N PRIYANKA
1BM22CS167

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.

```
import java.util.Scanner;
class First{
	public static void main(String args[]){
		int a,b,c;
		double r1,r2,d;
		System.out.println("enter the coefficients of quadratic equation");
		Scanner s1= new Scanner(System.in);
		a=s1.nextInt();
		b=s1.nextInt();
		c=s1.nextInt();
		if(a==0){
			System.out.println("coefficients are invalid");
		}
		else{
			d=b*b-(4*a*c);
			if(d>0){
				System.out.println("it has real and distinct roots");
				r1=((-b+Math.sqrt(d)))/(2*a);
				r2=((-b-Math.sqrt(d)))/(2*a);
				System.out.println("the roots are "+r1+" and "+r2);
			}
			else if(d==0){
				System.out.println("it has real and equal roots");
				r1=(-b)/(2*a);
				System.out.println("the roots are "+r1+" and "+r1);
			}
			else{
				System.out.println("it has no real roots");
			}
		}
	}
}
```

Output:

```
priyanka@PRIYANKAs-MacBook-Air desktop % javac First.java
priyanka@PRIYANKAs-MacBook-Air desktop % java First
enter the coefficients of quadratic equation
1
4
1
it has real and distinct roots
the roots are -0.2679491924311228 and -3.732050807568877
priyanka@PRIYANKAs-MacBook-Air desktop % javac First.java
priyanka@PRIYANKAs-MacBook-Air desktop % java First
enter the coefficients of quadratic equation
114
1
1
it has no real roots
priyanka@PRIYANKAs-MacBook-Air desktop % javac First.java
priyanka@PRIYANKAs-MacBook-Air desktop % java First
enter the coefficients of quadratic equation
1
2
1
it has real and equal roots
the roots are -1.0 and -1.0
priyanka@PRIYANKAs-MacBook-Air desktop %
```

N PRIYANKA
1BM22CS167

18/12/23

LAB-1.

→ Develop a java program that prints all real solutions to the quadratic equation $ax^2 + bx + c = 0$. Read the co-efficients and use the quadratic formula. If the discriminate $b^2 - 4ac$ is negative, display a message stating that there are no real solutions.

```
import java.util.Scanner;
class First
{
    public static void main(String args[])
    {
        int a, b, c, d;
        float r1, r2;
        System.out.println("Enter the co-efficients
                of quadratic equation");

        Scanner s1 = new Scanner(System.in);
        a = s1.nextInt();
        b = s1.nextInt();
        c = s2.nextInt();

        if (a==0)
        {
            System.out.println("co-efficients ax
                invalid");
        }
        else
        {
            d = (b*b) - (4*a*c);
            if (d>0)
            {
            System.out.println("It has real
                and distinct roots");
```

N PRIYANKA
1BM22CS167

```
        r1 = (-b + Sqrt(d)) / (2*a);
        r2 = (-b - Sqrt(d)) / (2*a);
        System.out.println("The roots are" + r1 +
                           "and" + r2);
    }
    else if (d == 0)
    {
        System.out.println("It has real and
                           equal roots");
        r1 = (-b)/(2*a);
        System.out.println("The roots are " + r1 +
                           "and" + r2);
    }
    else {
        System.out.println("there are no real
                           solutions");
    }
  }
}
```

o/p:

```
enter the co-efficients of quadratic equation
1
4
1
It has real and distinct roots
The roots are -0.26794 and -3.7320

enter the co-efficients of quadratic equation
1
1
1
It has no real roots
```

N PRIYANKA
1BM22CS167

Enter the coefficients of quadratic equation
1
2
1
It has real and equal roots
the roots are -1.0 and -1.0.

N PRIYANKA
1BM22CS167

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.

```
import java.util.Scanner;

class Subject {
    int subjectMarks;
    int credits;
    int grade;
}
class Student {
    Subject subject[];
    String name;
    String USN;
    double SGPA;
    Scanner sc;

    public Student() {
        subject = new Subject[8];
        for(int i = 0; i < 8; i++) {
            subject[i] = new Subject();
        }
    }

    void getStudentDetails() {
        sc = new Scanner(System.in);
        System.out.print("Enter name : ");
        this.name = sc.next();

        System.out.print("Enter your USN : ");
        this.USN = sc.next();
    }

    void getMarks() {
        sc = new Scanner(System.in);
        for(int i = 0; i < 8; i++) {
            System.out.print("Enter " + (i+1) + " subject marks : ");
            subject[i].subjectMarks = sc.nextInt();

            System.out.print("Enter number of credits : ");
            subject[i].credits = sc.nextInt();

            subject[i].grade = (subject[i].subjectMarks/10) + 1;

            if(subject[i].grade == 11) subject[i].grade = 10;

            if(subject[i].grade <= 4) subject[i].grade = 0;
        }
    }
```

N PRIYANKA
1BM22CS167

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

```
import java.util.*;

class subject
{
    int subjectMarks;
    int credits;
    int grade;
}

Class Student {
    Subject subject[];
    String name;
    String USN;
    double sgpa;
    Scanner sc;

    Student()
    {
        int i;
        subject = new Subject[8];
        sc = new Scanner(System.in);
        for(i=0; i<8; i++)
            subject[i] = new Subject();
    }

    void getStudentdetails()
    {
```

```
    void computeSGPA() {
        int totalCredits = 0;
        for(int i = 0; i < 8; i++) {
            totalCredits += subject[i].credits;
        }

        int totalGradeAndCredit = 0;
        for(int i = 0; i < 8; i++) {
            totalGradeAndCredit += (subject[i].credits * subject[i].grade);
        }

        this.SGPA = ((float)totalGradeAndCredit/totalCredits);

        System.out.println("SGPA of the student is : " + this.SGPA);
    }
}


class studentMain {
    public static void main(String[] args) {
        Student s1 = new Student();
        s1.getStudentDetails();
        s1.getMarks();
        s1.computeSGPA();
    }
}
```

Output:

N PRIYANKA
1BM22CS167

```
        sgpa = (double) effectiveScore / (double) totalCredits;
    }
}

class Main{
    public static void main (String args[])
    {
        Student S1 = new Student();
        S1.getStudentDetails();
        S1.getMarks();
        S1.computeSGPA();
        System.out.println("Name:" + S1.name);
        System.out.println("USN:" + S1.USN);
        System.out.println("SGPA=" + S1.sgpa);
    }
}
```

O/P:

```
Enter your name:
Priyanka
Enter USN:
1BM22CS167
Enter the marks for sub1:
95
Enter the credits
10
Enter the marks for sub2:
98
Enter the credits
10
Enter the marks for sub3:
80
Enter the credits
9
Enter the marks for sub4:
80
```

N PRIYANKA
1BM22CS167

```
System.out.println("Enter your name:");
this.name = sc.next();
System.out.println("Enter your USN:");
this.USN = sc.next();
}

void getMarks()
{
    for(int i = 0; i<8; i++)
    {
        System.out.println("Enter marks for
    subject " + (i+1) + ":");
    subject[i].subjectMarks =
    sc.nextInt();
    System.out.println("Enter the credits for
    subject " + (i+1) + ":");
        subject[i].credits = sc.nextInt();
        subject[i].grade = subject[i].subject
    Marks/10 + 1;
    if(subject[i].grade == 11)
        subject[i].grade = 10;
    if(subject[i].grade <= 4)
        subject[i].grade = 0;
    }
}

void computeSGPA()
{
    int effectiveScore = 0;
    int totalCredits = 0;
    for(int i = 0; i < 8; i++)
    {
        effectiveScore += (subject[i].grade *
    subject[i].credits);
        total credits += subject[i].credits;
    }
```

off.

Enter the credits
3
Enter the marks for sub5
90
Enter the credits
10
Enter the marks for sub6
90
Enter the credits
10
Enter the marks for sub7
97
Enter the credits
10
Enter the marks for sub8
99
Enter the credits
10
Name : Priyanka
USN = 1BM22CS162
SGPA : 9.769230

11/24

N PRIYANKA
1BM22CS167

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

```
Import java.util.Scanner;

class Books {
{
    String name, author;
    int price, numPages;

    Books (String name, String author, int price,
int numPages)
    {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }
    public String toString()
    {
        String name, author price, numPages;
        name = "Book names:" + this.name + "\n";
        author = "Author name :"+ this.author +
        "\n";
        price = "Price : " + this.numPages + "\n";
        return name + author + price + numPages;
    }
}
```

N PRIYANKA
1BM22CS167

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.

```
import java.util.Scanner;

class Books
{
	String name;
	String author;
	int price;
	int numPages;

	Books(String name,String author,int price,int numPages)
	{
		this.name=name;
		this.author=author;
		this.price=price;
		this.numPages=numPages;
	}

	public String toString()
	{
		String name,author,price,numPages;
		name="Book name:" +this.name+ "\n";
		author="Author name:" +this.author+ "\n";
		price="Price:" +this.price+ "\n";
		numPages="Number of pages:" +this.numPages+ "\n";
		return name+author+price+numPages;
	}
}

public class Mainbook
{
	public static void main(String args[])
	{
		Scanner s=new Scanner(System.in);
		int n;
		int i;
		String name;
		String author;
		int price;
		int numPages;

		System.out.println("Enter the number of books:");
		n=s.nextInt();

		Books b[];
		b=new Books[n];
```

```
            for(i=0;i<n;i++)
            {
                System.out.println("Enter the details of book" + (i+1) + ":");
                System.out.println("Enter the name of the book:");
                name=s.next();
                System.out.println("Enter the author name:");
                author=s.next();
                System.out.println("Enter the price:");
                price=s.nextInt();
                System.out.println("Enter the number of pages:");
                numPages=s.nextInt();

                b[i]=new Books(name,author,price,numPages);
            }

            System.out.println("Book Details:");
            for(i=0;i<n;i++)
            {
                System.out.println(b[i]);
            }
        }
}
```

Output:

```
priyanka@PRIYANKAs-MacBook-Air desktop % javac Mainbook.java
priyanka@PRIYANKAs-MacBook-Air desktop % java Mainbook
Enter the number of books:
2
Enter the details of book1:
Enter the name of the book:
ends_with_us
Enter the author name:
collen_hoover
Enter the price:
345
Enter the number of pages:
567
Enter the details of book2:
Enter the name of the book:
starts_with_us
Enter the author name:
collen_hoover
Enter the price:
345
Enter the number of pages:
678
Book Details:
Book name:ends_with_us
Author name:collen_hoover
Price:345
Number of pages:567

Book name:starts_with_us
Author name:collen_hoover
Price:345
Number of pages:678
```

N PRIYANKA
1BM22CS167

(3) ~~create~~

(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 bo. objects.

```
Import java.util.Scanner;

class Books{
{
    String name, author;
    int price, numPages;

    Books (String name, String author, int price,
    int numPages)
    {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }
    public String toString()
    {
        String name, author price, numPages;
        name = "Book names:" + this.name + "\n";
        author = "Author name :"+ this.author +
        "\n";
        price = "Price : " + this.numPages + "\n";
        return name + author + price + numPages;
    }
}
```

N PRIYANKA
1BM22CS167

```
class BookMain{
    public static void main (String [] args)
    {
        Scanner sc = new Scanner (System.in);
        int n;
        String name, author;
        int price, numPages;
        System.out.println ("Enter the number
    of books:");
        n = sc.nextInt ();
        Books b[] = new Books [n];
        System.out.println ("Enter Name,
    author, price and number of pages:");
        for (int i = 0; i<n; i++)
        {
            name = sc.next();
            author = sc.next();
            price = sc.next();
            numPages = sc.nextInt();
            b[i] = new Books (name, author,
        price, numPages);
        }
        System.out.println("Book details:");
        for (int i =0; i<n; i++)
        {
            System.out.println (b[i].toString ());
        }
    }
}
```

N PRIYANKA
1BM22CS167

output:

```
Entr the number of books:
2
Entr Name, author, price and number of pages:
ends with us
Colleen hoover
345
567
Starts with us
Colleen hover
345
678
Book details:
Book name: ends with us
Author name: Colleen hoover
Price : 345
Number of pages : 567

Book name : starts with us
Author name : Colleen hoover
Price : 345
Number of pages : 678.
```

N PRIYANKA
1BM22CS167

Program 4:

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

```
import java.util.Scanner;

class InputScanner {

    protected Scanner scanner;

    public InputScanner() {
        scanner = new Scanner(System.in);
    }

    public int getInput(String message) {
        System.out.println(message);
        return scanner.nextInt();
    }
}

abstract class Shape {

    protected int a, b;
    protected InputScanner inputScanner;

    public Shape() {
        inputScanner = new InputScanner();
    }

    abstract public void printArea();
}

class Rectangle extends Shape {

    public Rectangle() {
        super();
    }

    public void printArea() {
        a = inputScanner.getInput("Enter the length:");
        b = inputScanner.getInput("Enter the breadth:");
        int area = a * b;
        System.out.println("Area of the Rectangle: " + area);
    }
}

class Triangle extends Shape {
```

```
    public Triangle() {
        super();
    }

    public void printArea() {
        a = inputScanner.getInput("Enter the side1:");
        b = inputScanner.getInput("Enter the side2:");
        double area = 0.5 * a * b;
        System.out.println("Area of the Triangle: " + area);
    }
}

class Circle extends Shape {

    public Circle() {
        super();
    }

    public void printArea() {
        a = inputScanner.getInput("Enter the radius:");
        double area = 3.14 * a * a;
        System.out.println("Area of the Circle: " + area);
    }
}

public class MainShape {

    public static void main(String[] args) {
        Rectangle r = new Rectangle();
        Triangle t = new Triangle();
        Circle c = new Circle();

        r.printArea();
        t.printArea();
        c.printArea();
    }
}
```

Output:

```
priyanka@PRIYANKAs-MacBook-Air desktop % javac MainShape.java
priyanka@PRIYANKAs-MacBook-Air desktop % java MainShape
Enter the length:
12
Enter the breadth:
3
Area of the Rectangle: 36
Enter the side1:
23
Enter the side2:
45
Area of the Triangle: 517.5
Enter the radius:
67
Area of the Circle: 14095.46
priyanka@PRIYANKAs-MacBook-Air desktop %
```

N PRIYANKA
1BM22CS167

Develop a java program to create an abstract class named Shape that contains two integers and an empty method 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 area of the given shape.

```
import java.util.Scanner;

class inputScanner
{
    protected Scanner s;
    public inputScanner()
    {
        s = new Scanner (System.in);
    }
    public int getInput( String message)
    {
        System.out.println(message);
        return Scanner.nextInt();
    }
}

abstract class Shape extends inputScanner
{
    protected int a,b;
    public Shape()
    {
        super();
    }
```

N PRIYANKA
1BM22CS167

```
abstract public void printArea();
}

class Rectangle extends Shape
{
    protected int a, b;
    public Rectangle()
    {
        super();
    }
    public void printArea()
    {
        a = getInput("Enter length");
        b = getInput("Enter the breadth");
        int area = a*b;
        System.out.println("Area of the Rectangle:"
                + area);
    }
}

class Triangle extends Shape
{
    protected int a, b;
    public Triangle()
    {
        super();
    }

    public void printArea()
    {
        a = getInput("Enter the side1:");
        b = getInput("Enter the side2:");
        double area = 0.5*a*b;
        System.out.println("Area of triangle =" + area);
    }
}
```

N PRIYANKA
1BM22CS167

```
Class circle extends Shape
{
    protected int a;
    public Circle()
    {
        super();
    }
    public void printArea()
    {
        a = getInput("Enter the radius");
        double area = 3.14 * a * a;
        System.out.println("Area of the circle:" +
        area);
    }
}
public class MainShape
{
    public static void main(String[] args)
    {
        Rectangle r = new Rectangle();
        Triangle t = new Triangle();
        Circle c = new Circle();

        r.printArea();
        t.printArea();
        c.printArea();
    }
}
```

N PRIYANKA
1BM22CS167

Output:

```
Enter the length:
12
Enter the breadth:
3
Area of Rectangle :36
Enter the base:
23
Enter the height:
45
Area of the triangle : 517.5
Enter the radius:
67
Area of the circle : 14095.46
```

N PRIYANKA
1BM22CS167

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.

```
import java.util.Scanner;
class account
{
	String name;
	int accno;
	String type;
	double balance;

	account(String name,int accno,String type,double balance)
	{
		this.name=name;
		this.accno=accno;
		this.type=type;
		this.balance=balance;
	}
	void deposit(double amount)
	{
		balance+=amount;
	}
	void withdraw(double amount)
	{
		if((balance-amount)>=0)
		{
			balance-=amount;
		}
		else
		{
			System.out.println("insufficient balance,cant withdraw");
		}
	}

	void display()
	{

System.out.println("name:"+name+"accno:"+accno+"type:"+type+"balance:"+balance);
```

```
	}
}
class savAcct extends account
{

	private static double rate=5;
	savAcct(String name,int accno,double balance)
	{
		super(name,accno,"savings",balance);

	}

	void interest()
	{
		balance+=balance*(rate)/100;
		System.out.println("balance:"+balance);
	}


}
class curAcct extends account
{

	private double minBal=500;
	private double serviceCharges=50;

	curAcct(String name,int accno,double balance)
	{
		super(name,accno,"current",balance);

	}


	void checkmin()
	{

		if(balance<minBal)
		{
			System.out.println("balance is less than min balance,service charges
imposed:"+serviceCharges);
			balance-=serviceCharges;
			System.out.println("balance is:"+balance);
		}

	}

}
class accountMain
{
	public static void main(String a[])
	{
```

```
Scanner s=new Scanner(System.in);
System.out.println("enter the name :");
String name=s.next();
System.out.println("enter the type(current/savings):");
String type=s.next();
System.out.println("enter the account number:");
int accno=s.nextInt();
System.out.println("enter the intial balance:");
double balance=s.nextDouble();
int ch;
double amount1,amount2;
account acc=new account(name,accno,type,balance);
savAcct sa=new savAcct(name,accno,balance);
curAcct ca=new curAcct(name,accno,balance);
while(true)
{
	if(acc.type.equals("savings"))
	{
		System.out.println("\nMenu\n1.deposit 2.withdraw 3.compute interest
4.display");
		System.out.println("enter the choice:");
		ch=s.nextInt();
		switch(ch)
		{
			case 1:System.out.println("enter the amount:");
				amount1=s.nextInt();
				sa.deposit(amount1);
				break;
			case 2:System.out.println("enter the amount:");
				amount2=s.nextInt();
				sa.withdraw(amount2);
				break;
			case 3:sa.interest();
				break;
			case 4:sa.display();
				break;
			case 5:System.exit(0);
			default:System.out.println("invalid input");
				break;
		}
	}
	else
	{
		System.out.println("\nMenu\n1.deposit 2.withdraw  3.display");
		System.out.println("enter the choice:");
		ch=s.nextInt();
		switch(ch)
		{
			case 1:System.out.println("enter the amount:");
				amount1=s.nextInt();
				ca.deposit(amount1);
```

```
                                        break;
                                case 2:System.out.println("enter the amount:");
                                        amount2=s.nextInt();
                                        ca.withdraw(amount2);
                                        ca.checkmin();
                                        break;

                                case 3:ca.display();
                                        break;
                                case 4:System.exit(0);
                                default:System.out.println("invalid input");
                                        break;
                        }
                }
        }

    }
}
```

Output:

```
C:\Users\EXAM\Desktop\1BM22CS167>javac accountMain.java

C:\Users\EXAM\Desktop\1BM22CS167>java accountMain
enter the name :
priyanka
enter the type(current/savings):
savings
enter the account number:
167
enter the intial balance:
1000

Menu
1.deposit 2.withdraw 3.compute interest 4.display
enter the choice:
2
enter the amount:
200

Menu
1.deposit 2.withdraw 3.compute interest 4.display
enter the choice:
3
balance:840.0

Menu
1.deposit 2.withdraw 3.compute interest 4.display
enter the choice:
4
name:priyankaaccno:167type:savingsbalance:840.0

Menu
1.deposit 2.withdraw 3.compute interest 4.display
enter the choice:
5

C:\Users\EXAM\Desktop\1BM22CS167>
```

N PRIYANKA
1BM22CS167

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 facilities. The current account provides cheque book facility with no interest. Current account holders should also maintain 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 necessary methods in order to archive the following tasks

(a) Accept deposit from customer and update the balance.
(b) Display the balance
(c) compute the deposit interest
(d) Permit withdrawal and update the balance.

N PRIYANKA
1BM22CS167

```
import java.util.Scanner;
class account
{
    String name;
    int accno;
    String type;
    double balance;
    account(String name, int accno, String type, double
            balance);
    {
        this.name = name;
        this.accno = accno;
        this.type = type;
        this.balance = balance;
    }
    void deposit(double amount)
    {
        balance += amount;
    }
    void withdraw(double amount)
    {
        if ((balance - amount)>=0)
        {
            balance -= amount;
        }
        else
        {
            System.out.println("insufficient balance,
                can't withdraw");
        }
    }
    void display()
    {
        System.out.println("name" + name +
```

N PRIYANKA
1BM22CS167

```
'accno:" + accno + " type " + type + " balance"
    + balance);
    }
}
class SavAcct extends account
{
    private static double rate = 5;
    SavAcct (string name, int accno, double
        balance)
    {
        super (name, accno, "savings", balance);
    }
    void interest()
    {
        balance += balance * (rate)/100;
        System.out.println ("balance" + balance);
    }
}
class CurrAcct extends account
{
    private double minBal = 500;
    private double service charges = 50;
    CurrAcct ( string name, int accno, double balan
    {
        super ( name, accno, "current", balance);
    }
    void checkmin()
    {
        if (balance < minBal)
        {
            System.out.println ("balance is less than
            min balance, service charges
            imposed " + service charges);
            balance -= service charges;
```

N PRIYANKA
1BM22CS167

```
            System.out.println ("balance is" + balance);
        }
    }
}
class account Main
{
    public static void main (String args[])
    {
        Scanner s = new Scanner (System.in);
        System.out.println("Enter the name:");
        String name = s.next();
        System.out.println("Enter the type (current/savings);
        String type = s.next();
        System.out.println("enter account number:");
        int accno = s.nextInt(), s.nextInt();
        System.out.println("enter initial balance:");
        double balance = s.nextDouble();
        int ch;
        double amount1, amount2;
        account acc = new account (name, accno,
        type, balance);
        SavAcct sa = new SavAcct(name, accno, balance);
        CurAcct ca = new CurAcct( name, accno, balance);
        while(true)
        {
            if (acc.type.equals("savings"))
            {
                System.out.println ("\n Menu\n 1.
                deposit \n 2. Withdraw \n 3. Compute
                interest 4. Display");
                System.out.println("Enter the
                    Choice");
                ch = s.nextInt();
                switch (ch)
```

N PRIYANKA
1BM22CS167

```
case 1:
System.out.println("Enter the amount:");
    amount1 = s.nextInt();
    sa.deposit(amount1);
    break;
case 2:
System.out.println("Enter the amount");
    amount2 = s.nextInt();
    sa.withdraw(amount2);
    break;
case 3:
    sa.interest();
    break;
case 4:
    sa.display();
    break;
case 5:
    System.exit(0);
default:
    System.out.println("Invalid input");
    break;
        }
    }
else
    {
    System.out.println("In Menu \n 1.deposit
    \n 2.withdraw 3.display");
    System.out.println("Enter the choice");
    ch = s.nextInt();
    switch(ch)
    {
    case 1:
        System.out.println("Enter the amount
        amount1 = s.nextInt();
```

N PRIYANKA
1BM22CS167

```
            ca.deposit(amount1);
            break;
        case2:
            System.out.println("Enter the amount");
            amount2 = s.nextInt();
            ca.withdraw(amount2);
            ca.checkmin();
            break;
        case3:
            ca.display();
            break;
        case4:
            System.exit(0);
        default:
            System.out.println("Invalid Input");
            break;
        }
    }
}
}
```

Output:

```
enter the name:
Priyanka
enter the type(current/savings)
savings
enter account number
167
enter initial balance
1000

Menu
1. deposit 2. withdraw 3. compute interest 4. display
5. exit;
```

N PRIYANKA
1BM22CS167

Date
Page

entr your choice
2
Entr amount:
200

Menu
1. Deposit 2. withdraw 3. compute Intrst
4. Display 5. exit
entr your choice
2
Entr the amount:
200

Menu
1. Deposit 2. withdraw 3. compute intrst
4. Display 5. exit
entr your choice
3
balance = 840.0

Menu
1. Deposit 2. withdraw 3. compute interest
4. Display 5. exit
entr your choice
4
name: Priyanka accno: 167 type: savings
balance: 840.0

Menu
1. Deposit 2. withdraw 3. compute interst
4. Display 5. exit
Entr your choice
5

N PRIYANKA
1BM22CS167

Program 6:
Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. The class internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student. 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.

```
//Students.java
package CIE;

import java.util.Scanner;

public class Student
{
   protected String usn = new String();
   protected String name = new String();
   protected int sem;

   public void inputStudentDetails()
         {
      Scanner s = new Scanner(System.in);
      System.out.print("Enter USN: ");
      usn = s.next();
      System.out.print("Enter Name: ");
      name = s.next();
      System.out.print("Enter Semester: ");
      sem = s.nextInt();
   }

   public void displayStudentDetails()
         {
      System.out.println("USN: " + usn);
      System.out.println("Name: " + name);
      System.out.println("Semester: " + sem);
   }
//Internals.java
package CIE;

import java.util.Scanner;

public class Internals extends Student
{
   protected int marks[] = new int[5];

   public Internals() {}

   public void inputCIEmarks()
         {
      Scanner s = new Scanner(System.in);
      System.out.println("Enter Internal Marks for " + name);
```

```
        for (int i = 0; i < 5; i++)
                        {
            System.out.print("Subject " + (i + 1) + " marks: ");
            marks[i] = s.nextInt();
        }
    }
}


//Externals.java
package SEE;

import CIE.Internals;

import java.util.Scanner;

public class Externals extends Internals
{
    protected int marks[];
    protected int finalMarks[];

    public Externals()
            {
        marks = new int[5];
        finalMarks = new int[5];
    }

    public void inputSEEmarks()
            {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter SEE Marks for " + name);
        for (int i = 0; i < 5; i++)
                        {
            System.out.print("Subject " + (i + 1) + " marks: ");
            marks[i] = s.nextInt();
        }
    }

    public void calculateFinalMarks()
            {
        for (int i = 0; i < 5; i++)
            finalMarks[i] = marks[i] / 2 + super.marks[i];
    }

    public void displayFinalMarks()
            {
        displayStudentDetails();
        for (int i = 0; i < 5; i++)
            System.out.println("Subject " + (i + 1) + ": " + finalMarks[i]);
    }
}
```

N PRIYANKA
1BM22CS167

```
//Main.java
import SEE.Externals;

public class Main
{
	public static void main(String args[])
	{
		int numOfStudents=2;
		Externals finalMarks[]=new Externals[numOfStudents];

		for(int i=0;i<numOfStudents;i++)
		{
			finalMarks[i]=new Externals();
			finalMarks[i].inputStudentDetails();
			System.out.println("Enter CIE marks:");
			finalMarks[i].inputCIEmarks();
			System.out.println("Enter SEE marks:");
			finalMarks[i].inputSEEmarks();
		}

		System.out.println("DIsplaying data:\n");
		for(int i=0;i<numOfStudents;i++)
		{
			finalMarks[i].calculateFinalMarks();
			finalMarks[i].displayFinalMarks();
		}
	}
}
```

Output:

```
priyanka@PRIYANKAs-MacBook-Air java % javac Main.java
priyanka@PRIYANKAs-MacBook-Air java % java Main
Enter USN: 167
Enter Name: pri
Enter Semester: 3
Enter CIE marks:
Enter Internal Marks for pri
Subject 1 marks: 37
Subject 2 marks: 36
Subject 3 marks: 32
Subject 4 marks: 38
Subject 5 marks: 40
Enter SEE marks:
Enter SEE Marks for pri
Subject 1 marks: 89
Subject 2 marks: 90
Subject 3 marks: 92
Subject 4 marks: 88
Subject 5 marks: 93
Enter USN: 234
Enter Name: sakshi
Enter Semester: 3
Enter CIE marks:
Enter Internal Marks for sakshi
Subject 1 marks: 33
Subject 2 marks: 28
Subject 3 marks: 30
Subject 4 marks: 34
Subject 5 marks: 36
Enter SEE marks:
Enter SEE Marks for sakshi
Subject 1 marks: 98
Subject 2 marks: 94
Subject 3 marks: 92
Subject 4 marks: 95
Subject 5 marks: 80
```

```
DIsplaying data:

USN: 167
Name: pri
Semester: 3
Subject 1: 81
Subject 2: 81
Subject 3: 78
Subject 4: 82
Subject 5: 86
USN: 234
Name: sakshi
Semester: 3
Subject 1: 82
Subject 2: 75
Subject 3: 76
Subject 4: 81
Subject 5: 76
priyanka@PRIYANKAs-MacBook-Air java %
```

N PRIYANKA
1BM22CS167

lab-4

Create a package CIE which has two classes Student and Internals. The class Student has members like USN, name and sem. The class Internals has an array that store the internal marks scored in 5 courses of the current semester of the student. Create another package SEE which has the class External which is derived class of the Student. This class has an array that stored these marks scored in the courses of the current semester of the student. Import the two packages in a file that declares the final marks of the students in all five courses.

//student java file.

```
package CIE;
import java.util.Scanner;
public class Student
{
    protected String USN = new String();
    protected String name = new String();
    protected int sem;
    public void InputStudentDetails()
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter USN");
        USN = s.next();
        System.out.println("Enter Name");
        name = s.next();
```

N PRIYANKA
1BM22CS167

```
        System.out.println("Enter sem");
        sem = s.nextInt();
    }
    public void displayStudentDetails()
    {
        System.out.println("USN" + usn);
        System.out.println("Name" + name);
        System.out.println("sem" + sem);
    }
}

//Internals java file

package CIE;
import java.util.Scanner;

public class internals extends student
{
    protected int marks[] = new int[5];
    public Internals(){}
    public void inputCIEmarks()
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter Internal marks
            for " + name);
        for(int i=0; i<5; i++)
        {
            System.out.println("subject" + (i+1) +
                "marks");
            marks[i] = s.nextInt();
        }
    }
}
```

N PRIYANKA
1BM22CS167

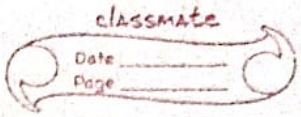
// Externals java file

```
package SEE;
import CIE.internals;
import java.util.Scanner;
public class Externals extends Internals
{
        protected int marks[];
        protected int finalmarks[];
        public Externals()
        {
                marks = new int[5];
                finalmarks = new int[5];
        }
    public void inputSEEmarks()
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter SEE marks for "+name);
        for(i=0; i<5; i++)
        {
            System.out.println("subject "+(i+1)+" marks");
            marks[i] = s.nextInt();
        }
}
public void calculateFinalMarks()
{
        for(int i=0; i<5; i++)
            finalmarks[i] = marks[i]/2 + super.marks
                [i];
}
public void displayFinalMarks()
{
        displayStudentDetails();
        for(int i=0; i<5; i++)
```

N PRIYANKA
1BM22CS167

```
        System.out.println("Subject" + (i+1) + ":" +
                                finalMarks[i]);
    }
}

// Main java file

import SEE.Externals;

public class Main
{
    public static void main(String args[])
    {
        int num of Students = 2;
        Externals finalMarks[] = new Externs
            [num of students];
        for(int i=0; i < num of Students; i++)
        {
            finalMarks[i] = new Externals();
            finalMarks[i] = inputStudentDetails();
            System.out.println("Enter CIE marks");
            finalMarks[i].inputCIEmarks();
            System.out.println("Enter SEE
                marks:");
            finalMarks[i].inputSEEmarks();
        }
        System.out.println("Displaying data:\n");
        for(int i=0; i< num of Students ;i++)
        {
            finalMarks[i].calculateFinalMarks();
            finalMarks[i].displayFinalMarks();
        }
    }
}
```

N PRIYANKA
1BM22CS167



Output:

Enter USN : 167
Enter Name : Pri
Enter sem : 3
Enter CIE marks :
Enter Internal marks for Pri
Subject 1 marks : 37
Subject 2 marks : 36
Subject 3 marks : 32
Subject 4 marks : 38
Subject 5 marks : 40
Enter SEE marks for Pri
Subject 1 marks : 89
Subject 2 marks : 90
Subject 3 marks : 92
Subject 4 marks : 88
Subject 5 marks : 93

Enter USN : 234
Enter name : Sakshi
~~Sub~~
Enter Semester : 3
Enter CIE marks :
Enter internal marks for Sakshi
Subject 1 marks : 33 :
Subject 2 marks : 28
Subject 3 marks : 30 :
Subject 4 marks : 34
Subject 5 marks : 36
Enter SEE marks
Subject 1 marks : ~~33~~ 98
Subject 2 marks : 94
Subject 3 marks : 92

N PRIYANKA
1BM22CS167

Subject 4 marks : 95
subject 5 marks : 80

Displaying data:
USN:1b1
name: Pri
Semstr 3
Subject 1 = 81
Subject 2 = 81
Subject 3 = 78
Subject 4 = 82
Subject 5 = 86

USN :234
Name : Sakshi
Semester : 3
Subject 1 : 82
Subject 2 : 75
Subject 3: 76
Subject 4: 81
Subject 5: 76

29/1/24

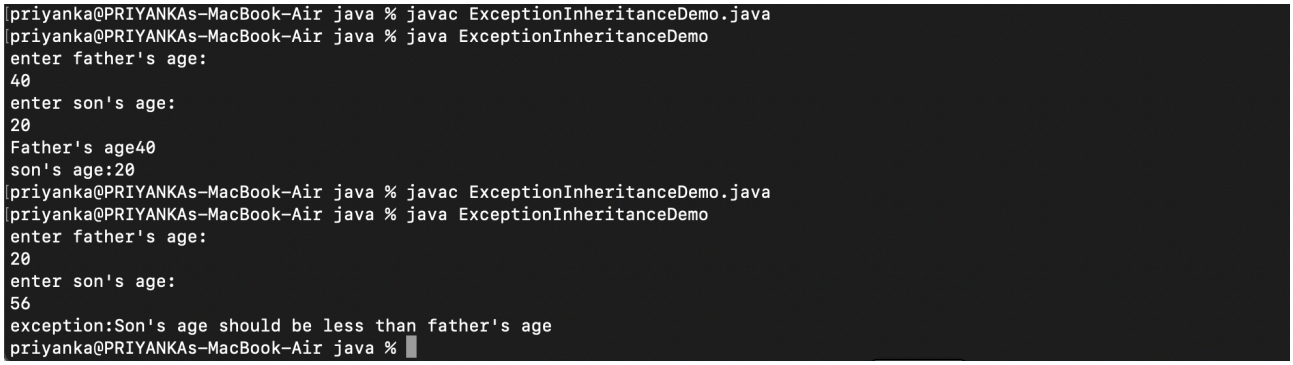
N PRIYANKA
1BM22CS167

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

```
import java.util.Scanner;

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

class Father{
private int fatherAge;

public Father(int age)throws WrongAgeException{
if(age<0){
throw new WrongAgeException("Age cannot be negetive");
}
this.fatherAge=age;
}
}
class son extends Father{
private int sonAge;
public son(int fatherAge,int sonAge) throws WrongAgeException{
super(fatherAge);
if(sonAge>=fatherAge){
throw new WrongAgeException("Son's age should be less than father's age");
}
this.sonAge=sonAge;
System.out.println("Father's age"+fatherAge);
System.out.println("son's age:"+sonAge);
}
}
public class ExceptionInheritanceDemo{
public static void main (String args[]){
Scanner scanner =new Scanner(System.in);
try{
System.out.println("enter father's age:");
int fatherAge=scanner.nextInt();
Father father=new Father(fatherAge);

System.out.println("enter son's age:");
int sonAge=scanner.nextInt();
son son= new son(fatherAge,sonAge);
}catch(WrongAgeException e){
System.out.println("exception:"+e.getMessage());
}
```

```
}
}
```

Output:


priyanka@PRIYANKAs-MacBook-Air java % javac ExceptionInheritanceDemo.java
priyanka@PRIYANKAs-MacBook-Air java % java ExceptionInheritanceDemo
enter father's age:
40
enter son's age:
20
Father's age40
son's age:20
priyanka@PRIYANKAs-MacBook-Air java % javac ExceptionInheritanceDemo.java
priyanka@PRIYANKAs-MacBook-Air java % java ExceptionInheritanceDemo
enter father's age:
20
enter son's age:
56
exception:Son's age should be less than father's age
priyanka@PRIYANKAs-MacBook-Air java %

N PRIYANKA
1BM22CS167

write a program that demonstrates handling of exceptions in inheritence tree. Create a base class called Father and derived class "son" which extends the base class. In father class implement a constructor which takes the age and throws the exception WrongAge() when the input age<0. In son class, implement a constructor

```
import java.util.Scanner;

Class WrongAageException extends Exception
{
    public WrongAgeException(String message)
    {
        super(message);
    }
}

Class Father {
    private int fatherAage;
    public Father(int age) throws WrongAageException
    {
        if (age<0)
        {
            throw new WrongAageException("Aage
            cannot be negative");
        }
        this.fatherAge = age;
    }
}
```

N PRIYANKA
1BM22CS167

```
class Son extends Father
{
    private int SonAage;
    public Son(int fatherage, int SonAage)
     throws WrongAageException
    {
        super(fatherAage);
        if (SonAage >= fatherAage)
        {
            throw new WrongAageException
            ("son's age should be less than fath
             age");
        }
        this.SonAge = SonAge;
        System.out.println("Father's age": +
                                fatherAage);
        System.out.println("son's age": +
                                SonAage);
    }
}
public class ExceptionInheritanceDemo
{
    public static void main(string arg[])
    {
        Scanner scanner = new Scanner(System.in)
        try
        {
            System.out.println("Enter Father's
                                Age");
            int fatherAge = Scanner.nextInt();
            Father father = new Father(fatherAage

            System.out.println("Enter son's age");
            int SonAage = Scanner.nextInt();
```

N PRIYANKA
1BM22CS167

```
        Son son = new Son (fatherAge, sonAge);
    }
    catch (WrongException e)
    {
        System.out.println ("Exception:" +
                            e.getMessage());
    }
}
}
```

Output:

```
Enter Father's age : 40
Enter son's age : 20
Father's age : 40
Son's age : 20
Enter another son's age : 45
Exception: Son's age cannot be greater than
           or equal to father's age.
```

29/11/24

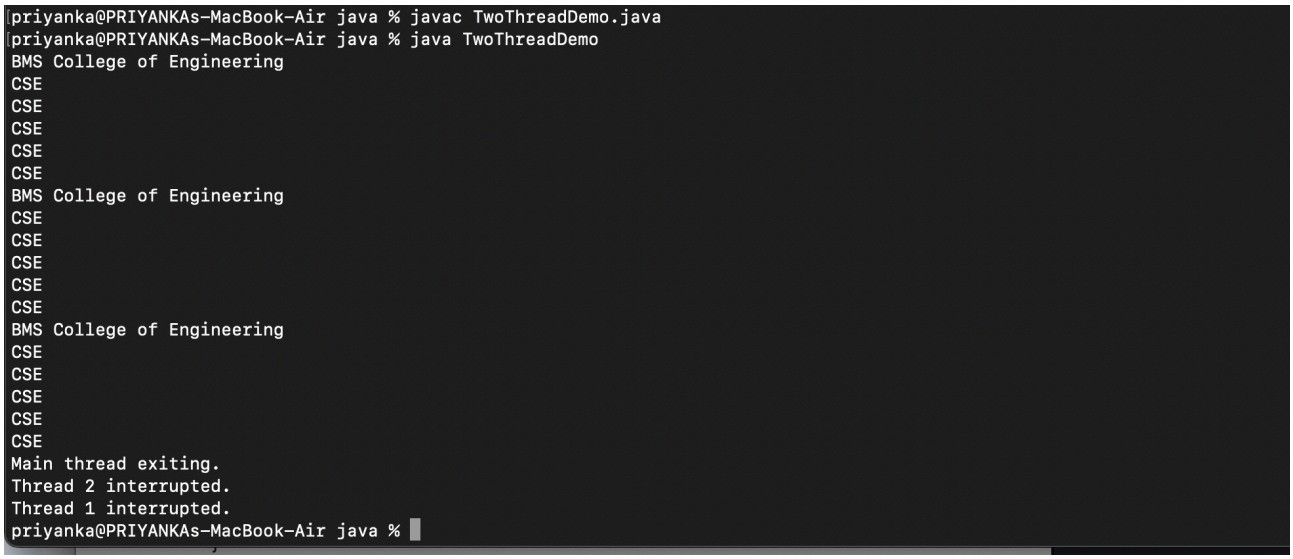
N PRIYANKA
1BM22CS167

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

```
class DisplayMessageThread extends Thread {
    private final String message;
    private final long interval; // in milliseconds

    DisplayMessageThread(String message, long interval) {
        this.message = message;
        this.interval = interval;
    }

    public void run() {
        try {
            while (true) {
                System.out.println(message);
                Thread.sleep(interval);
            }
        } catch (InterruptedException e) {
            System.out.println(Thread.currentThread().getName() + " interrupted.");
        }
    }
}

public class TwoThreadDemo {
    public static void main(String[] args) {
        DisplayMessageThread thread1 = new DisplayMessageThread("BMS College of
Engineering", 10000); // 10 seconds
        DisplayMessageThread thread2 = new DisplayMessageThread("CSE", 2000); // 2
seconds

        thread1.setName("Thread 1");
        thread2.setName("Thread 2");

        thread1.start();
        thread2.start();

        try {
            // Let the threads run for a while
            Thread.sleep(30000); // Let the program run for 30 seconds
        } catch (InterruptedException e) {
            System.out.println("Main thread interrupted.");
        }

        // Interrupt both threads to stop them
        thread1.interrupt();
        thread2.interrupt();

        System.out.println("Main thread exiting.");
    }
```

}

Output:


priyanka@PRIYANKAs-MacBook-Air java % javac TwoThreadDemo.java
priyanka@PRIYANKAs-MacBook-Air java % java TwoThreadDemo
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
Main thread exiting.
Thread 2 interrupted.
Thread 1 interrupted.
priyanka@PRIYANKAs-MacBook-Air java %

N PRIYANKA
1BM22CS167

→ Write a program that creates two threads one thread displays BMSCE every 10 seconds and another thread displays CSE every two seconds.

```
class DisplayMessageThread extends Thread
{
    private final String message;
    private final long interval;

    DisplayMessageThread(String message, long interval)
    {
        this.message = message;
        this.interval = interval;
    }

    public void run()
    {
        try
        {
            while(true)
            {
                System.out.println(message);
                Thread.sleep(interval);
            }
        }
        catch(InterruptedException e)
        {
            System.out.println(Thread.
            currentThread().getName() + " interrupt-
            -ed");
        }
    }
}
```

N PRIYANKA
1BM22CS167

```
publicclass TwoThreadDemo
{
    public static void main (String [] args)
    {
        DisplayMessageThread thread1 = new
          DisplayMessageThread ("BMSCE", 1000
        DisplayMessageThread thread2 = new
          DisplayMessageThread ("CSE", 2000);

        thread1.setName("Thread1");
        thread2.setName("Thread2");

        thread1.start();
        thread2.start();

        try
        {
            Thread.sleep(30000);
        }
        catch (InterruptedException e)
        {
            System.out.println ("Main thread
              Interrupted");
        }
        thread1.interrupt();
        thread2.interrupt();

        System.out.println("Main thread exiting")
    }
}
```

N PRIYANKA
1BM22CS167

Output:

BMSCE
CSE
CSE
CSE
CSE
CSE
BMSCE
CSE
CSE
CSE
CSE
BMSCE
CSE
CSE
CSE
CSE
CSE
Main thread exiting
Thread 2 interrupted
Thread 1 interrupted.

N PRIYANKA
1BM22CS167

Program 9:

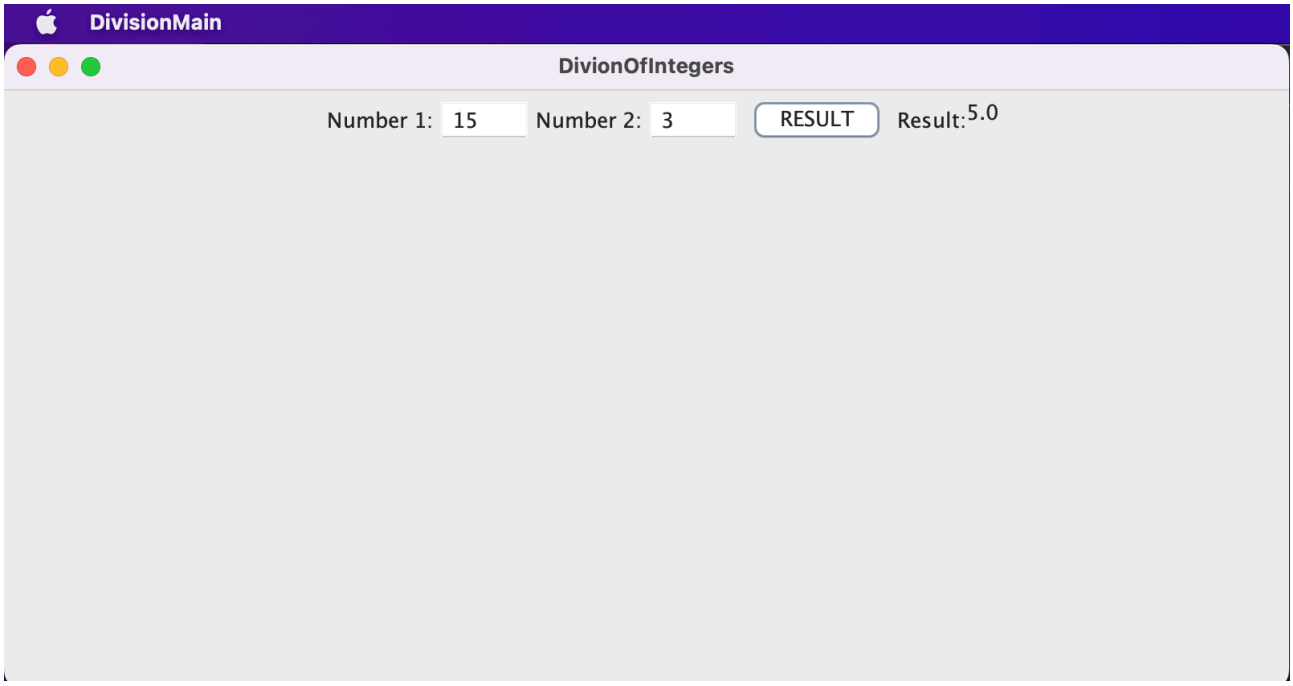
Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException.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

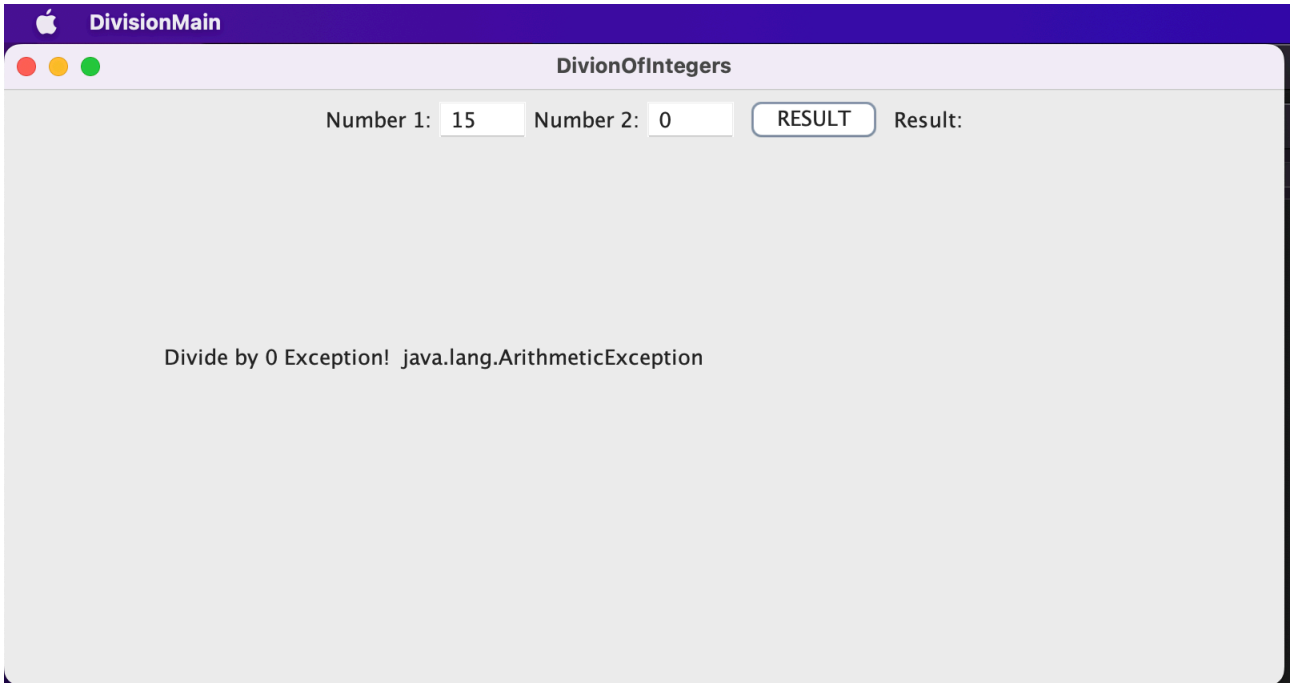
```
import java.awt.*;
import java.awt.event.*;

public class DivisionMain extends Frame implements ActionListener
{
	TextField num1,num2;
	Button dResult;
	Label outResult;
	String out="";
	double resultNum;
	int flag=0;

	public DivisionMain()
	{
		setLayout(new FlowLayout());

		dResult = new Button("RESULT");
		Label number1 = new Label("Number 1:",Label.RIGHT);
		Label number2 = new Label("Number 2:",Label.RIGHT);
		num1=new TextField(5);
		num2=new TextField(5);
		outResult = new Label("Result:",Label.RIGHT);

		add(number1);
		add(num1);
		add(number2);
		add(num2);
		add(dResult);
		add(outResult);

		num1.addActionListener(this);
		num2.addActionListener(this);
		dResult.addActionListener(this);
		addWindowListener(new WindowAdapter()
		{
			public void windowClosing(WindowEvent we)
			{
				System.exit(0);
			}
		});
	}

	public void actionPerformed(ActionEvent ae)
	{
```

```
		double n1,n2;
		try
		{
			if (ae.getSource() == dResult)
			{
				n1=Double.parseDouble(num1.getText());
				n2=Double.parseDouble(num2.getText());

				if(n2==0)
					throw new ArithmeticException();
				out=n1+" "+n2;
				resultNum=n1/n2;
				out=String.valueOf(resultNum);
				repaint();

			}
		}
		catch(ArithmeticException e2)
		{
			flag=1;
			out="Divide by 0 Exception!  "+e2;
			repaint();
		}
		catch(NumberFormatException e1)
		{
			flag=1;
			out="Number Format Exception!  "+e1;
			repaint();
		}

	}

	public void paint(Graphics g)
	{
		if(flag==0)
		g.drawString(out,outResult.getX()+outResult.getWidth(),outResult.getY()
+outResult.getHeight()-8);
		else
		g.drawString(out,100,200);
		flag=0;
	}

	public static void main(String[] args)
	{
		DivisionMain dm=new DivisionMain();
		dm.setSize(new Dimension(800,400));
		dm.setTitle("DivisionOfIntegers");
		dm.setVisible(true);
	}

}
```

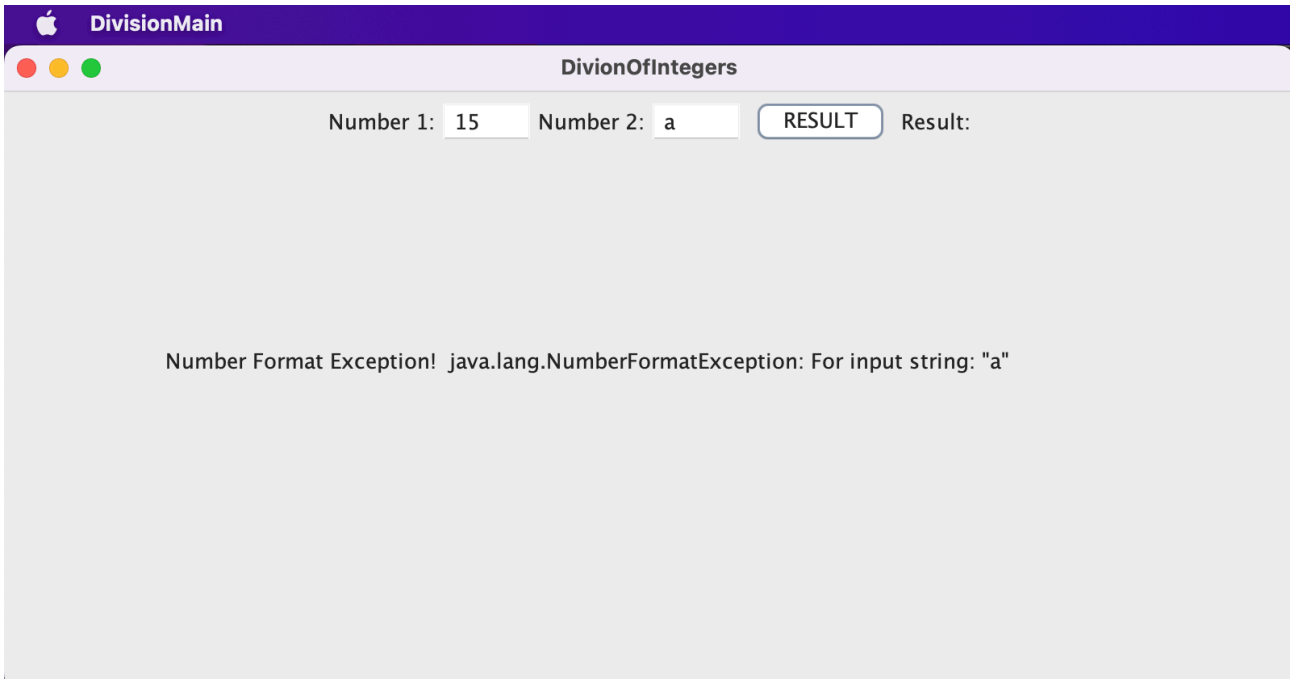
N PRIYANKA
1BM22CS167

Output





N PRIYANKA
1BM22CS167



N PRIYANKA
1BM22CS167

Program - 9

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

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

public Class DivisionMain extends Frame implements
  ActionListener
{
    TextField num1, num2;
    Button dresult;
    Label outResult;
    String out = "";
    double resultNum;
    int flag = 0;

    public DivisionMain()
    {
        setLayout(new FlowLayout());
        dResult = new Button("RESULT");
        Label number1 = new Label("Number1:
          Label.RIGHT);
        label number2 = new Label("Number2:
        , Label.RIGHT);
        num1 = new TextField(5);
        num2 = new TextField(5);
```

N PRIYANKA
1BM22CS167

```
outResult = new Label("Result:", Label.RIGHT);

add(numbers);
add(nums);
add(number2);
add(num2);
add(dResult);
add(outResult);

nums.addActionListener(this);
num2.addActionListener(this);
dResult.addActionListener(this);
addWindowListener(new WindowAdapter()
{
    public void windowClosing(
        WindowEvent we)
    {
        System.exit(0);
    }
});
}
public void actionPerformed(ActionEvent ae)
{
    double n1, n2;
    try
    {
        if(ae.getSource() == dResult)
        {
            n1 = Double.parseDouble(nums.
                getText());
            n2 = Double.parseDouble(num2.
                getText());
            if(n2 == 0)
                throw new ArithmeticExcep-
                    tion();
```

N PRIYANKA
1BM22CS167

```
            out = ns1 + " " + ns2;
            resultNum = ns1/ns2;
            out = String.valueOf(resultNum);
            repaint();
        }
    }
    catch (ArithmeticException e2)
    {
        flag = 1;
        out = "Divide by 0 Exception! " + e2;
        repaint();
    }
    catch (NumberFormatException e1)
    {
        flag=1;
        out = "Number Format Exception! " + e1;
        repaint();
    }
}
public void paint(Graphics g)
{
    if(flag==0)
    g.drawString(out, outResult.getX() + outResult.
                getWidth(), outResult.getY() +
                outResult.getHeight() - 8);
    else
    g.drawString(out, 100, 200);
    flag = 0;
}
public static void main(String[] args)
{
    DivisionMain dm = new DivisionMain();
    dm.setSize(new Dimension(800,400));
    dm.setTitle("Division of Integers");
```

N PRIYANKA
1BM22CS167

```
        dm.setVisible(true);
    }
}
```

O/P:

Number 1: [15], Number 2: [3] Result: 5

Program 10 Report:

Program 10 (Report)

AWT "Abstract Window toolkit

→ The AWT is a package in java which provides class to create and manage GUI interphases AWT includes various components event, handlers, layout managers and other utilities.

Components:

Frame:
A Frame is a window with title and border. used as main window in which other components are added.
Syntax: class IntegerDivision extends Frame.

Dialog:
It takes some form of input from the user, used to display model dialog to intract with user, used to perform uniform action, prompt the user, display messages.

Text Field:
It is a single line text displayed used for guiding the user to accept input.

label:
- Display area for short text
- Display static text
- prompts the user to enter text/input

eg: Num1 [ ]

Button:
- Triggers the actions when clicked
- Used for performing specific actions

N PRIYANKA
1BM22CS167

- used for submitting forms, conforming etc

Check Box:
- component which can be checked or unchecked.
- used to enable or disable
- used in the form with multiple choices.

Event Handling:

Event:
- Events are generated by user actions. Used for handling user input, such as mouse clicks, keypress or window events
- Events are processed by event listeners attached to the relevant components.

Event listener:
- objects that receives and handles events.
- This is for the implementation of event driven behaviour in GUI's.
- Event listener are registered with specific components to listen for particular type of events.

Action event:
- Event that indicates that component defined action occured.
- used for handling user actions
- Action generated when user clicks on component like button, checkbox etc

N PRIYANKA
1BM22CS167

Action listener:
- It recives action events
- It responds to the user actions.

Window event:
- Event indicating change to the data of a window
- Handles window related ~~em~~ events such as window opening, closing, resizing etc.

Window listener:
- It recives window events
- Used to implement methods to handle window-related events.

Layout Management

Flow layout:
- Arranges components in a centered flow one after the other.
- Used for arranging row or column with equal spacing between them.
- Used for creating forms etc.

Border-layout:
- layout manager that divides the container into 5 regions: north, south, east, weast and center.
- Used for arranging components at the edges, centre of the container.

layout Manager:
- Responsible for arranging components in container