B.M.S. COLLEGE OF ENGINEERING

Basavanagudi, Bengaluru- 560019

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LAB REPORT

On

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted By :

Adarsh Dev Singh

1BM22CS011

In partial fulfilment of

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

2023-24

Faculty-In-Charge

Swathi Sridharan

Assistant Professor

Department of Computer Science and Engineering

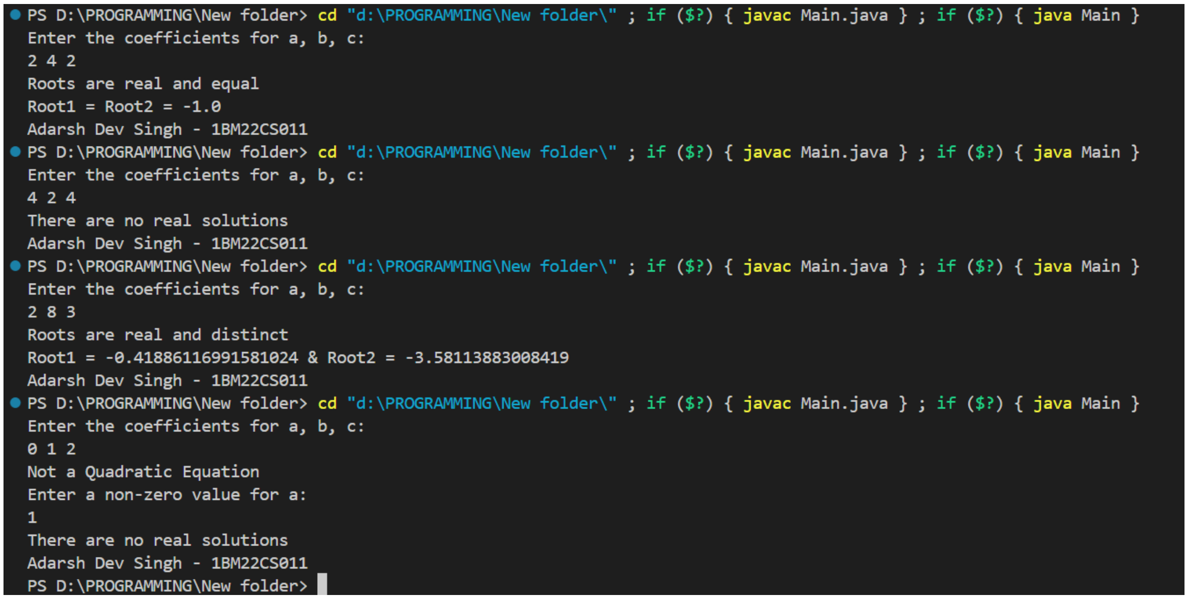
Program-1

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 Quadratic {
    int a, b, c;
    double r1, r2, d;

    void getd() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the coefficients for a, b, c:");
        a = s.nextInt();
        b = s.nextInt();
        c = s.nextInt();
    }

    void Compute() {
        while (a == 0) {
            System.out.println("Not a Quadratic Equation");
            System.out.println("Enter a non-zero value for a:");
            Scanner s = new Scanner(System.in);
            a = s.nextInt();
        }
        d = b * b - 4 * a * c;
        if (d == 0) {
            r1 = (-b) / (2.0 * a);
            System.out.println("Roots are real and equal");
            System.out.println("Root1 = Root2 = " + r1);

        } else if (d > 0) {
            r1 = ((-b) + (Math.sqrt(d))) / (2.0 * a);
            r2 = ((-b) - (Math.sqrt(d))) / (2.0 * a);
            System.out.println("Roots are real and distinct");
            System.out.println("Root1 = " + r1 + " & Root2 = " + r2);

        } else if (d < 0) {
            System.out.println("There are no real solutions");
        }
    }
}
```

```
public class Main {
    public static void main(String[] args) {
        Quadratic q = new Quadratic();
        q.getd();
        q.Compute();
        System.out.println("Adarsh Dev Singh - 1BM22CS011");
    }
}
```

OUTPUT :-



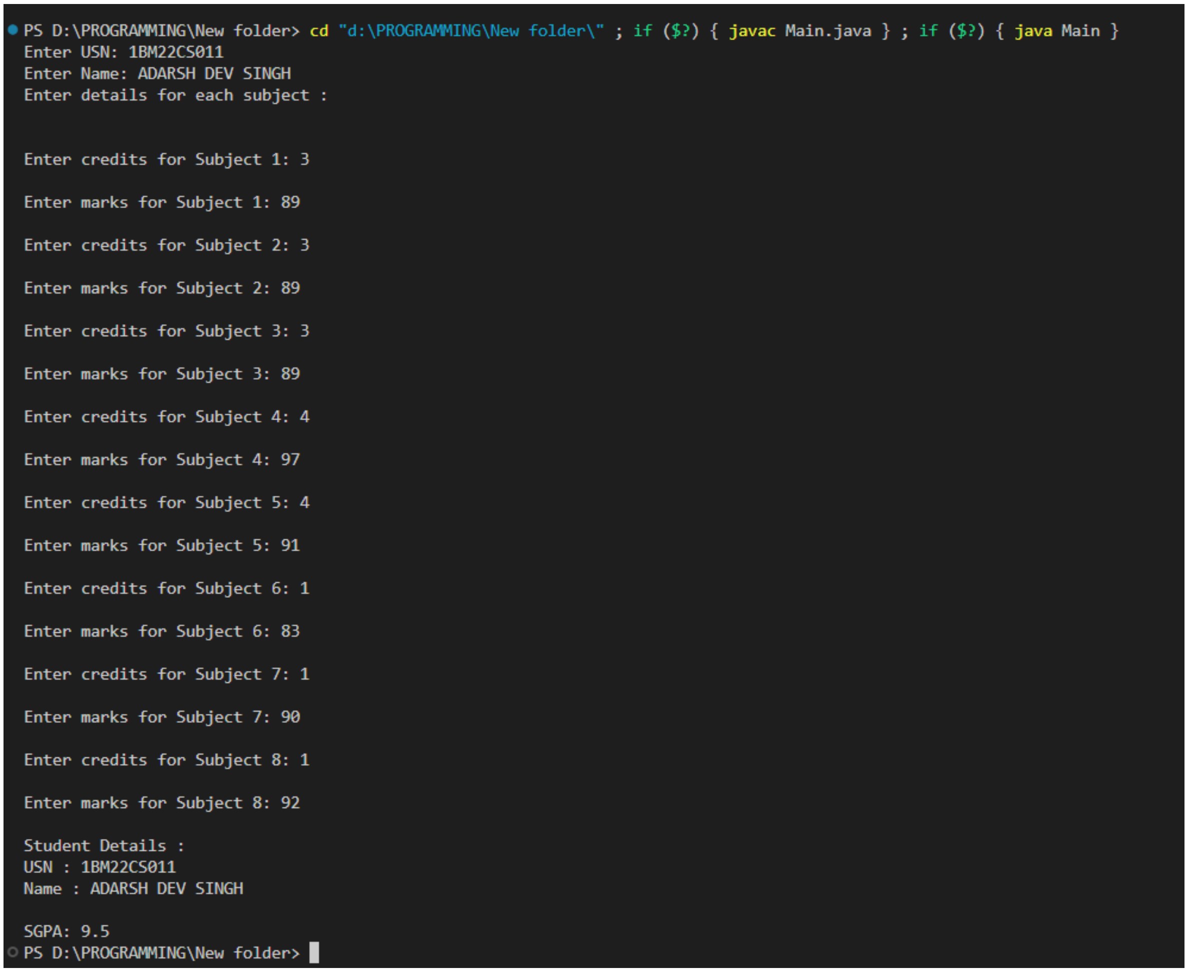
```
● PS D:\PROGRAMMING\New folder> cd "d:\PROGRAMMING\New folder\" ; if ($?) { javac Main.java } ; if ($?) { java Main }
  Enter the coefficients for a, b, c:
  2 4 2
  Roots are real and equal
  Root1 = Root2 = -1.0
  Adarsh Dev Singh - 1BM22CS011
● PS D:\PROGRAMMING\New folder> cd "d:\PROGRAMMING\New folder\" ; if ($?) { javac Main.java } ; if ($?) { java Main }
  Enter the coefficients for a, b, c:
  4 2 4
  There are no real solutions
  Adarsh Dev Singh - 1BM22CS011
● PS D:\PROGRAMMING\New folder> cd "d:\PROGRAMMING\New folder\" ; if ($?) { javac Main.java } ; if ($?) { java Main }
  Enter the coefficients for a, b, c:
  2 8 3
  Roots are real and distinct
  Root1 = -0.41886116991581024 & Root2 = -3.58113883008419
  Adarsh Dev Singh - 1BM22CS011
● PS D:\PROGRAMMING\New folder> cd "d:\PROGRAMMING\New folder\" ; if ($?) { javac Main.java } ; if ($?) { java Main }
  Enter the coefficients for a, b, c:
  0 1 2
  Not a Quadratic Equation
  Enter a non-zero value for a:
  1
  There are no real solutions
  Adarsh Dev Singh - 1BM22CS011
  PS D:\PROGRAMMING\New folder>
```

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.

```java
import java.util.Scanner;

class Student {
    String usn;
    String name;
    int[] credits = new int[8];
    int[] marks = new int[8];

    public void acceptDetails() {
        Scanner scanner = new Scanner(System.in);

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

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

        System.out.println("Enter details for each subject : \n");

        for (int i = 0; i < credits.length; i++) {
            System.out.print("\nEnter credits for Subject " + (i + 1) + ":
");
            credits[i] = scanner.nextInt();

            System.out.print("\nEnter marks for Subject " + (i + 1) + ": ");
            marks[i] = scanner.nextInt();
        }
        scanner.close();
    }

    public double calculateSGPA() {
        int totalCredits = 0;
        int weightedSum = 0;
        double ans;

        for (int i = 0; i < credits.length; i++) {
            totalCredits += credits[i];
            int gradePoints;

            gradePoints = (marks[i]/10)+1;

            if(gradePoints == 11){
                gradePoints=10;
            }
            else if(gradePoints<=4){
                gradePoints=0;
            }

            weightedSum += gradePoints * credits[i];
        }

        ans = (double) weightedSum / (double) totalCredits;
        return ans;
    }
}

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

        // Create a Student object
        Student student = new Student();

        // Accept and display details
        student.acceptDetails();

        System.out.println("\nStudent Details :");

        System.out.println("USN : " + student.usn);
        System.out.println("Name : " + student.name);

        // Calculate and display SGPA
        double sgpa = student.calculateSGPA();
        System.out.println("\nSGPA: " + sgpa);
        scanner.close();
    }
}
```

Output :-



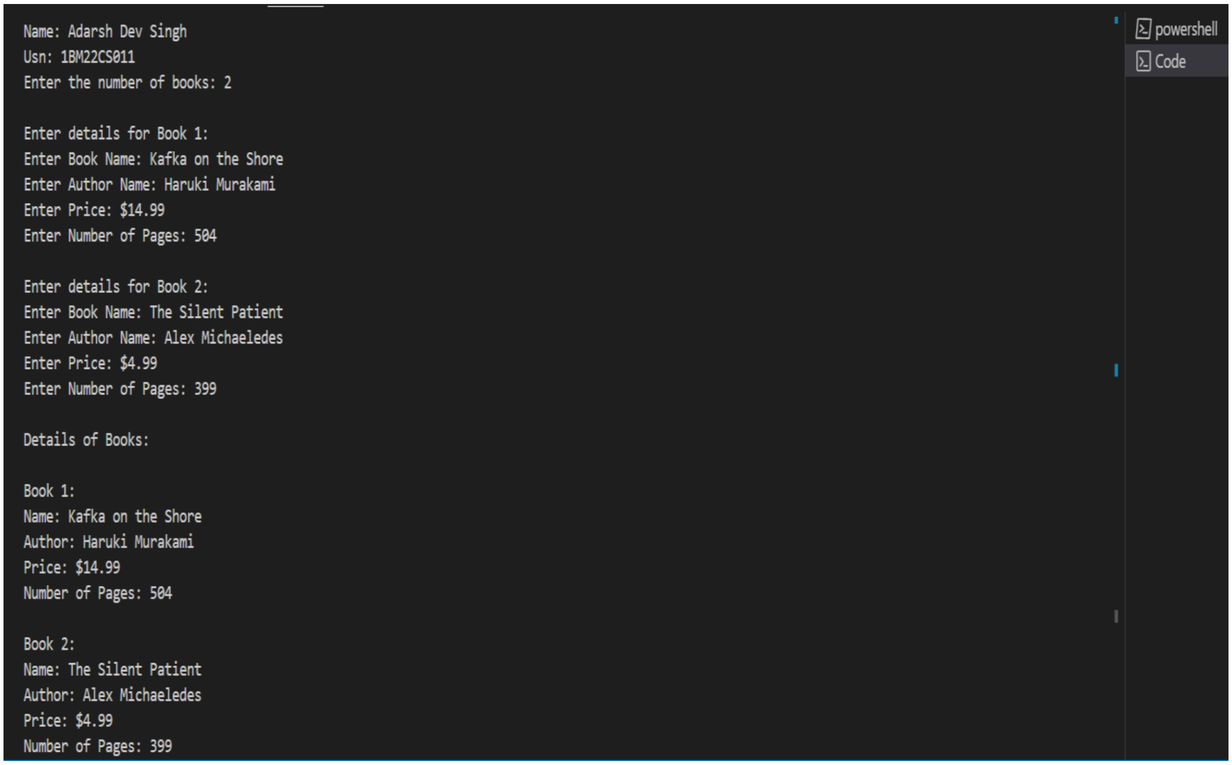
```
PS D:\PROGRAMMING\New folder> cd "d:\PROGRAMMING\New folder\" ; if ($?) { javac Main.java } ; if ($?) { java Main }
Enter USN: 1BM22CS011
Enter Name: ADARSH DEV SINGH
Enter details for each subject :


Enter credits for Subject 1: 3

Enter marks for Subject 1: 89

Enter credits for Subject 2: 3

Enter marks for Subject 2: 89

Enter credits for Subject 3: 3

Enter marks for Subject 3: 89

Enter credits for Subject 4: 4

Enter marks for Subject 4: 97

Enter credits for Subject 5: 4

Enter marks for Subject 5: 91

Enter credits for Subject 6: 1

Enter marks for Subject 6: 83

Enter credits for Subject 7: 1

Enter marks for Subject 7: 90

Enter credits for Subject 8: 1

Enter marks for Subject 8: 92

Student Details :
USN : 1BM22CS011
Name : ADARSH DEV SINGH

SGPA: 9.5
PS D:\PROGRAMMING\New folder>
```

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 Book {
    String name;
    String author;
    double price;
    int numPages;

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

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

    public String getDetails() {
        return "Name: " + name + "\nAuthor: " + author + "\nPrice: $" +
price + "\nNumber of Pages: " + numPages;
    }

    public String toString() {
        return getDetails();
    }
}

public class Main{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Name: Adarsh Dev Singh");
        System.out.println("Usn: 1BM22CS011");
        System.out.print("Enter the number of books: ");
        int n = scanner.nextInt();

        Book[] books = new Book[n];

        scanner.nextLine();
        for (int i = 0; i < n; i++) {
            System.out.println("\nEnter details for Book " + (i + 1) + ":");
            System.out.print("Enter Book Name: ");
            String name = scanner.nextLine();

            System.out.print("Enter Author Name: ");
            String author = scanner.nextLine();

            System.out.print("Enter Price: $");
            double price = scanner.nextDouble();

            System.out.print("Enter Number of Pages: ");
            int numPages = scanner.nextInt();
            scanner.nextLine();

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

        System.out.println("\nDetails of Books:");
        for (int i = 0; i < n; i++) {
            System.out.println("\nBook " + (i + 1) + ":\n" +
books[i].getDetails());
        }

        scanner.close();
    }
}
```

OUTPUT :-



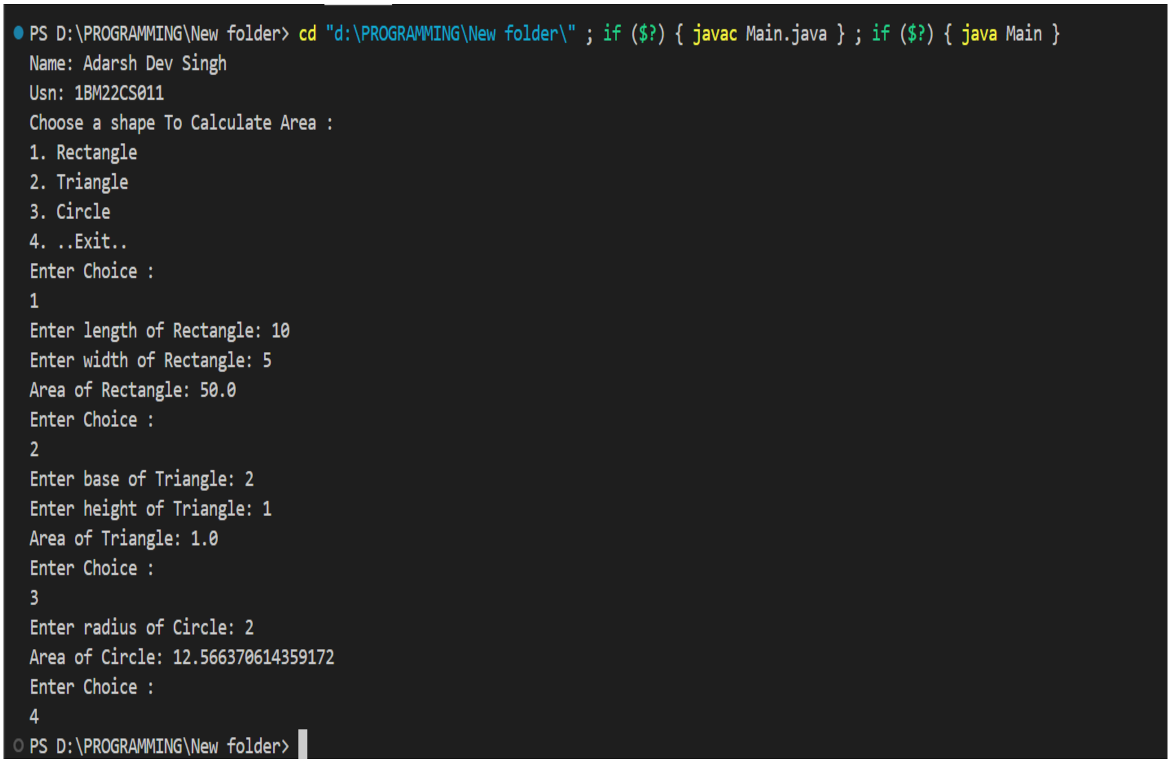
```
Name: Adarsh Dev Singh
Usn: 1BM22CS011
Enter the number of books: 2

Enter details for Book 1:
Enter Book Name: Kafka on the Shore
Enter Author Name: Haruki Murakami
Enter Price: $14.99
Enter Number of Pages: 504

Enter details for Book 2:
Enter Book Name: The Silent Patient
Enter Author Name: Alex Michaeledes
Enter Price: $4.99
Enter Number of Pages: 399

Details of Books:

Book 1:
Name: Kafka on the Shore
Author: Haruki Murakami
Price: $14.99
Number of Pages: 504

Book 2:
Name: The Silent Patient
Author: Alex Michaeledes
Price: $4.99
Number of Pages: 399
```

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;

abstract class Shape {
    double dimension1;
    double dimension2;

    public abstract void printArea();
}

class Rectangle extends Shape {
    // Constructor to initialize dimensions
    public Rectangle(double length, double width) {
        this.dimension1 = length;
        this.dimension2 = width;
    }

    // Implementation of abstract method
    @Override
    public void printArea() {
        double area = dimension1 * dimension2;
        System.out.println("Area of Rectangle: " + area);
    }
}

class Triangle extends Shape {
    // Constructor to initialize dimensions
    public Triangle(double base, double height) {
        this.dimension1 = base;
        this.dimension2 = height;
    }

    // Implementation of abstract method
    @Override
    public void printArea() {
        double area = 0.5 * dimension1 * dimension2;
        System.out.println("Area of Triangle: " + area)
```

```java
    }
}

class Circle extends Shape {
    // Constructor to initialize dimension
    public Circle(double radius) {
        this.dimension1 = radius;
    }

    // Implementation of abstract method
    @Override
    public void printArea() {
        double area = Math.PI * dimension1 * dimension1;
        System.out.println("Area of Circle: " + area);
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Name: Adarsh Dev Singh");
        System.out.println("Usn: 1BM22CS011");
        System.out.println("Choose a shape To Calculate Area :");
        System.out.println("1. Rectangle");
        System.out.println("2. Triangle");
        System.out.println("3. Circle");
        System.out.println("4. ..Exit..");

        int run = 1;
        while (run == 1) {
            System.out.println("Enter Choice : ");
            int choice = scanner.nextInt();
            switch (choice) {
                case 1:
                    System.out.print("Enter length of Rectangle: ");
                    double length = scanner.nextDouble();
                    System.out.print("Enter width of Rectangle: ");
                    double width = scanner.nextDouble();
                    Rectangle rectangle = new Rectangle(length, width);
                    rectangle.printArea();
                    break;
                case 2:
                    System.out.print("Enter base of Triangle: ");
                    double base = scanner.nextDouble();
                    System.out.print("Enter height of Triangle: ");
                    double height = scanner.nextDouble();
                    Triangle triangle = new Triangle(base, height);
                    triangle.printArea();
                    break;

                case 3:
                    System.out.print("Enter radius of Circle: ");
                    double radius = scanner.nextDouble();
                    Circle circle = new Circle(radius);
                    circle.printArea();
                    break;

                case 4:
                    run = 0;
                    break;
                default:
                    System.out.println("Invalid choice. Please choose a
valid choice....");
            }
        }

        scanner.close();
    }
}
```



```
● PS D:\PROGRAMMING\New folder> cd "d:\PROGRAMMING\New folder\" ; if ($?) { javac Main.java } ; if ($?) { java Main }
  Name: Adarsh Dev Singh
  Usn: 1BM22CS011
  Choose a shape To Calculate Area :
  1. Rectangle
  2. Triangle
  3. Circle
  4. ..Exit..
  Enter Choice :
  1
  Enter length of Rectangle: 10
  Enter width of Rectangle: 5
  Area of Rectangle: 50.0
  Enter Choice :
  2
  Enter base of Triangle: 2
  Enter height of Triangle: 1
  Area of Triangle: 1.0
  Enter Choice :
  3
  Enter radius of Circle: 2
  Area of Circle: 12.566370614359172
  Enter Choice :
  4
○ PS D:\PROGRAMMING\New folder>
```

Program-5

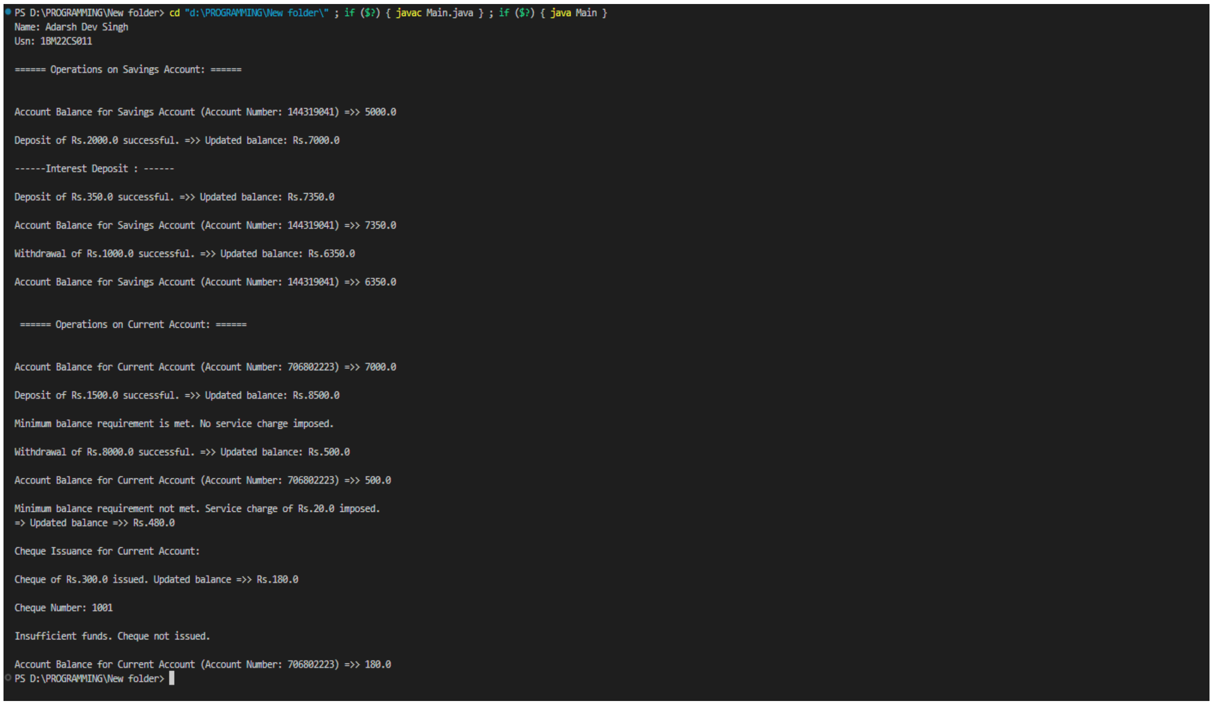
Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book 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.

```
class Account {
    String customerName;
    long accountNumber;
    String accountType;
    double balance;

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

    public void deposit(double amount) {
        balance += amount;
        System.out.println("\nDeposit of Rs." + amount + " successful. =>> Updated balance: Rs." + balance);
    }

    public void displayBalance() {
        System.out.println("\nAccount Balance for " + accountType + " Account (Account Number: " +
accountNumber + ") =>> " + balance);
    }

    public void depositInterest(double rate) {
        if (accountType.equals("Savings")) {
            double interest = balance * (rate / 100);
            System.out.println("\n------Interest Deposit : ------");
            deposit(interest);
        } else {
            System.out.println("\nInterest is not applicable for Current Account.");
        }
    }

    public void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
            System.out.println("\nWithdrawal of Rs." + amount + " successful. =>> Updated balance: Rs." +
balance);
        } else {
            System.out.println("\nInsufficient funds. Withdrawal not allowed.");
        }
    }
}

class CurAcct extends Account {
    private double minBalance;
    private double serviceCharge;
    private int chequeNumber = 1001;

    public CurAcct(String customerName, long accountNumber, double balance, double minBalance, double
serviceCharge) {
        super(customerName, accountNumber, "Current", balance);
        this.minBalance = minBalance;
        this.serviceCharge = serviceCharge;
    }

    public void checkMinBalance() {
        if (balance < minBalance) {
            balance -= serviceCharge;
            System.out.println("\nMinimum balance requirement not met. Service charge of Rs." +
serviceCharge + " imposed.");
            System.out.print("=> Updated balance =>> Rs." + balance);
        } else {
            System.out.println("\nMinimum balance requirement is met. No service charge imposed.");
        }
    }

    public void issueCheque(double amount) {
        if (balance >= amount) {
            balance -= amount;
            System.out.println("\nCheque of Rs." + amount + " issued. Updated balance =>> Rs." + balance);
            System.out.println("\nCheque Number: " + chequeNumber++);
        } else {
            System.out.println("\nInsufficient funds. Cheque not issued.");
        }
    }
}

class SavAcct extends Account {
    double interestRate;

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

public class Main {
    public static void main(String[] args) {
            System.out.println("Name: Adarsh Dev Singh");
            System.out.println("Usn: 1BM22CS011");

            SavAcct savingsAccount = new SavAcct("Adarsh ", 144319041, 5000, 5);
            CurAcct currentAccount = new CurAcct("Ansh", 706802223, 7000, 1000, 20);

            System.out.println("\n====== Operations on Savings Account: ======\n");
            savingsAccount.displayBalance();
            savingsAccount.deposit(2000);
            savingsAccount.depositInterest(5);
            savingsAccount.displayBalance();
            savingsAccount.withdraw(1000);
            savingsAccount.displayBalance();

            System.out.println("\n\n ====== Operations on Current Account: ======\n");
            currentAccount.displayBalance();
            currentAccount.deposit(1500);
            currentAccount.checkMinBalance();
            currentAccount.withdraw(8000);
            currentAccount.displayBalance();
            currentAccount.checkMinBalance();

            System.out.println("\n\nCheque Issuance for Current Account:");
            currentAccount.issueCheque(300);
            currentAccount.issueCheque(500);
            currentAccount.displayBalance();
        }
}
```

OUTPUT :-



```
PS D:\PROGRAMMING\New folder> cd "d:\PROGRAMMING\New folder\" ; if ($?) { javac Main.java } ; if ($?) { java Main }
Name: Adarsh Dev Singh
Usn: 1BM22CS011

====== Operations on Savings Account: ======


Account Balance for Savings Account (Account Number: 144319041) =>> 5000.0

Deposit of Rs.2000.0 successful. =>> Updated balance: Rs.7000.0

------Interest Deposit : ------

Deposit of Rs.350.0 successful. =>> Updated balance: Rs.7350.0

Account Balance for Savings Account (Account Number: 144319041) =>> 7350.0

Withdrawal of Rs.1000.0 successful. =>> Updated balance: Rs.6350.0

Account Balance for Savings Account (Account Number: 144319041) =>> 6350.0


 ====== Operations on Current Account: ======


Account Balance for Current Account (Account Number: 706802223) =>> 7000.0

Deposit of Rs.1500.0 successful. =>> Updated balance: Rs.8500.0

Minimum balance requirement is met. No service charge imposed.

Withdrawal of Rs.8000.0 successful. =>> Updated balance: Rs.500.0

Account Balance for Current Account (Account Number: 706802223) =>> 500.0

Minimum balance requirement not met. Service charge of Rs.20.0 imposed.
=> Updated balance =>> Rs.480.0

Cheque Issuance for Current Account:

Cheque of Rs.300.0 issued. Updated balance =>> Rs.180.0

Cheque Number: 1001

Insufficient funds. Cheque not issued.

Account Balance for Current Account (Account Number: 706802223) =>> 180.0
PS D:\PROGRAMMING\New folder> 
```

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.

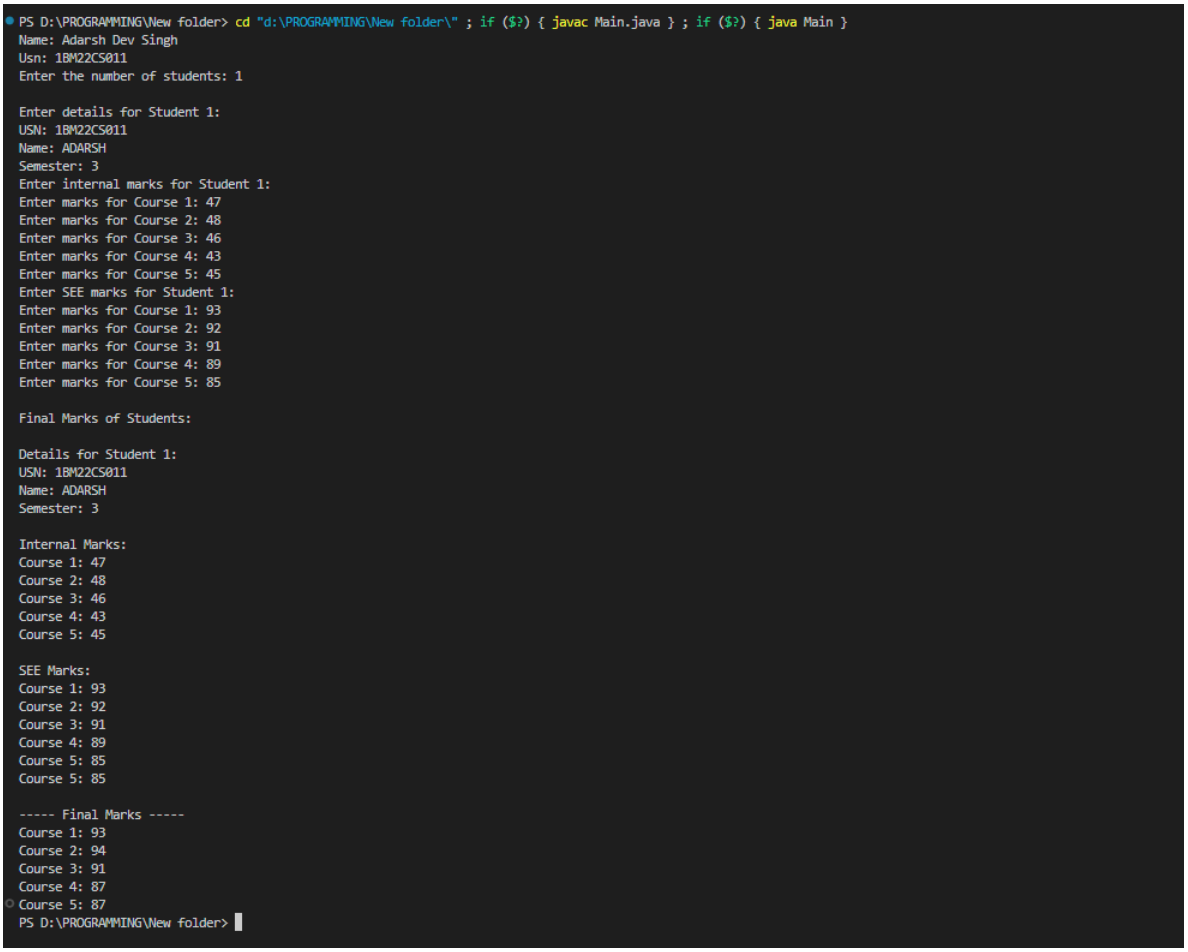
```
package CIE;
public class Internals {
    public int marks []=new int[5];
}

package CIE;

public class Student {
    public String usn;
    public String name;
    public int sem;
}

package SEE;
import CIE.Student;

public class External extends Student {
    public int[] seeMarks = new int[5];
}
```

```
import CIE.Student;
import CIE.Internals;
import SEE.External;
import java.util.Scanner;

public class FinalMarks {
    public static void main(String[] args) {
   System.out.println("Name: Adarsh Dev Singh");
    System.out.println("Usn: 1BM22CS011");

        Scanner scanner = new Scanner(System.in);

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

        Student[] students = new Student[n];
        Internals[] internals = new Internals[n];
        External[] externals = new External[n];
```

```
for (int i = 0; i < n; i++) {
    students[i] = new Student();
    internals[i] = new Internals();
    externals[i] = new External();

    System.out.println("\nEnter details for Student " + (i + 1) + ":");
    System.out.print("USN: ");
    students[i].usn = scanner.next();
    System.out.print("Name: ");
    students[i].name = scanner.next();
    System.out.print("Semester: ");
    students[i].sem = scanner.nextInt();

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

    System.out.println("Enter SEE marks for Student " + (i + 1) + ":");
    for (int j = 0; j < 5; j++) {
        System.out.print("Enter marks for Course " + (j + 1) + ": ");
        externals[i].seeMarks[j] = scanner.nextInt();
    }
}

System.out.println("\nFinal Marks of Students:");
for (int i = 0; i < n; i++) {
    System.out.println("\nDetails for Student " + (i + 1) + ":");
    System.out.println("USN: " + students[i].usn);
    System.out.println("Name: " + students[i].name);
    System.out.println("Semester: " + students[i].sem);

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

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

    // Calculate and display final marks
    System.out.println("\n----- Final Marks -----");
    for (int j = 0; j < 5; j++) {
        int finalMarks = internals[i].internalMarks[j] + ( externals[i].seeMarks[j] / 2 );
        System.out.println("Course " + (j + 1) + ": " + finalMarks);
    }
}
```

```
        scanner.close();
    }}
```

COURSE :-



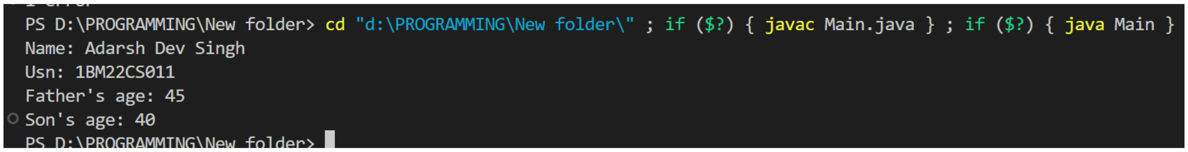
```
PS D:\PROGRAMMING\New folder> cd "d:\PROGRAMMING\New folder\" ; if ($?) { javac Main.java } ; if ($?) { java Main }
Name: Adarsh Dev Singh
Usn: 1BM22CS011
Enter the number of students: 1

Enter details for Student 1:
USN: 1BM22CS011
Name: ADARSH
Semester: 3
Enter internal marks for Student 1:
Enter marks for Course 1: 47
Enter marks for Course 2: 48
Enter marks for Course 3: 46
Enter marks for Course 4: 43
Enter marks for Course 5: 45
Enter SEE marks for Student 1:
Enter marks for Course 1: 93
Enter marks for Course 2: 92
Enter marks for Course 3: 91
Enter marks for Course 4: 89
Enter marks for Course 5: 85

Final Marks of Students:

Details for Student 1:
USN: 1BM22CS011
Name: ADARSH
Semester: 3

Internal Marks:
Course 1: 47
Course 2: 48
Course 3: 46
Course 4: 43
Course 5: 45

SEE Marks:
Course 1: 93
Course 2: 92
Course 3: 91
Course 4: 89
Course 5: 85
Course 5: 85

----- Final Marks -----
Course 1: 93
Course 2: 94
Course 3: 91
Course 4: 87
Course 5: 87
PS D:\PROGRAMMING\New folder> 
```

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.

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

class Father {
    private int age;

    public Father(int age) throws WrongAge {
        if (age < 0) {
            throw new WrongAge("Age cannot be negative");
        }
        this.age = age;
    }

    public int getAge() {
        return age;
    }
}

class Son extends Father {
    private int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAge {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new WrongAge("Son's age cannot be greater than or equal to father's age");
        }
        this.sonAge = sonAge;
    }

    public int getSonAge() {
        return sonAge;
    }
}

public class ExceptionInheritanceDemo {
    public static void main(String[] args) {
            System.out.println("Name: Adarsh Dev Singh");
        System.out.println("Usn: 1BM22CS011");
        try {
            Father father = new Father(45);
            Son son = new Son(45, 40);
            System.out.println("Father's age: " + father.getAge());
            System.out.println("Son's age: " + son.getSonAge());
        } catch (WrongAge e) {
            System.out.println("Exception caught: " + e.getMessage());
        }
    }
}
```

OUTPUT: -



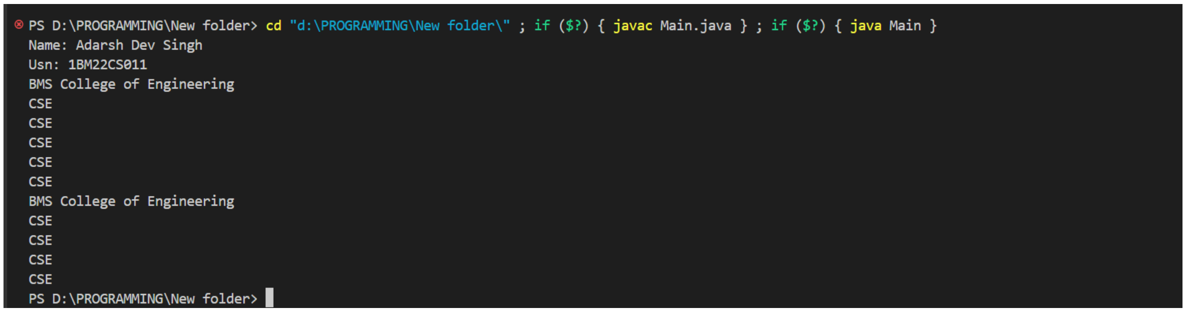
```
PS D:\PROGRAMMING\New folder> cd "d:\PROGRAMMING\New folder\" ; if ($?) { javac Main.java } ; if ($?) { java Main }
Name: Adarsh Dev Singh
Usn: 1BM22CS011
Father's age: 45
Son's age: 40
PS D:\PROGRAMMING\New folder>
```

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 DisplayMessage extends Thread {
    String message;
    int interval;

    public DisplayMessage(String message, int interval) {
        this.message = message;
        this.interval = interval;
    }

    public void run() {
        while (true) {
            try {
                System.out.println(message);
                Thread.sleep(interval * 1000); // Convert seconds to milliseconds
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

public class Main {
    public static void main(String[] args) {
         System.out.println("Name: Adarsh Dev Singh");
        System.out.println("Usn: 1BM22CS011");
        DisplayMessage thread1 = new DisplayMessage("BMS College of Engineering", 10);
        DisplayMessage thread2 = new DisplayMessage("CSE", 2);

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

Output:-



```
PS D:\PROGRAMMING\New folder> cd "d:\PROGRAMMING\New folder\" ; if ($?) { javac Main.java } ; if ($?) { java Main }
Name: Adarsh Dev Singh
Usn: 1BM22CS011
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
PS D:\PROGRAMMING\New folder> 
```

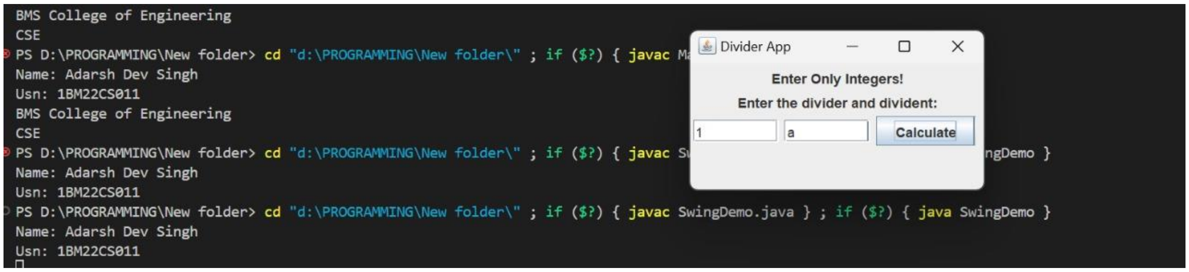
Program-9

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

class SwingDemo {
   SwingDemo() {
      // create jframe container
      JFrame jfrm = new JFrame("Divider App");
      jfrm.setSize(275, 150);
      jfrm.setLayout(new FlowLayout());
      // to terminate on close
      jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

      // text label
      JLabel jlab = new JLabel("Enter the divider and divident:");

      // add text field for both numbers
      JTextField ajtf = new JTextField(8);
      JTextField bjtf = new JTextField(8);

      // calc button
      JButton button = new JButton("Calculate");

      // labels
      JLabel err = new JLabel();
      JLabel alab = new JLabel();
      JLabel blab = new JLabel();
      JLabel anslab = new JLabel();
      // add in order :)
      jfrm.add(err); // to display error bois
      jfrm.add(jlab);
      jfrm.add(ajtf);
      jfrm.add(bjtf);
      jfrm.add(button);
      jfrm.add(alab);
      jfrm.add(blab);
      jfrm.add(anslab);

      ActionListener l = new ActionListener() {
          public void actionPerformed(ActionEvent evt) {
              System.out.println("Action event from a text field");
          }
      };
      ajtf.addActionListener(l);
      bjtf.addActionListener(l);

      button.addActionListener(new ActionListener() {
          public void actionPerformed(ActionEvent evt) {
              try {
                  int a = Integer.parseInt(ajtf.getText());
                  int b = Integer.parseInt(bjtf.getText());
                  int ans = a / b;

                  alab.setText("\nA = " + a);
                  blab.setText("\nB = " + b);
                  anslab.setText("\nAns = " + ans);
                  err.setText("");
              } catch (NumberFormatException e) {
                  alab.setText("");
                  blab.setText("");
                  anslab.setText("");
                  err.setText("Enter Only Integers!");
              } catch (ArithmeticException e) {
                  alab.setText("");
                  blab.setText("");
                  anslab.setText("");
                  err.setText("B should be NON zero!");
              }
          }
      });

      // display frame
```

```
        jfrm.setVisible(true);
    }

    public static void main(String args[]) {
            System.out.println("Name: Adarsh Dev Singh");
        System.out.println("Usn: 1BM22CS011");
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new SwingDemo();
            }
        });
    }
}
```

OUTPUT : -



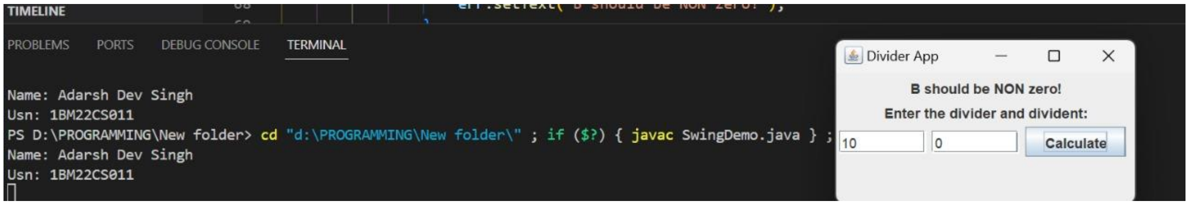
```
BMS College of Engineering
CSE
PS D:\PROGRAMMING\New folder> cd "d:\PROGRAMMING\New folder\" ; if ($?) { javac Ma
Name: Adarsh Dev Singh
Usn: 1BM22CS011
BMS College of Engineering
CSE
PS D:\PROGRAMMING\New folder> cd "d:\PROGRAMMING\New folder\" ; if ($?) { javac Su                              ngDemo }
Name: Adarsh Dev Singh
Usn: 1BM22CS011
PS D:\PROGRAMMING\New folder> cd "d:\PROGRAMMING\New folder\" ; if ($?) { javac SwingDemo.java } ; if ($?) { java SwingDemo }
Name: Adarsh Dev Singh
Usn: 1BM22CS011
```

Divider App

Enter Only Integers!

Enter the divider and divident:

1 | a



```
err.setText("B should be NON zero!");
```

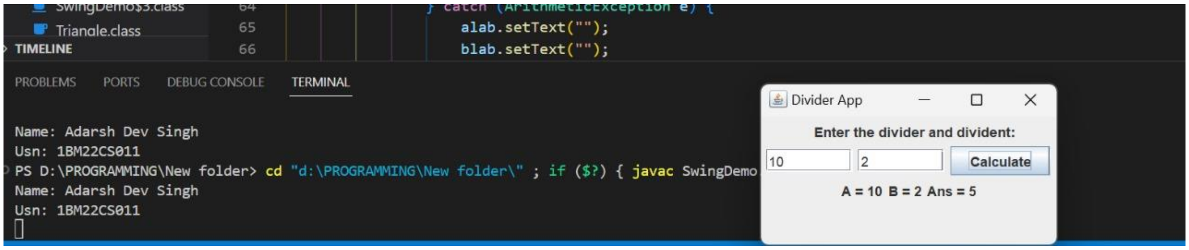
```
Name: Adarsh Dev Singh
Usn: 1BM22CS011
PS D:\PROGRAMMING\New folder> cd "d:\PROGRAMMING\New folder\" ; if ($?) { javac SwingDemo.java } ;
Name: Adarsh Dev Singh
Usn: 1BM22CS011
```

Divider App

B should be NON zero!

Enter the divider and divident:

10 | 0



```
} catch (ArithmeticException e) {
    alab.setText("");
    blab.setText("");
```

```
Name: Adarsh Dev Singh
Usn: 1BM22CS011
PS D:\PROGRAMMING\New folder> cd "d:\PROGRAMMING\New folder\" ; if ($?) { javac SwingDemo
Name: Adarsh Dev Singh
Usn: 1BM22CS011
```

Divider App

Enter the divider and divident:

10 | 2

A = 10 B = 2 Ans = 5