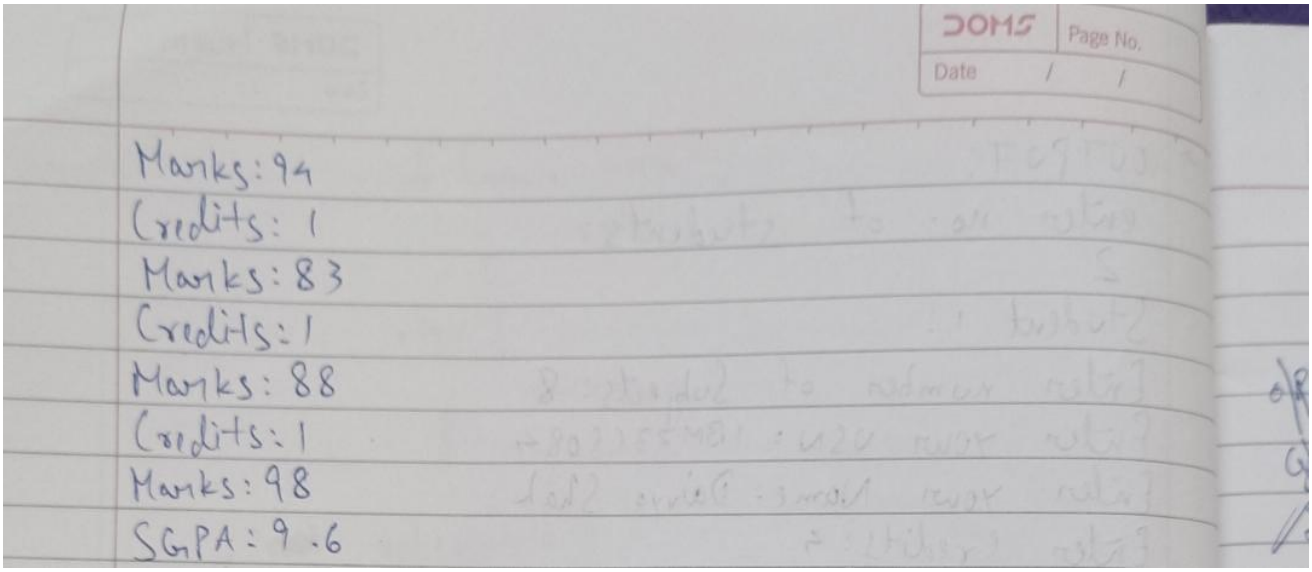
```
import java.util.Scanner;

class Student {
    String usn;
    String name;
    int[] credits;
    int[] marks;
    int n;
    void getdetails(){
        Scanner s= new Scanner(System.in);
        System.out.println("Enter no. of subjects:");
        n=s.nextInt();
        for(int i=0;i<n;i++){
            System.out.println("Enter credits:");
            credits[i]=s.nextInt();
```



```
        System.out.println("Enter marks obtained:");
        marks[i]= s.nextInt();
    }
    System.out.println("Enter USN:");
    usn= s.nextLine();
    System.out.println("Enter Name:");
    name= s.nextLine();
}
void display(){
    System.out.println("Name:" +name);
    System.out.println("USN:" +usn);
    for(int i=0; i<n; i++){
        System.out.println("Credits:" + credits[i]);
        System.out.println("Marks:" +marks[i]);
    }
}
private int getgp(int marks){
    if(marks>=90){
        return 10;
    }
    if(marks>=80 && marks<90){
        return 9;
    }
    if(marks>=70 && marks<80){
        return 8;
    }
    if(marks>=60 && marks<70){
        return 7;
    }
    if(marks>=50 && marks<60){
        return 6;
    }
```

→ OUTPUT:

```
enter no. of students:
2
Student 1:
Enter number of Subjects: 8
Enter your USN: 1BM23CS084
Enter your Name: Divya Shah
Enter Credits: 4
Enter Marks obtainde: 90
Enter Credits: 4
Enter Marks obtained: 94
Enter Credits: 3
Enter Marks obtained: 83
Enter Credits: 3
Enter Marks obtained: 88
Enter Credits: 3
Enter Marks obtained: 94
Enter Credits: 1
Enter Marks obtained: 83
Enter Credits: 1
Enter Marks obtained: 85
Enter Credits: 1
Enter Marks obtained: 98
Credits: 4
Marks: 90
Credits: 4
Marks: 94
Credits: 3
Marks: 88
Credits: 3
Marks: 88
Credits: 3
```



Code:

```
import java.util.Scanner;
class Mystud{
    String usn;
    String name;
    int[] credits;
    int[] marks;
    int n;
    public void getdetails(){
        Scanner s = new Scanner(System.in);
        System.out.println("enter usn:");
        usn = s.nextLine();
        System.out.println("enter name:");
        name = s.nextLine();
        System.out.println("enter no of subjects:");
        n = s.nextInt();
        credits = new int[n];
        marks = new int[n];
        s.nextLine();
        for(int i=0;i<n;i++){
            System.out.println("enter credits:");
            credits[i] = s.nextInt();
            System.out.println("enter marks obtained:");
            marks[i] = s.nextInt();
        }
```

```
}
public void showdetails(){
   System.out.println("Name:"+name);
   System.out.println("USN:"+usn);
   for(int i=0;i<n;i++){
      System.out.println("Credits:"+credits[i]);
      System.out.println("Marks:"+marks[i]);
   }
}
int getgp(int marks){
   if(marks>=90){
      return 10;
   }
   else if(marks>=80 && marks<90){
      return 9;
   }
   else if(marks>=70 && marks<80){
      return 8;
   }
   else if(marks>=60 && marks<70){
      return 7;
   }
   else if(marks>=50 && marks<60){
      return 6;
   }
   else if(marks>=40 && marks<50){
      return 5;
   }
   else{
      return 0;
   }
}
public void calcgp(){
   int tc = 0;
   for(int i=0;i<n;i++){
      tc+=credits[i];
   }
   int totalgp = 0;
   for(int i=0;i<n;i++){
      int gp = getgp(marks[i]);
      totalgp += (credits[i]*gp);
```

```
        }
        double cgpa = (double) totalgp /tc;
        System.out.println("CGPA:"+cgpa);
    }
}
public class Calccgpa {
    public static void main(String[] args){
        Mystud m = new Mystud();
        m.getdetails();
        m.showdetails();
        m.calcgp();
        System.out.println("Daivya Priyankkumar Shah");
        System.out.println("1BM23CS084");
    }
}
```

OUTPUT:

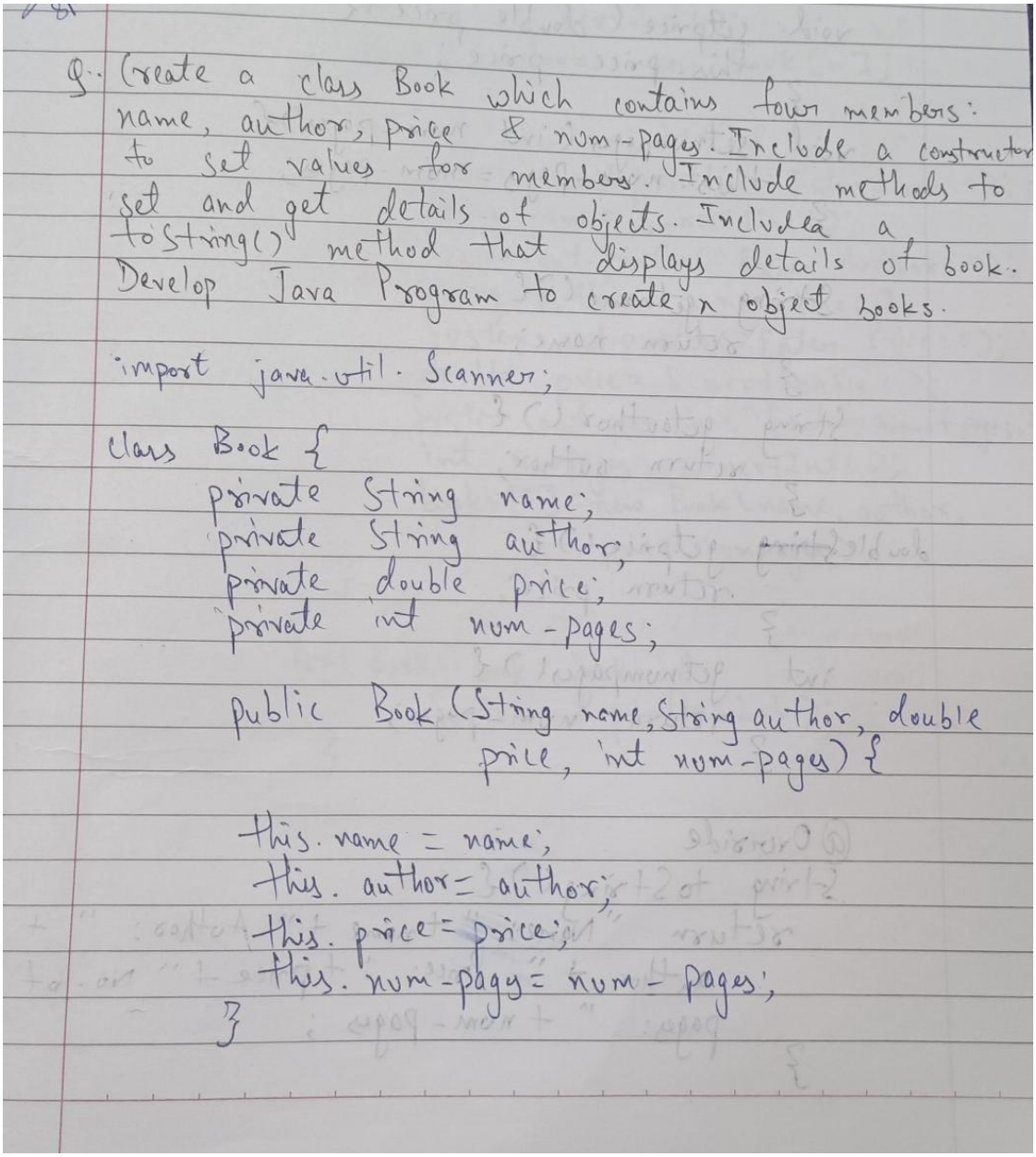
```
enter usn:
1BM23CS084
enter name:
Daivya
enter no of subjects:
5
enter credits:
4
enter marks obtained:
93
enter credits:
3
enter marks obtained:
90
enter credits:
3
enter marks obtained:
87
enter credits:
2
enter marks obtained:
81
enter credits:
4
enter marks obtained:
88
```

```
Name:Daivya
USN:1BM23CS084
Credits:4
Marks:93
Credits:3
Marks:90
Credits:3
Marks:87
Credits:2
Marks:81
Credits:4
Marks:88
CGPA:9.4375
Daivya Priyankkumar Shah
1BM23CS084
```

Program 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.

Algorithm:



Q. Create a class Book which contains four members: name, author, price & num-pages. Include a constructor to set values for members. Include methods to set and get details of objects. Includea a toString() method that displays details of book. Develop Java Program to create n object books.

```
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-pagg = num-pages;
    }
```

```
void setname (String name){
    this. name = name;
  }
void setauthor (String author){
    this. author = author;
  }
void setprice ( double price){
    this. price = price;
  }
void setnumpages (int num-pages){
    this. num-pages = num-pages;
}

String getname (){
    return name;
  }
String getauthor (){
    return author;
  }
doubleString getprice (){
    return price;
  }
int getnumpages (){
    return num-pages;
  }

@Override
String toString (){
  return "Name: " + name + " Author: " +
    author + " Price: " + price + " No. of
    pages: " + num-pages;
}
```

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

        Scanner s= new Scanner(System.in);
        System.out.println("Enter no. of books:");
        int n = s.nextInt();
        Book[] books= new Book[n];

        for(int i=0;i<n;i++){
            System.out.println("Enter Name:");
            String name= s.nextLine();
            System.out.println("Enter Author:");
            String author = s.nextLine();
            System.out.println("Enter Price:");
            double price= s.nextDouble();
            System.out.println("Enter No.of pages:");
            int numpages= s.nextInt();
            books[i]=new Book(name, author,
                    price, numpages);
        }

    for(Book book: books){
            System.out.println(book);
    }

    }
}
```

Seen
execute

→ OUTPUT:

```
Enter no. of books:
2
Name of book:
Wings of fire
Name of author:
APJ Abdul Kalam
Price:
1000
No of Pagg:
500
Name of book:
Harry Potter
Name of author:
JK Rowling
Price:
1500
No. of Pages:
750

Name: Wings of fire Author: APJ Abdul Kalam Price: 1000.0 Pag
Name: Harry Potter Author: JK Rowling Price: 1500.0 Pages: 750
```

o/p seen

15/10/24

Code:

```
import java.util.Scanner;
class Book{
   String name;
   String author;
   int price;
   int pages;
   public Book(String name,String author,int price,int pages){
      this.name = name;
      this.author = author;
      this.price = price;
      this.pages = pages;
   }
   public void setdetails(String name,String author,int price,int pages){
      this.name = name;
      this.author = author;
      this.price = price;
      this.pages = pages;
   }
   String getname(){
      return name;
   }
   String getauthor(){
      return author;
   }
   int getprice(){
      return price;
   }
   int getpages(){
      return pages;
   }
   public String toString(){
      return "Name:"+name+" Author:"+author+" Price:"+price+" Pages:"+pages;
   }
}
public class Bookdetails {
   public static void main(String[] args){
      Scanner s = new Scanner(System.in);
      System.out.println("enter no of books:");
      int n = s.nextInt();
      Book[] books = new Book[n];
```

```
        s.nextLine();
        for(int i=0;i<n;i++){
            System.out.println("enter name:");
            String name = s.nextLine();
            System.out.println("enter author:");
            String author = s.nextLine();
            System.out.println("enter price:");
            int price = s.nextInt();
            System.out.println("enter no of pages:");
            int pages = s.nextInt();
            s.nextLine();
            books[i] = new Book(name,author,price,pages);
        }
        for(Book book: books){
            System.out.println(book);
        }
        System.out.println("Daivya Priyankkumar Shah");
        System.out.println("1BM23CS084");
    }
}
```

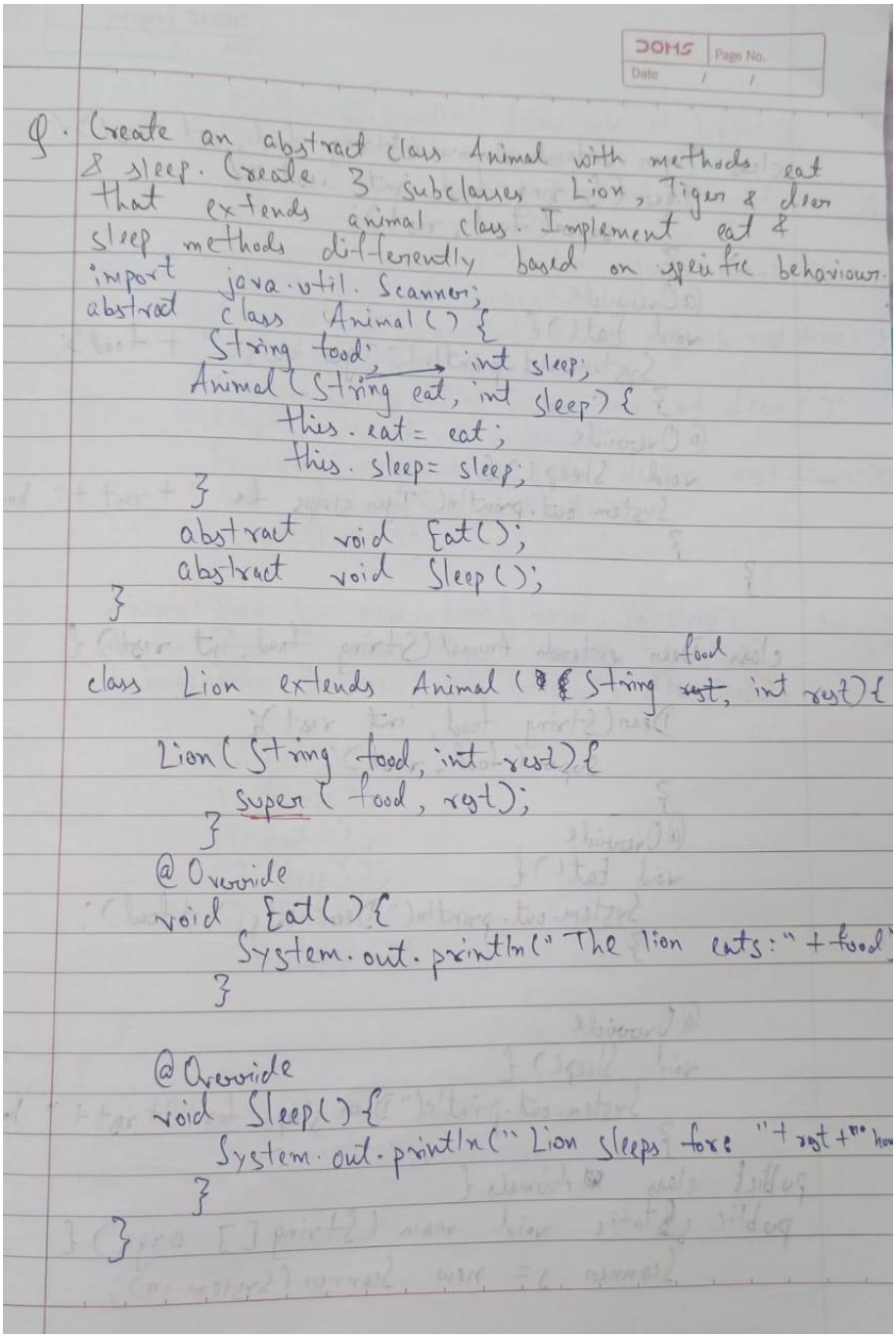
OUTPUT:

```
enter no of books:
2
enter name:
Harry Potter
enter author:
JK Rowling
enter price:
800
enter no of pages:
600
enter name:
Wings Of Fire
enter author:
APJ Abdul Kalam
enter price:
500
enter no of pages:
800
Name:Harry Potter Author:JK Rowling Price:800 Pages:600
Name:Wings Of Fire Author:APJ Abdul Kalam Price:500 Pages:800
Daivya Priyankkumar Shah
1BM23CS084
```

Program 4:

Create an abstract class Animal with methods eat and sleep. Create 3 subclasses Lion, Tiger and deer that extends animal class. Implement these methods differently.

Algorithm:

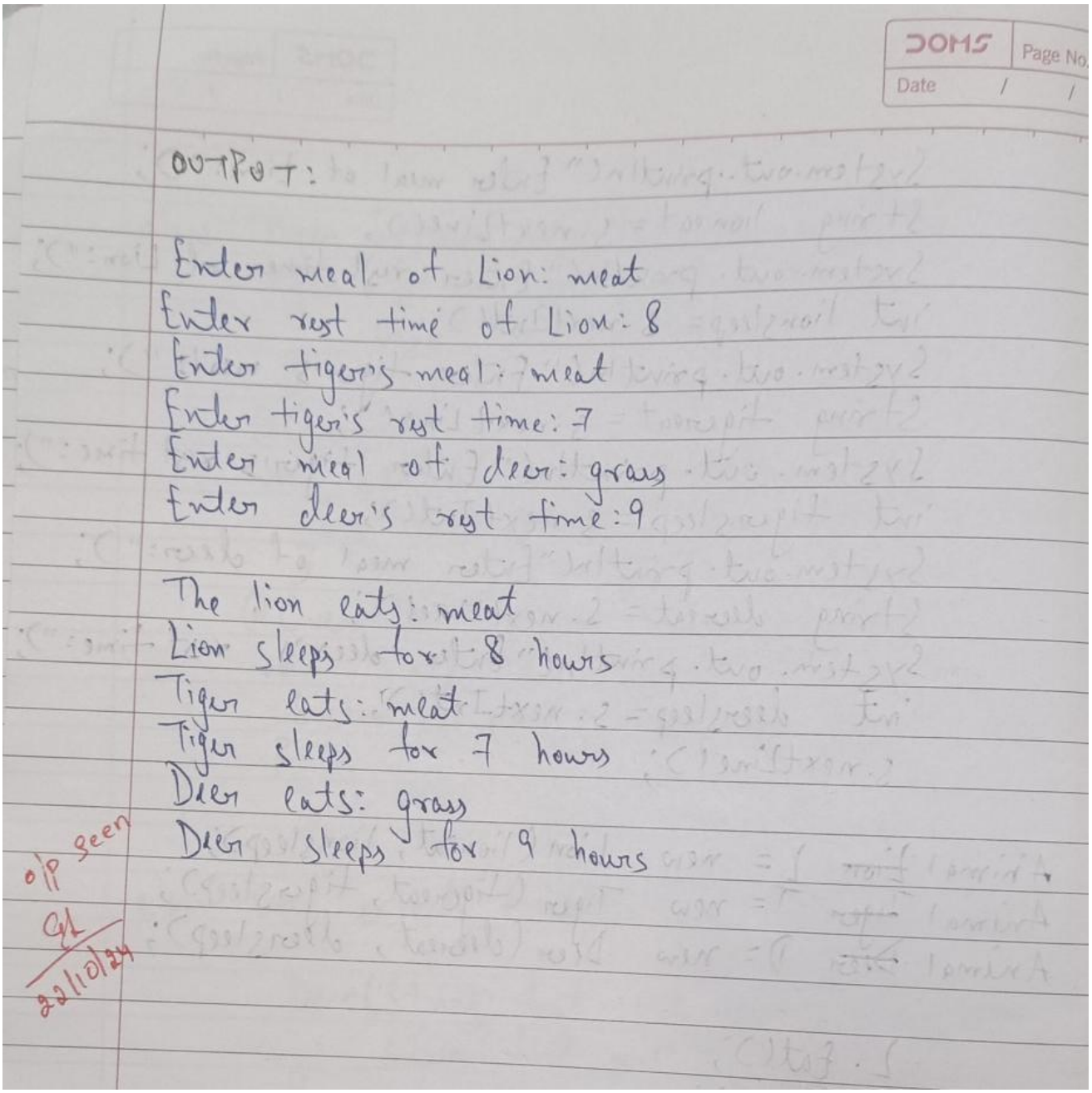


```
Q. Create an abstract class Animal with methods eat
& sleep. Create 3 subclasses Lion, Tiger & deer
that extends animal class. Implement eat &
sleep methods differently based on specific behaviours.
import java.util.Scanner;
abstract class Animal(){
    String food;    int sleep;
    Animal(String eat, int sleep){
        this.eat = eat;
        this.sleep= sleep;
    }
    abstract void Eat();
    abstract void Sleep();
}

class Lion extends Animal(String food, int rest){

    Lion(String food, int rest){
        super(food, rest);
    }
    @Override
    void Eat(){
        System.out.println("The lion eats:" +food
    }

    @Override
    void Sleep(){
        System.out.println("Lion sleeps for: "+rest+"
    }
}
```

```
class Tiger extends Animal{String food, int rest){
    Tiger (String food, int rest){
        super(food, rest);
    }
    @Override
    void Eat(){
        System.out.println("Tiger eats:" + food);
    }
    @Override
    void Sleep(){
        System.out.println("Tiger sleeps for " + rest + "
    }
}

class Deer extends Animal(String food, int rest){
    Deer(String food, int rest){
        super(food, rest);
    }
    @Override
    void Eat(){
        System.out.println("Deer eats:" + food);
    }

    @Override
    void Sleep(){
        System.out.println("Deer sleeps for "+ rest + " hour
public class Animals {
public static void main (String[] args){
    Scanner s= new Scanner(System.in);
```

```
System.out.println("Enter meal of Lion:");
String lioneat = s.nextLine();
System.out.println("Enter rest time of Lion:");
int lionsleep = s.nextInt();
System.out.println("Enter tiger's meal:");
String tigereat = s.nextLine();
System.out.println("Enter tiger's rest time:");
int tigersleep = s.nextInt();
System.out.println("Enter meal of deer:");
String deereat = s.nextLine();
System.out.println("Enter deer's rest time:");
int deersleep = s.nextInt();
s.nextLine();

Animal Lion L = new Lion(lioneat, lionsleep);
Animal Tiger T = new Tiger(tigereat, tigersleep);
Animal Dter D = new Dter(dteat, dtersleep);

L.Eat();
L.Sleep();
T.Eat();
T.Sleep();
D.Eat();
D.Sleep();

}
}
```

Seen / execute



Code:

```
import java.util.Scanner;

abstract class Animal {
    String food;
    int sleepHours;
    public Animal(String food, int sleepHours) {
        this.food = food;
        this.sleepHours = sleepHours;
```

```
    }
    abstract void eat();
    abstract void sleep();
}
class Lion extends Animal {
    public Lion(String food, int sleepHours) {
        super(food, sleepHours);
    }

    @Override
    public void eat() {
        System.out.println("The lion eats: " + food);
    }

    @Override
    public void sleep() {
        System.out.println("The lion sleeps for: " + sleepHours + " hours.");
    }
}
class Tiger extends Animal {
    public Tiger(String food, int sleepHours) {
        super(food, sleepHours);
    }
    @Override
    public void eat() {
        System.out.println("The tiger eats: " + food);
    }
    @Override
    public void sleep() {
        System.out.println("The tiger sleeps for: " + sleepHours + " hours.");
    }
}
class Deer extends Animal {
    public Deer(String food, int sleepHours) {
        super(food, sleepHours);
    }
    @Override
    public void eat() {
        System.out.println("The deer eats: " + food);
    }
    @Override
```

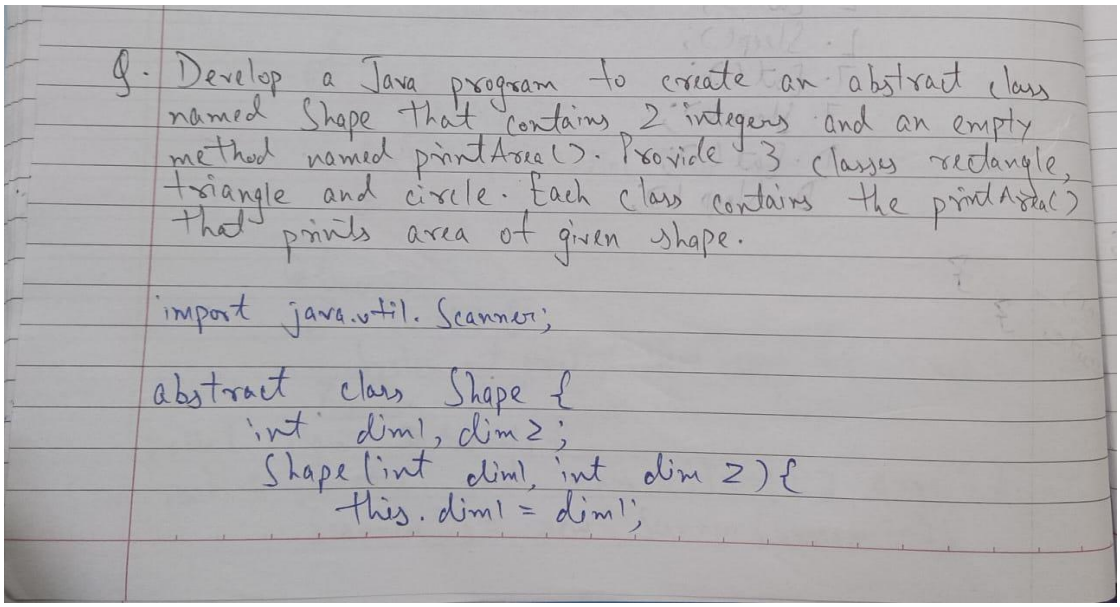
```
    public void sleep() {
        System.out.println("The deer sleeps for: " + sleepHours + " hours.");
    }
}
public class Animals {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter food for Lion: ");
        String lionFood = scanner.nextLine();
        System.out.print("Enter sleep hours for Lion: ");
        int lionSleepHours = scanner.nextInt();
        Animal lion = new Lion(lionFood, lionSleepHours);
        scanner.nextLine();
        System.out.print("Enter food for Tiger: ");
        String tigerFood = scanner.nextLine();
        System.out.print("Enter sleep hours for Tiger: ");
        int tigerSleepHours = scanner.nextInt();
        Animal tiger = new Tiger(tigerFood, tigerSleepHours);
        scanner.nextLine();
        System.out.print("Enter food for Deer: ");
        String deerFood = scanner.nextLine();
        System.out.print("Enter sleep hours for Deer: ");
        int deerSleepHours = scanner.nextInt();
        Animal deer = new Deer(deerFood, deerSleepHours);
        lion.eat();
        lion.sleep();

        tiger.eat();
        tiger.sleep();

        deer.eat();
        deer.sleep();
    }
}
```

OUTPUT:

```
Enter food for Lion: meat
Enter sleep hours for Lion: 8
Enter food for Tiger: meat
Enter sleep hours for Tiger: 7
Enter food for Deer: grass
Enter sleep hours for Deer: 10
The lion eats: meat
The lion sleeps for: 8 hours.
The tiger eats: meat
The tiger sleeps for: 7 hours.
The deer eats: grass
The deer sleeps for: 10 hours.
Daivya Priyankkumar Shah
1BM23CS084
```

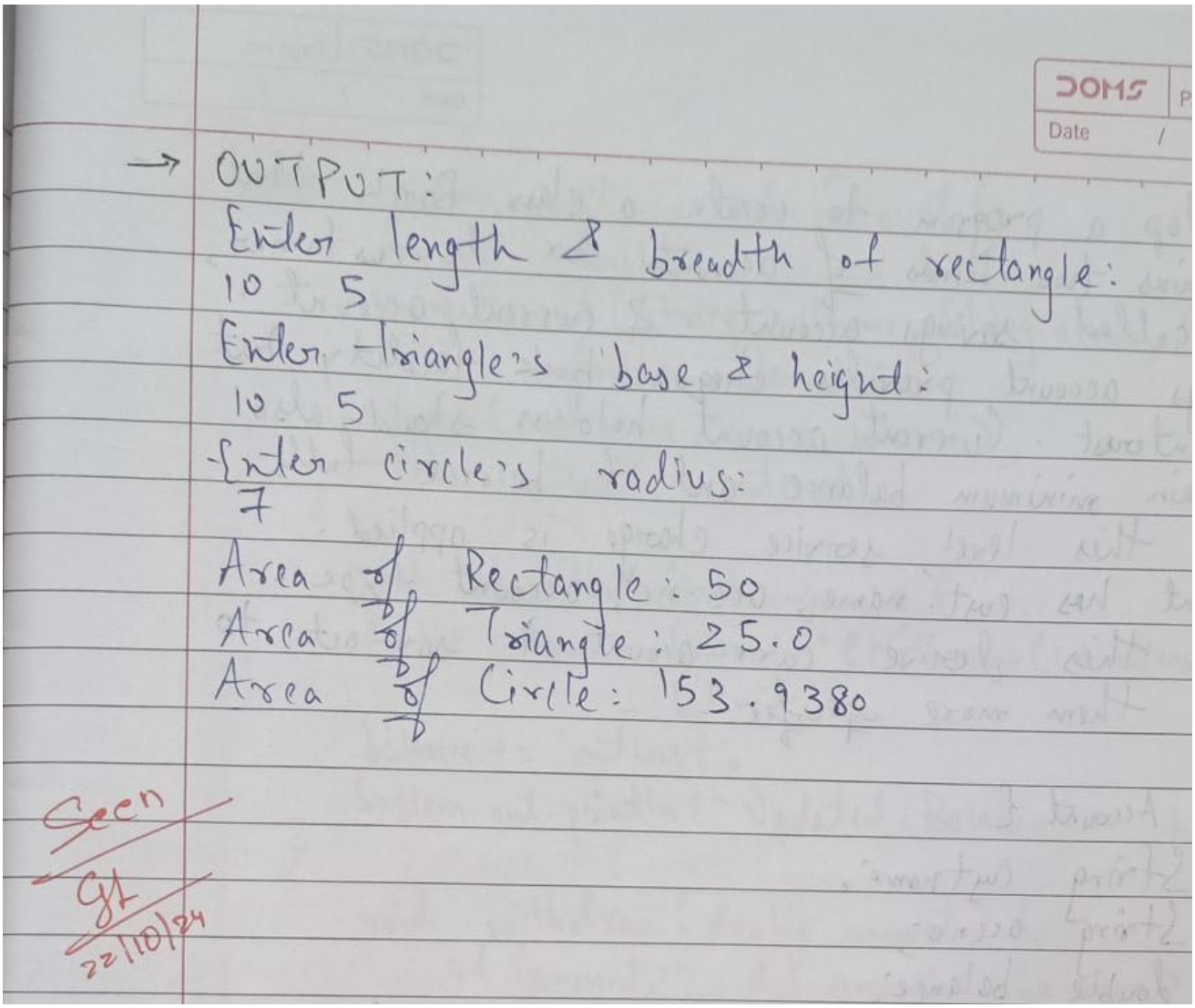
Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea()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

Algorithm:



```
        this.dim2 = dim2;
    }
    abstract void print();
}

class Rectangle extends Shape {
    Rectangle(int len, int br) {
        super(len, br);
    }
    @Override
    void print() {
        int area = dim1 * dim2;
        System.out.println("Area of Rectangle: " + area);
    }
}

class Triangle extends Shape {
    Triangle(int base, int height) {
        super(base, height);
    }
    @Override
    void print() {
        double area = 0.5 * dim1 * dim2;
        System.out.println("Area of triangle: " + area);
    }
}

class Circle extends Shape {
    Circle(int r) {
        super(r, 0);
    }
    @Override
```

```
    void print() {
        double area = 3.14 * dim1 * dim1;
        System.out.println("Area of Circle: "+area);
    }
}

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

        Scanner s= new Scanner(System.in);

        System.out.println("Enter length & breadth of
            rectangle :");
        int rectlen= s.nextInt();
        int rectbr= s.nextInt();

        System.out.println("Enter triangle's base & hei
        int tribs = s.nextInt();
        int triht = s.nextInt();

        System.out.println("Enter circle's radius:")
        int r = s.nextInt();

        Shape rectangle= new Rectangle(rectlen, recth
        Shape triangle = new Triangle(tribs, triht);
        Shape circle= new Circle(r);

        rectangle.print();
        triangle.print();
        circle.print();
    }
}
```



Code:

```
abstract class Shapes{
   int dim1;
   int dim2;
   Shapes(int dim1,int dim2){
      this.dim1 = dim1;
      this.dim2 = dim2;
   }
   abstract void getArea();
}
class Rectangles extends Shapes{
   Rectangles(int len,int br){
      super(len,br);
   }
```

```
    @Override
    public void getArea(){
        int area = dim1*dim2;
        System.out.println("area of rectangle:"+area);
    }
}
class Triangles extends Shapes{
    Triangles(int base,int height){
        super(base,height);
    }
    @Override
    public void getArea(){
        double area = 0.5*dim1*dim2;
        System.out.println("area of triangle:"+area);
    }
}
class Circles extends Shapes{
    Circles(int r){
        super(r,0);
    }
    @Override
    public void getArea(){
        double area = Math.PI*dim1*dim1;
        System.out.println("area of circle:"+area);
    }
}
public class Shapesabs {
    public static void main(String[] args){
        Rectangles r = new Rectangles(4,5);
        Triangles t = new Triangles(10,20);
        Circles c = new Circles(10);
        r.getArea();
        t.getArea();
        c.getArea();
        System.out.println("Daivya Priyankkumar Shah");
        System.out.println("1BM23CS084");
    }
}
```

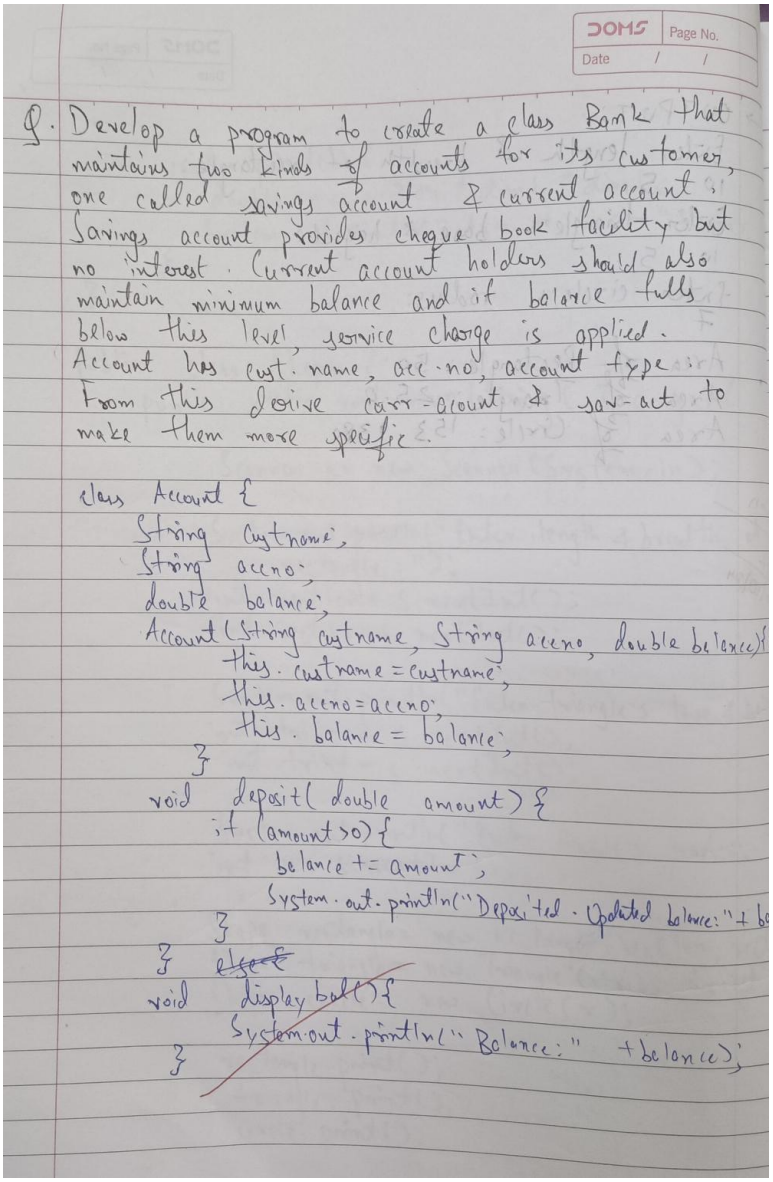
OUTPUT:

```
area of rectangle:20
area of triangle:100.0
area of circle:314.1592653589793
Daivya Priyankkumar Shah
1BM23CS084
PS C:\Users\daivy\IdeaProjects\JavaPractice\src>
```

Program 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.

Algorithm:

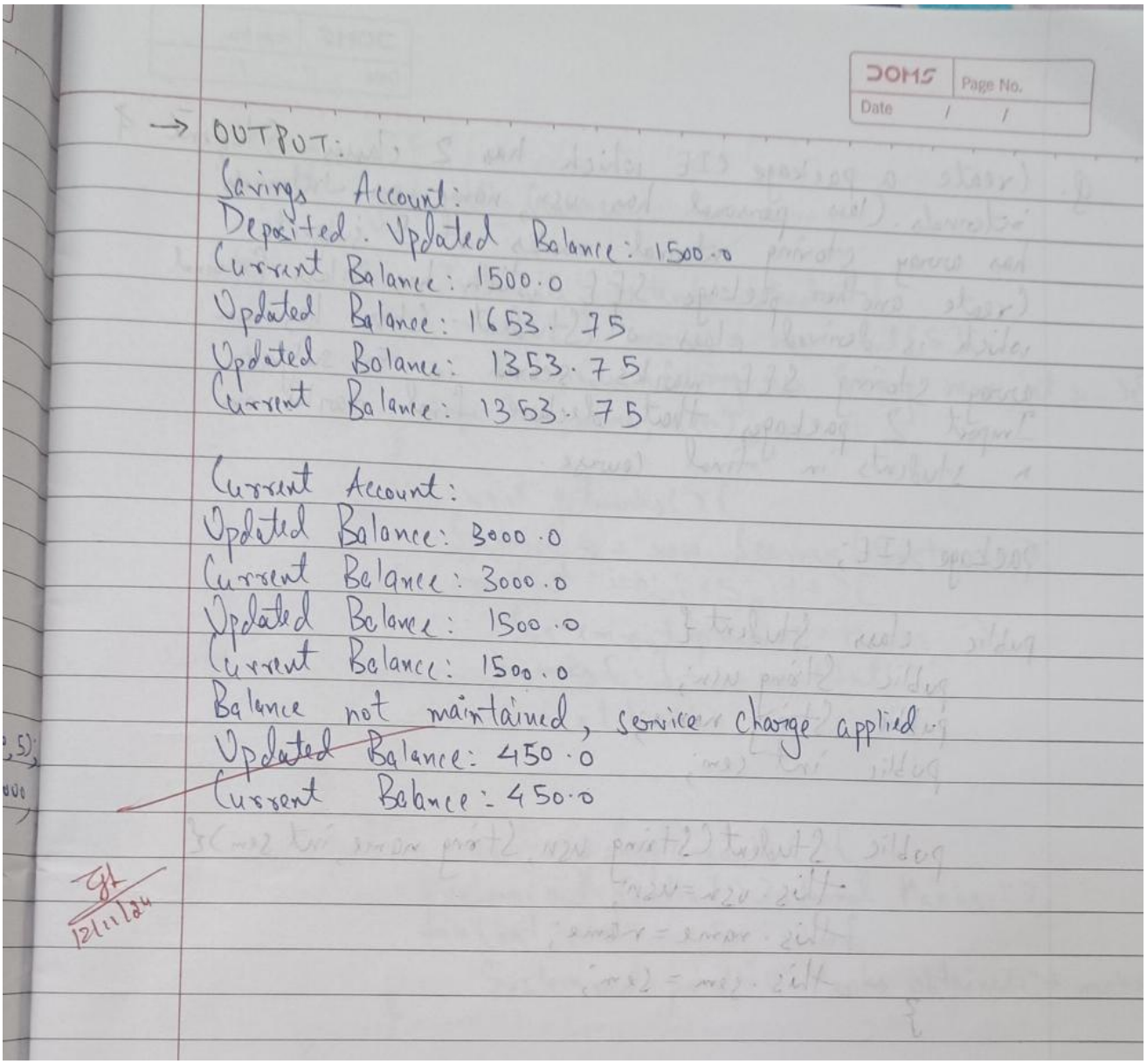


Q. Develop a program to create a class Bank that maintains two kinds of accounts for its customer, one called savings account & current account; Savings account provides cheque book facility but no interest. Current account holders should also maintain minimum balance and if balance falls below this level, service charge is applied. Account has cust. name, acc. no, account type; From this derive curr-acount & sav-act to make them more specific.

```
class Account {
    String custname;
    String accno;
    double balance;
    Account(String custname, String accno, double balance){
        this.custname = custname;
        this.accno = accno;
        this.balance = balance;
    }
void deposit(double amount) {
    if (amount > 0) {
        balance += amount;
        System.out.println("Deposited. Updated balance: " + b
    }
}
void displayBal(){
    System.out.println("Balance: " + balance);
}
```

```
class savingsaccount extends Account {
    double interst;
    savingsaccount (String custname, String accno, double balance,
                    double interst) {
    super (custname, accno, balance);
    this.interst = interst;
}
    void compinterst (int years) {
    double interst = balance * (Math.pow (1 + interst/100, years)
                     - balance;
    balance += interst;
    System.out.println ("Updated Balance:", balance);
}
    void withdraw (double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println ("Balance:" + balance);
        }
        else {
            System.out.println ("Invalid Balance");
        }
    }
}
class currentaccount extends Account {
    double minbal;
    double service;
    currentaccount (String custname, String accno, double balance,
                    double minbal, double service) {
    super (custname, accno, balance);
    this.minbal = minbal;
    this.service = service;
}
```

```
    void withdraw (double amount){
        balance -= amount;
        if (balance < minbal){
            balance -= service;
            System.out.println("Balance not maintained,
            service charge applied");
        }
        System.out.println("Updated Balance:"+balance);
    }
}

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

        savingsaccount s1 = new savingsaccount("Aman", "SA1234",
        currentaccount c1 = new currentaccount("Bramha", "CA1234",
                                                1000, 50);
        System.out.println("Savings Account:");
        s1.deposit(500);
        s1.display();
        s1.compinterest(2);
        s1.withdraw(300);
        s1.display();

        System.out.println("Current Account:");
        c1.deposit(1000);
        c1.display();
        c1.withdraw(1500);
        c1.display();
        c1.withdraw(1000);
        c1.display();
    }
}
```



Code:

```
import java.util.Scanner;

class Account {
    protected String customerName;
    protected String accountNumber;
    protected String accountType;
    protected double balance;

    public Account(String customerName, String accountNumber, String accountType, double
balance) {
        this.customerName = customerName;
```

```
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = balance;
    }

    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposit successful. Updated balance: " + balance);
        } else {
            System.out.println("Invalid deposit amount.");
        }
    }

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

class SavAcct extends Account {
    private double interestRate;

    public SavAcct(String customerName, String accountNumber, double balance, double
interestRate) {
        super(customerName, accountNumber, "Savings", balance);
        this.interestRate = interestRate;
    }

    public void computeAndDepositInterest() {
        double interest = balance * (interestRate / 100);
        balance += interest;
        System.out.println("Interest of " + interest + " deposited. Updated balance: " + balance);
    }

    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawal successful. Updated balance: " + balance);
        } else {
            System.out.println("Invalid or insufficient funds for withdrawal.");
        }
```

```
    }
}

class CurAcct extends Account {
    private static final double MINIMUM_BALANCE = 1000.0;
    private static final double PENALTY = 100.0;

    public CurAcct(String customerName, String accountNumber, double balance) {
        super(customerName, accountNumber, "Current", balance);
    }

    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawal successful. Updated balance: " + balance);

            if (balance < MINIMUM_BALANCE) {
                balance -= PENALTY;
                System.out.println("Balance below minimum. Penalty of " + PENALTY + " imposed.
Updated balance: " + balance);
            }
        } else {
            System.out.println("Invalid or insufficient funds for withdrawal.");
        }
    }
}

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

        System.out.println("Enter details for Savings Account:");
        System.out.print("Customer Name: ");
        String savName = scanner.nextLine();
        System.out.print("Account Number: ");
        String savAccNo = scanner.nextLine();
        System.out.print("Initial Balance: ");
        double savBalance = scanner.nextDouble();
        System.out.print("Interest Rate: ");
        double interestRate = scanner.nextDouble();
        SavAcct savings = new SavAcct(savName, savAccNo, savBalance, interestRate);
```

```
        scanner.nextLine();
        System.out.println("\nEnter details for Current Account:");
        System.out.print("Customer Name: ");
        String curName = scanner.nextLine();
        System.out.print("Account Number: ");
        String curAccNo = scanner.nextLine();
        System.out.print("Initial Balance: ");
        double curBalance = scanner.nextDouble();
        CurAcct current = new CurAcct(curName, curAccNo, curBalance);

        System.out.println("\nPerforming operations on Savings Account:");
        savings.deposit(1000);
        savings.computeAndDepositInterest();
        savings.withdraw(500);
        savings.displayBalance();

        System.out.println("\nPerforming operations on Current Account:");
        current.deposit(2000);
        current.withdraw(2500);
        current.withdraw(500);
        current.displayBalance();
        System.out.println("Daivya Priyankkumar Shah");
        System.out.println("1BM23CS084");
        scanner.close();
    }
}
```

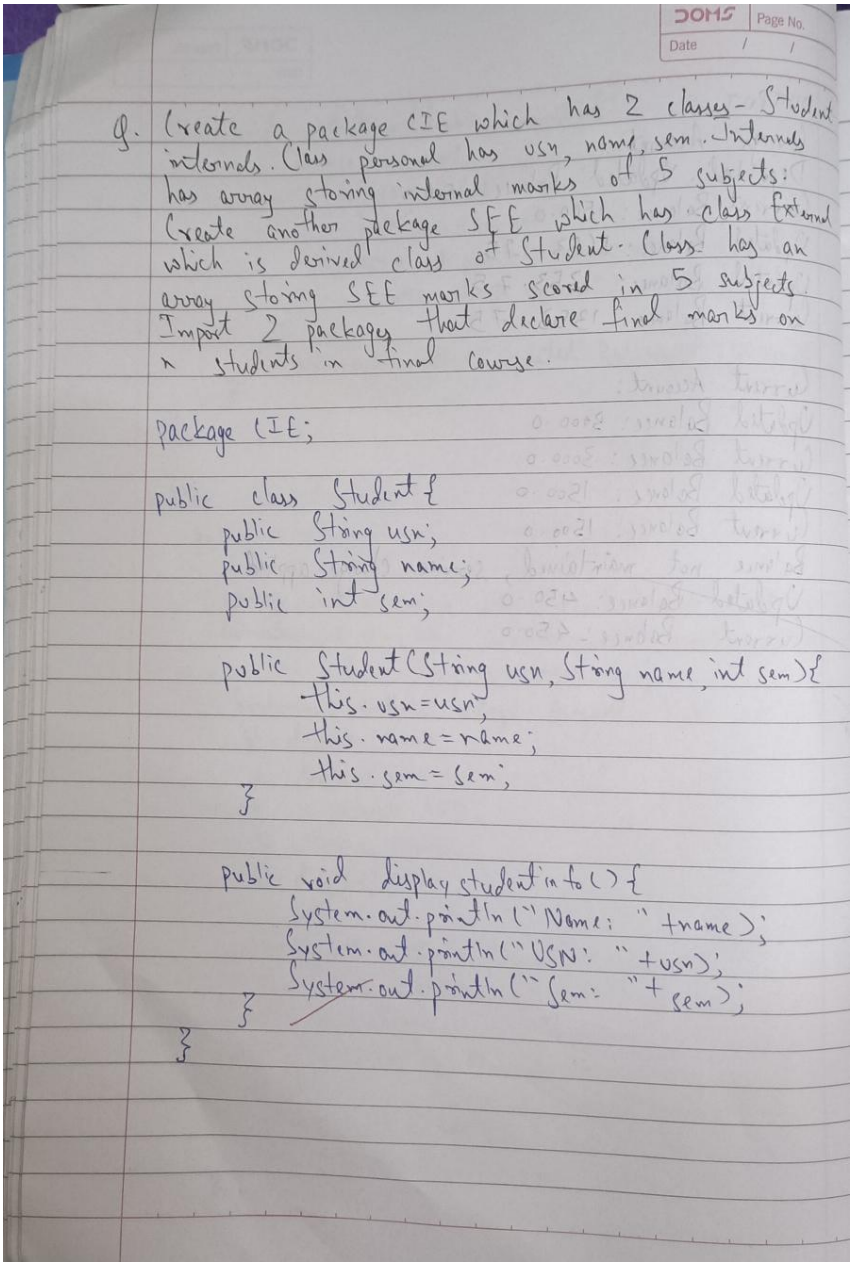
OUTPUT:

```
PS C:\Users\daivy\IdeaProjects\JavaPractice\src> javac Banks.java
PS C:\Users\daivy\IdeaProjects\JavaPractice\src> java Banks
Enter details for Savings Account:
Customer Name: Daivya
Account Number: 10
Initial Balance: 10000
Interest Rate: 10

Enter details for Current Account:
Customer Name: Daivya
Account Number: 11
Initial Balance: 100000

Performing operations on Savings Account:
Deposit successful. Updated balance: 11000.0
Interest of 1100.0 deposited. Updated balance: 12100.0
Withdrawal successful. Updated balance: 11600.0
Current balance: 11600.0

Performing operations on Current Account:
Deposit successful. Updated balance: 102000.0
Withdrawal successful. Updated balance: 99500.0
Withdrawal successful. Updated balance: 99000.0
Current balance: 99000.0
Daivya Priyankkumar Shah
1BM23CS084
```

Program 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

Algorithm:



Q. Create a package CIE which has 2 classes - Student, internals. Class personal has usn, name, sem. Internals has array storing internal marks of 5 subjects. Create another package SEE which has class external which is derived class of Student. Class has an array storing SEE marks scored in 5 subjects. Import 2 package that declare final marks of n students in final course.

```
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 displaystudentinfo(){
        System.out.println("Name: "+name);
        System.out.println("USN: "+usn);
        System.out.println("Sem: "+sem);
    }
}
```

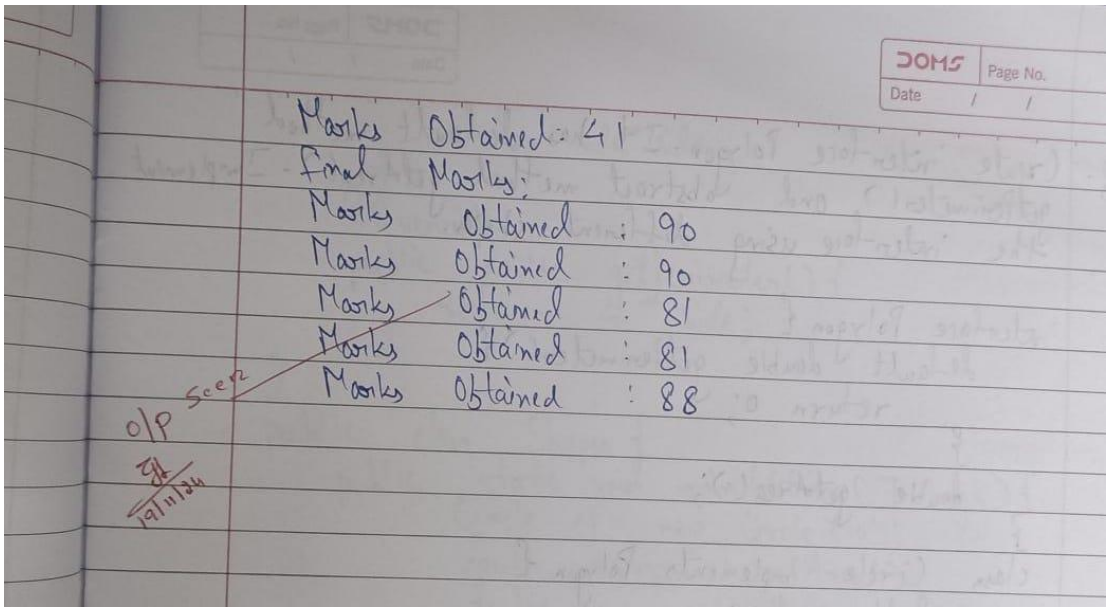
```
package CIE;
import java.util.Scanner;

public class Internals extends Student{
        public int[] marks = new int[5];
        public Internals(String usn, String name, int sem){
            super(usn, name, sem);
        }
        public void getmarks(){
            Scanner s = new Scanner(System.in);
            for(int i=0; i<5; i++){
                System.out.println("Marks:");
                marks[i] = s.nextInt();
                s.nextLine();
            }
        }
        public void displaymarks(){
            System.out.println("Internal Marks:");
            for(int i=0; i<5; i++){
                System.out.println("Marks obtained:"+ marks[i]);
            }
        }
}

package SEE;
import CIE.*;
import java.util.Scanner;

public class Externals extends Internals{
        public int[] ext = new int[5];
        public Externals(String usn, String name, int sem){
            super(usn, name, sem); }
```

```
public void getextmarks(){
    Scanner s= new Scanner(System.in);
    for(int i=0; i<5; i++){
        System.out.println("Marks:");
        ext[i]= s.nextInt();
        s.nextLine();
    }
}

public void displayextmarks(){
    System.out.println("External Marks:");
    for(int i=0; i<5; i++){
        System.out.println("Marks obtained:"+ext[i]);
    }
}

public void finalmarks(){
    System.out.println("Final Marks:");
    for(int i=0; i<5; i++){
        int final= marks[i]+ ext[i];
        System.out.println("Marks obtained:"+final);
    }
}

import ~~CE.*~~;
import SEE.*;
import java.util.Scanner;
public class Main{
    public static void main(String[] args){
        Scanner s= new Scanner(System.in);
```

```
System.out.println("No. of Students:");
int n=S.nextInt();
s.nextLine();

for(int i=0;i<n;i++){
    System.out.println("Enter details:");
    System.out.println("Name:");
    String name = S.nextLine();
    System.out.println("USN:");
    String usn = s.nextLine();
    System.out.println("Sem:");
    int sem = S.nextInt();

    Externals student = new Externals(usn,name,sem

    student.displaystudentinfo();
    System.out.println("Enter internal marks:")
    student.getmarks();
    student.displaymarks();

    System.out.println("Enter external marks:");
    student.getextmarks();
    student.displayextmarks();

    student.finalmarks();

}
}
```

→ OUTPUT

```
Enter no. of students: 1
Enter Details:
Name: Aditya
USN: 1BM23CS001
Sem: 3
Name: Aditya
USN: 1BM23CS001
Sem: 3
Enter Internal Marks:
Marks: 50
Marks: 45
Marks: 37
Marks: 35
Marks: 47
Internal Marks:
Marks Obtained : 50
Marks Obtained : 45
Marks Obtained : 37
Marks Obtained : 35
Marks Obtained : 47
Enter External Marks:
Marks : 40
Marks : 45
Marks : 44
Marks : 46
Marks : 41
External Marks:
Marks Obtained : 40
Marks Obtained : 45
Marks Obtained : 44
Marks Obtained : 46
```



Code:

```
package CIE;
public class Personal {
    String usn;
    String name;
    int sem;
    public Personal(String usn,String name,int sem){
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }
    public void displaydetails(){
        System.out.println("Name:"+name);
        System.out.println("USN:"+usn);
        System.out.println("Sem:"+sem);
    }
}
```

```
package CIE;
import java.util.Scanner;
public class Internals extends Personal {
    public Internals(String usn,String name,int sem){
        super(usn,name,sem);
    }
```

```
    public int[] intmarks = new int[5];
    public void getintmarks(){
        Scanner s = new Scanner(System.in);
        for(int i=0;i<5;i++) {
            System.out.println("enter internal marks:");
            intmarks[i] = s.nextInt();
            s.nextLine();
        }
    }
    public void dispintmarks(){
        for(int i=0;i<5;i++){
            System.out.println("internal marks:"+intmarks[i]);
        }
    }
}

package SEE;
import CIE.*;
import java.util.Scanner;
public class Externals extends Internals{
    public Externals(String usn,String name,int sem){
        super(usn,name,sem);
    }
    int[] extmarks = new int[5];
    public void getextmarks(){
        Scanner s = new Scanner(System.in);
        for(int i=0;i<5;i++){
            System.out.println("enter external marks:");
            extmarks[i] = s.nextInt();
            s.nextLine();
        }
    }
    public void displayextmarks(){
        for(int i=0;i<5;i++) {
            System.out.println("external marks:" + extmarks[i]);
        }
    }
    public void totalmarks(){
        for(int i=0;i<5;i++){
            int total = intmarks[i]+extmarks[i];
            System.out.println("total marks:"+total);
```

```
        }
    }
}
```

```
import CIE.*;
import SEE.*;
import java.util.Scanner;
public class Mainmarks {
    public static void main(String[] args){
        Scanner s = new Scanner(System.in);
        System.out.println("enter no of students:");
        int n = s.nextInt();
        s.nextLine();
        for(int i=0;i<n;i++){
            System.out.println("name:");
            String name = s.nextLine();
            System.out.println("usn:");
            String usn = s.nextLine();
            System.out.println("Sem:");
            int sem = s.nextInt();
            s.nextLine();
            Externals e = new Externals(usn,name,sem);
            e.displaydetails();
            e.getintmarks();
            e.dispintmarks();
            e.getextmarks();
            e.displayextmarks();
            e.totalmarks();
            System.out.println("Daivya Priyankkumar Shah");
            System.out.println("1BM23CS084");
        }
    }
}
```

OUTPUT:

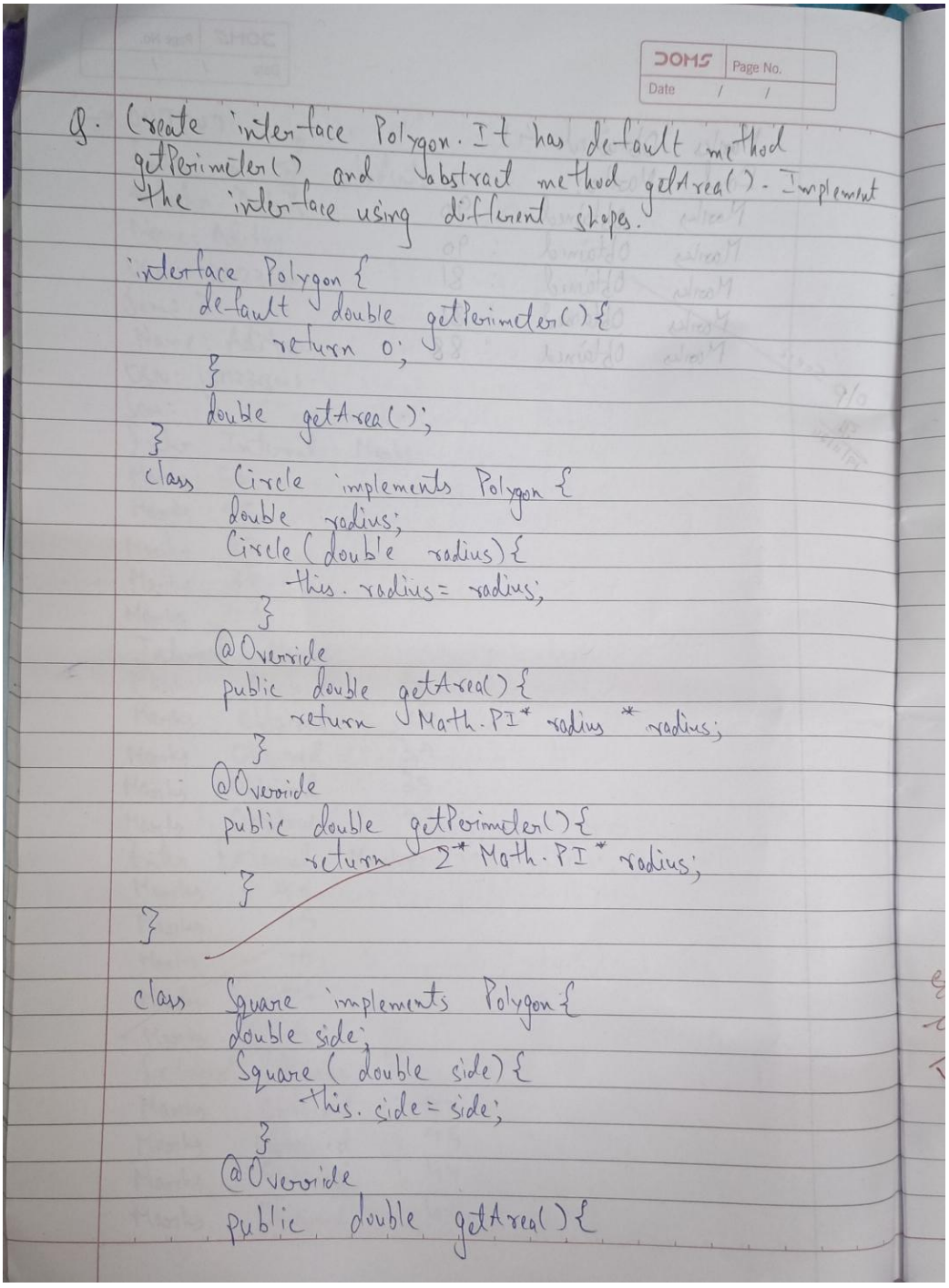
```
enter no of students:
1
name:
Daivya
usn:
1BM23CS084
Sem:
3
Name:Daivya
USN:1BM23CS084
Sem:3
enter internal marks:
45
enter internal marks:
43
enter internal marks:
42
enter internal marks:
41
enter internal marks:
40
internal marks:45
internal marks:43
internal marks:42
internal marks:41
internal marks:40
enter external marks:
45
```

```
enter external marks:
44
enter external marks:
43
enter external marks:
42
enter external marks:
41
external marks:45
external marks:44
external marks:43
external marks:42
external marks:41
total marks:90
total marks:87
total marks:85
total marks:83
total marks:81
Daivya Priyankkumar Shah
1BM23CS084
```

Program 7:

Create an interface Polygon. It has a default method getPerimeter() and abstract method getArea(). Implement the interface using different shapes.

Algorithm:



Q. Create interface Polygon. It has default method getPerimeter() and abstract method getArea(). Implement the interface using different shapes.

```
interface Polygon {
    default double getPerimeter(){
        return 0;
    }
    double getArea();
}
class Circle implements Polygon {
    double radius;
    Circle(double radius){
        this.radius = radius;
    }
    @Override
    public double getArea(){
        return Math.PI * radius * radius;
    }
    @Override
    public double getPerimeter(){
        return 2 * Math.PI * radius;
    }
}

class Square implements Polygon{
    double side;
    Square(double side){
        this.side = side;
    }
    @Override
    public double getArea(){
```

```
        return side * side;
    }
    @Override
    public double getPerimeter() {
        return 4 * side;
    }
}
public class Shapes {
    public static void main(String[] args) {
        Circle c1 = new Circle(10);
        double carea = c1.getArea();
        double cper = c1.getPerimeter();
        Square s1 = new Square(10);
        double sarea = s1.getArea();
        double sper = s1.getPerimeter();
        System.out.println("Area of circle: " + carea);
        System.out.println("Perimeter of circle: " + cper);
        System.out.println("Area of square: " + sarea);
        System.out.println("Perimeter of square: " + sper);
    }
}
```

→ OUTPUT

Executed
GJ
14/11/24

```
Area of circle: 314.1592653
Perimeter of Circle : 62.831853
Area of square: 100.0
Perimeter of square: 40.0
```

Code:

```
interface Polygon {
   default double getPerimeter(){
      return 0;
   }
   double getArea();
}
class Circless implements Polygon {
   double radius;
   Circless(double radius) {
      this.radius = radius;
   }
   @Override
   public double getArea() {
      return Math.PI * radius * radius;
   }
   @Override
   public double getPerimeter(){
      return 2*Math.PI*radius;
   }
}
class Square implements Polygon {
   double side;
   Square(double side) {
      this.side = side;
   }
   @Override
   public double getArea() {
      return side*side;
   }
   @Override
   public double getPerimeter(){
      return 4*side;
   }
}
public class Shapess{
   public static void main(String[] args){
      Circless c1 = new Circless(10);
      double carea=c1.getArea();
      double cperimeter=c1.getPerimeter();
      Square s1 = new Square(10);
```

```
        double sarea=s1.getArea();
        double sperimeter=s1.getPerimeter();
        System.out.println("Area of Circle:"+carea);
        System.out.println("Perimeter of Circle:"+cperimeter);
        System.out.println("Area of Sqaure:"+sarea);
        System.out.println("Perimeter of Square:"+sperimeter);
        System.out.println("Daivya Priyankkumar Shah");
        System.out.println("1BM23CS084");
    }
}
```

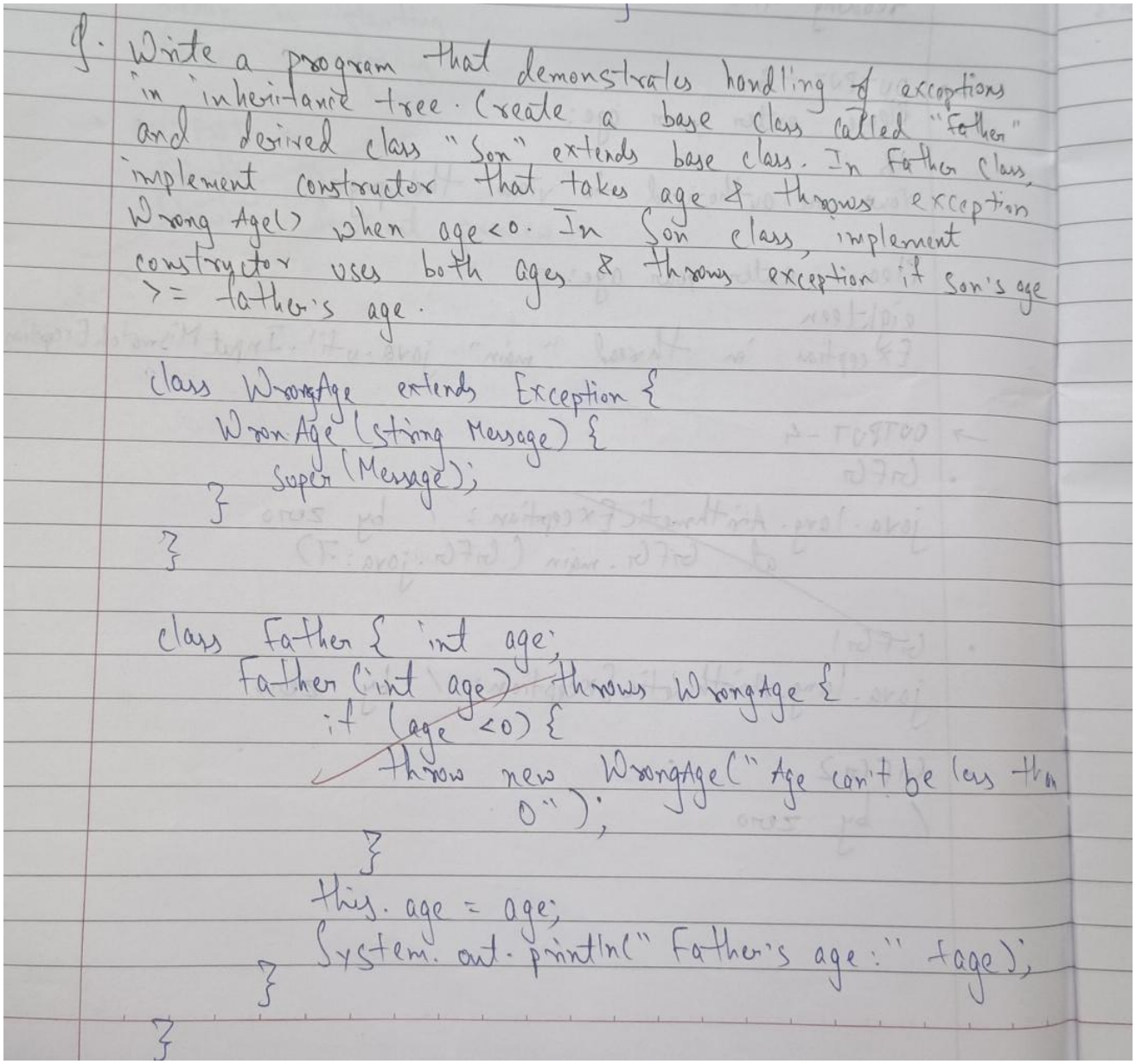
OUTPUT:

```
Area of Circle:314.1592653589793
Perimeter of Circle:62.83185307179586
Area of Sqaure:100.0
Perimeter of Square:40.0
Daivya Priyankkumar Shah
1BM23CS084
```

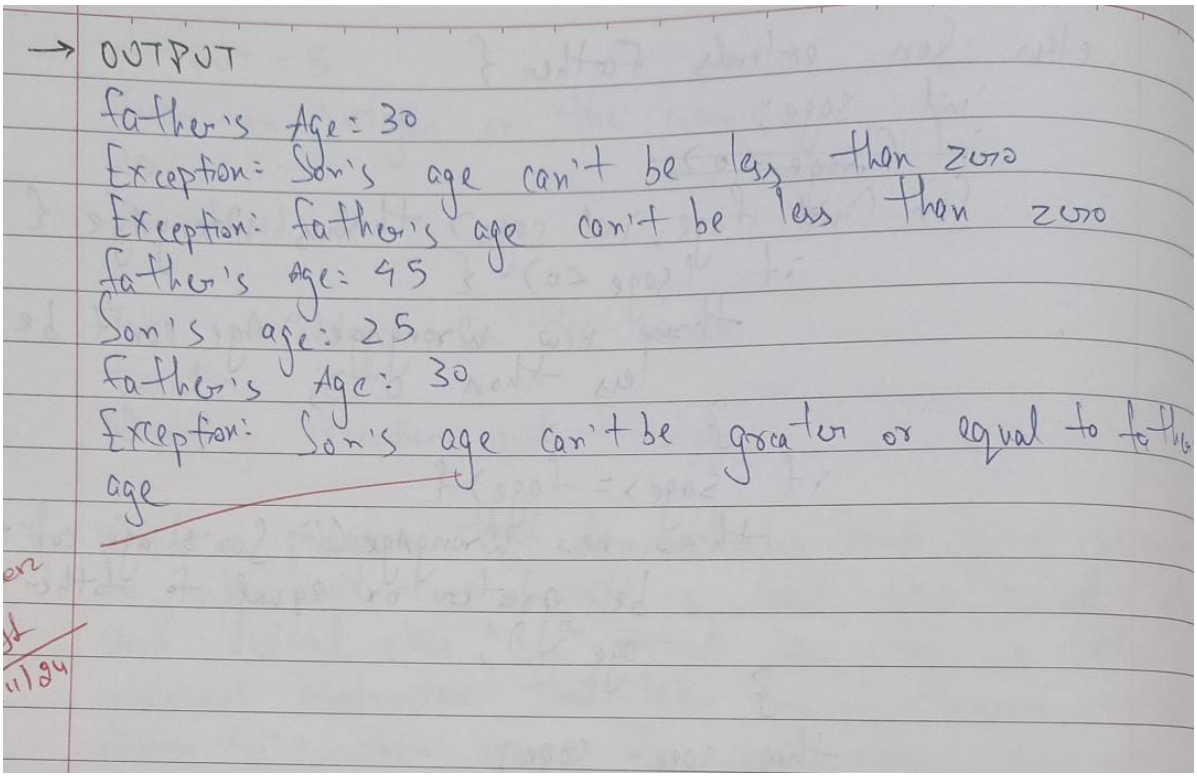
Program 8:

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>=father’s age

Algorithm:



```
class Son extends Father {
    int sage;
    if (sage < 0)
    Son (int fage, int sage) throws WrongAge {
            if (sage < 0) {
                throw new WrongAge("Age can't be
                    less than 0");
                }
            if (sage >= fage) {
                throw new WrongAge("Son's age can't
                    be greater or equal to father's
                    age");
                }
            this.sage = sage;
            System.out.println("Son's age: " + sage);
        }
}

public class ExceptionAge {
    public static void main (String[] args) {
        try {
            Son s = new Son (30, -2);
            Son ss = new Son (-8, 3);
            Son son = new Son (45, 25);
            Son newson = new Son (30, 35);
        }
        catch (WrongAge e) {
            System.out.println("Exception:" + e.getMessa
                                                    );
        }
    }
}
```



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 negative.");
        }
        this.age = age;
    }
}

class Son extends Father {
    public int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAgeException {
        super(fatherAge);
```

```
        if (sonAge < 0) {
            throw new WrongAgeException("Son's age cannot be negative.");
        }
        if (sonAge >= fatherAge) {
            throw new WrongAgeException("Son's age cannot be greater than or equal to Father's age.");
        }
        this.sonAge = sonAge;
    }
}

public class Familytree {
    public static void main(String[] args) {
        try {
            int fatherAge1 = 40;
            int sonAge1 = 15;
            Son son1 = new Son(fatherAge1, sonAge1);
            System.out.println("Father's age: " + son1.age);
            System.out.println("Son's age: " + son1.sonAge);

            int fatherAge2 = 50;
            int sonAge2 = 60;
            Son son2 = new Son(fatherAge2, sonAge2);
            System.out.println("Father's age: " + son2.age);
            System.out.println("Son's age: " + son2.sonAge);

        } catch (WrongAgeException e) {
            System.out.println("Error: " + e.getMessage());
        }
        System.out.println("Daivya Priyankkumar Shah");
        System.out.println("1BM23CS084");
    }
}
```

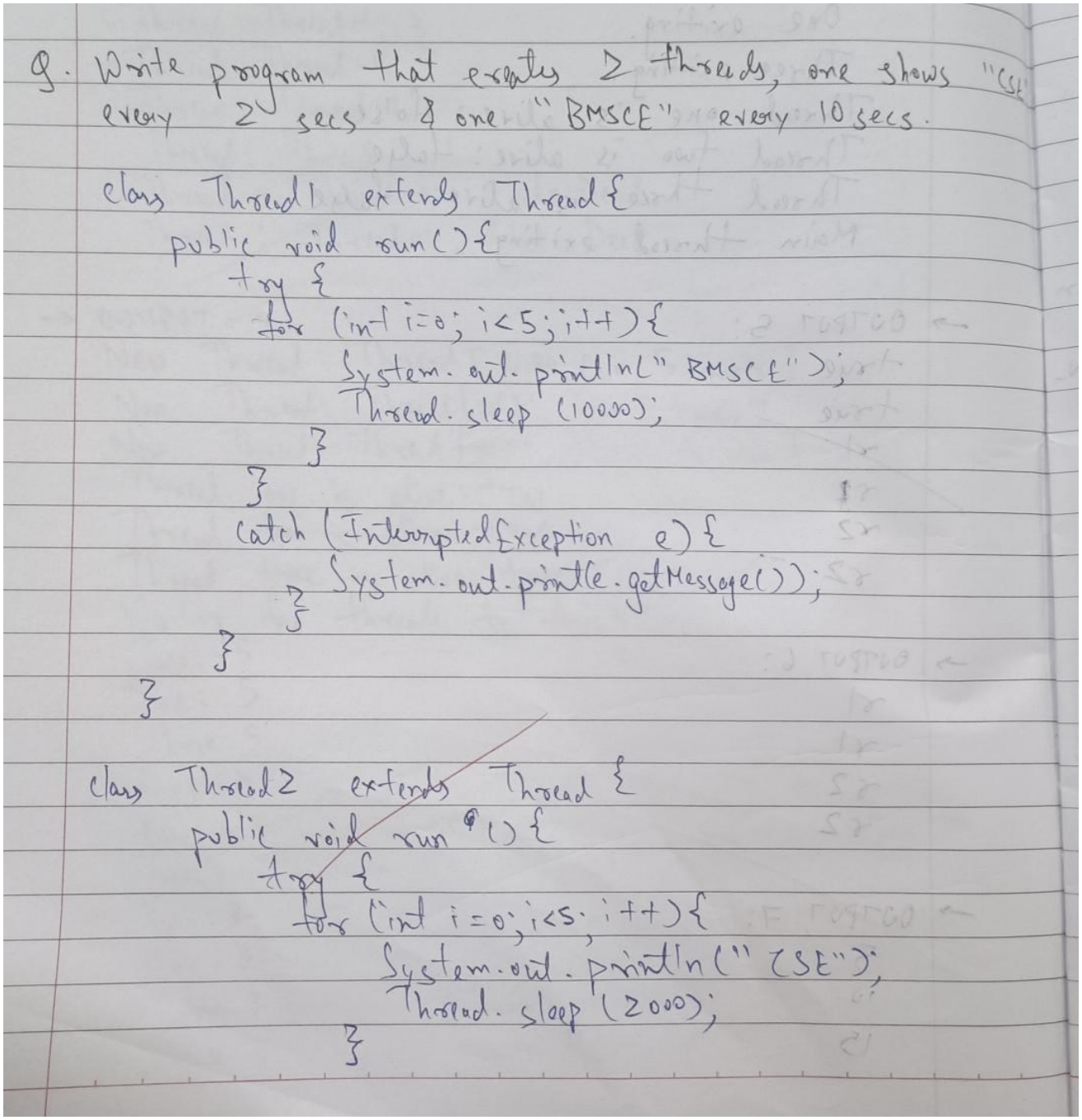
OUTPUT:

```
Father's age: 40
Son's age: 15
Error: Son's age cannot be greater than or equal to Father's age.
Daivya Priyankkumar Shah
1BM23CS084
```

Program 9:

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.

Algorithm:



```
Q. Write program that creates 2 threads, one shows "CSE"
every 2 secs & one "BMSCE" every 10 secs.

class Thread1 extends Thread{
    public void run(){
        try {
            for (int i=0; i<5; i++){
                System.out.println(" BMSCE");
                Thread.sleep (10000);
            }
        }
        catch (InterruptedException e){
            System.out.println(e.getMessage());
        }
    }
}

class Thread2 extends Thread {
    public void run(){
        try {
            for (int i=0; i<5; i++){
                System.out.println(" CSE");
                Thread.sleep (2000);
            }
```

```
        }
        catch (InterruptedException e){
            } System.out.println(e.getMessage());
        }
    }

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

→ OUTPUT:

BMSCE
CSE
CSE
CSE
CSE
CSE
BMSCE
BMSCE
BMSCE
BMSCE

03/12/24

Code:

```
class BMSCollegeThread extends Thread {
    public void run() {
        try {
            for(int i=0;i<5;i++) {
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000);
            }
        } catch (InterruptedException e) {
            System.out.println("BMSCollegeThread interrupted.");
        }
    }
}

class CSEThread extends Thread {
    public void run() {
        try {
            for(int i=0;i<5;i++) {
                System.out.println("CSE");
                Thread.sleep(2000);
            }
        } catch (InterruptedException e) {
            System.out.println("CSEThread interrupted.");
        }
    }
}

public class ThreadDisplay {
    public static void main(String[] args) {
        BMSCollegeThread bmsThread = new BMSCollegeThread();
        CSEThread cseThread = new CSEThread();

        bmsThread.start();
        cseThread.start();
        System.out.println("Daivya Priyankkumar Shah");
        System.out.println("1BM23CS084");
    }
}
```

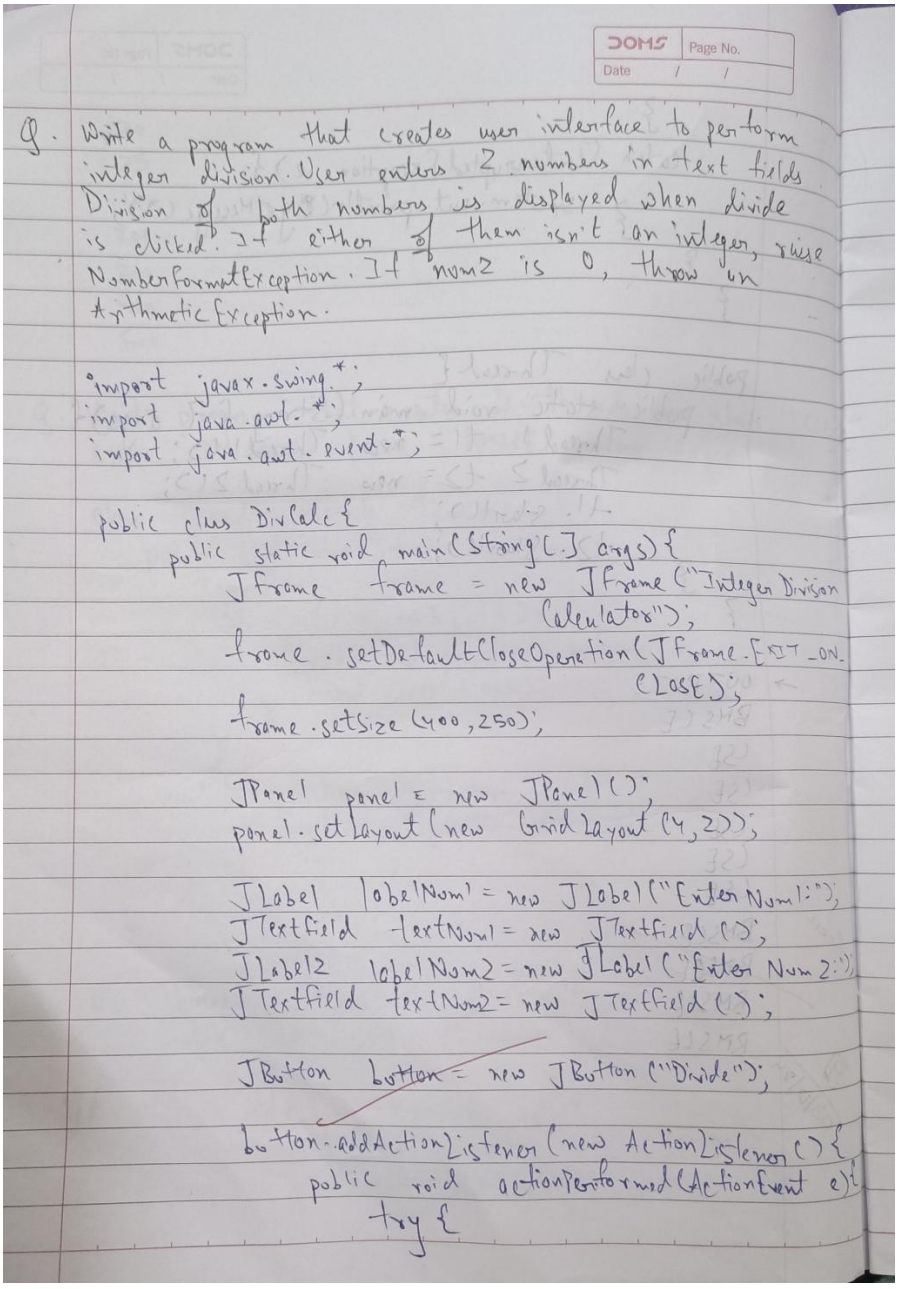
OUTPUT:

```
Daivya Priyankkumar Shah
1BM23CS084
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
PS C:\Users\daivy\IdeaProjects\JavaPractice\src>
```

Program 10:

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.

Algorithm:



Q. Write a program that creates user interface to perform integer division. User enters 2 numbers in text fields Division of both numbers is displayed when divide is clicked. If either of them isn't an integer, raise NumberFormatException. If num2 is 0, throw an Arithmetic Exception.

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

public class DivCalc{
    public static void main(String[] args){
        JFrame frame = new JFrame("Integer Division
                                   Calculator");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_
                                   CLOSE);
        frame.setSize(400,250);

        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(4,2));

        JLabel labelNum1 = new JLabel("Enter Num1:");
        JTextField textNum1 = new JTextField();
        JLabel2 labelNum2 = new JLabel("Enter Num 2:");
        JTextField textNum2 = new JTextField();

        JButton button = new JButton("Divide");

        button.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){
                try{
```

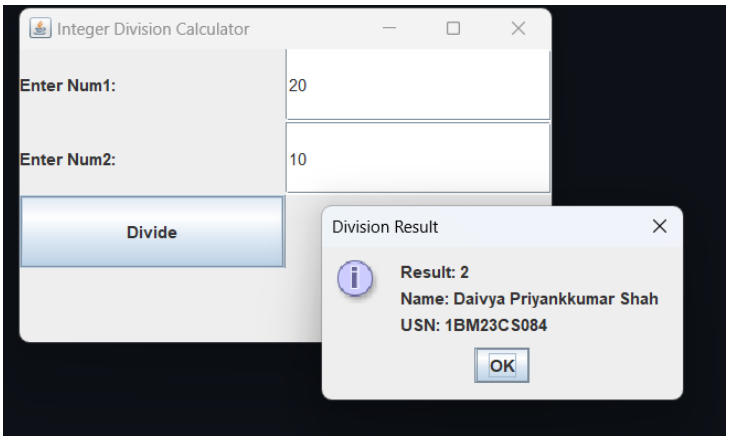
```
int num1 = Integer.parseInt(textNum1.getText());
int num2 = Integer.parseInt(textNum2.getText());
if (num2 == 0) {
    throw new ArithmeticException("Can't divide by 0");
}
int result = num1 / num2;
JOptionPane.showMessageDialog(frame, "Result: " + result
        "Division Result", JOptionPane.INFORMATION_MESSAGE);
}
catch(NumberFormatException e) {
    JOptionPane.showMessageDialog(frame, "Please Enter valid
        integers", "Input Error", JOptionPane.ERROR_
                                    MESSAGE);
}
catch(ArithmeticException e){
    JOptionPane.showMessageDialog(frame, e.getMessage()
        "Math Error", JOptionPane.ERROR_MESSAGE)
}
}
});
panel.add(labelNum1);
panel.add(labelNum2);
panel.add(textNum1);
panel.add(textNum2);
panel.add(buttonDivide);

frame.add(panel);
frame.setVisible(true);
}
}
```

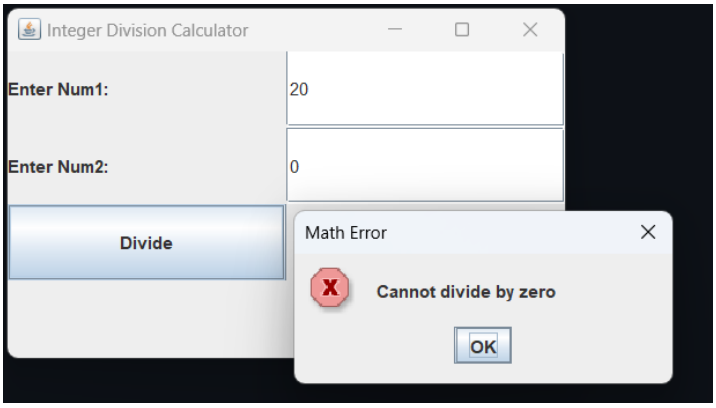
Code:

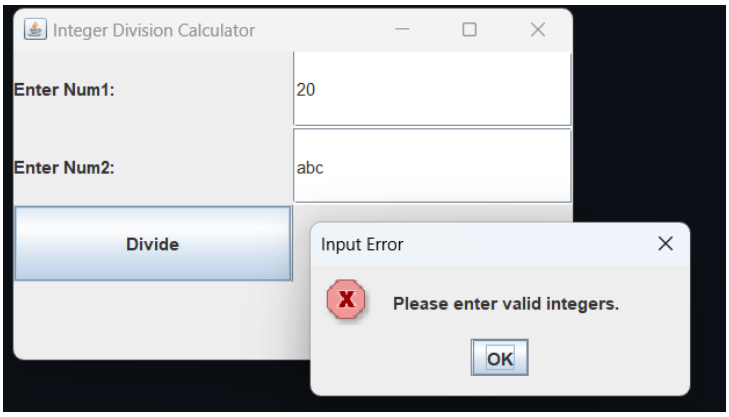
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class DivCalc {
   public static void main(String[] args) {
      JFrame frame = new JFrame("Integer Division Calculator");
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      frame.setSize(400, 250);

      JPanel panel = new JPanel();
      panel.setLayout(new GridLayout(4, 2));

      JLabel labelNum1 = new JLabel("Enter Num1: ");
      JTextField textNum1 = new JTextField();
      JLabel labelNum2 = new JLabel("Enter Num2: ");
      JTextField textNum2 = new JTextField();

      JButton buttonDivide = new JButton("Divide");

      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;
               JOptionPane.showMessageDialog(frame, "Result: " + result + "\nName: Daivya
Priyankkumar Shah\nUSN: 1BM23CS084", "Division Result",
JOptionPane.INFORMATION_MESSAGE);
            } catch (NumberFormatException ex) {
               JOptionPane.showMessageDialog(frame, "Please enter valid integers.", "Input Error",
JOptionPane.ERROR_MESSAGE);
            } catch (ArithmeticException ex) {
               JOptionPane.showMessageDialog(frame, ex.getMessage(), "Math Error",
JOptionPane.ERROR_MESSAGE);
```

```
            }
        }
    });

    panel.add(labelNum1);
    panel.add(textNum1);
    panel.add(labelNum2);
    panel.add(textNum2);
    panel.add(buttonDivide);

    frame.add(panel);
    frame.setVisible(true);
  }
}
```

OUTPUT:






Integer Division Calculator
Enter Num1:
20
Enter Num2:
abc
Divide
Input Error
Please enter valid integers.
OK