

**VISVESVARAYA TECHNOLOGICAL
UNIVERSITY, JNANASANGAMA,
BELGAUM - 590014, KARNATAKA**



LABORATORY RECORD
ON
Object Oriented Java Programming
(23CS3PCOOJ) Submitted by
Mohammed Uzair Obaid (1BM22CS159)

In partial fulfilment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING



B. M. S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU – 560019
December-2022 to April-2023

INDEX

Sl. No.	Date	Experiment Title	Page No.
1	18.12.2023	Laboratory Program – 1	3-5
2	01.01.2024	Laboratory Program – 2	6-9
3	01.01.2024	Laboratory Program – 3	10-14
4	08.01.2024	Laboratory Program – 4	15-18
5	08.01.2024	Laboratory Program – 5	19-26
6	22.01.2024	Laboratory Program – 6	27-31
7	22.01.2024	Laboratory Program – 7	32-34
8	05.02.2024	Laboratory Program – 8	35-37
9	19.02.2024	Laboratory Program – 9 (And Report on few AWT program)	38-46

LABORATORY PROGRAM - 1

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

```
import java.util.Scanner;

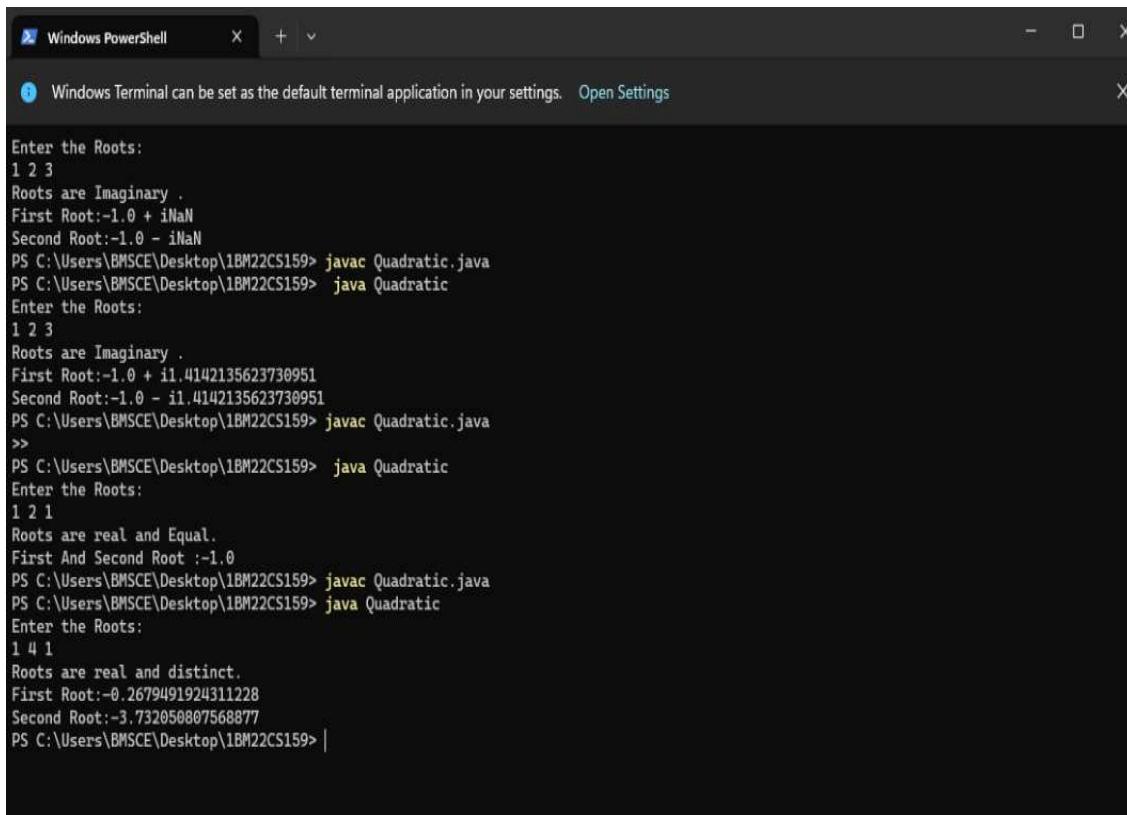
public class Quadratic {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter the coefficients a, b, and c:");
        double a = scanner.nextDouble();
        double b = scanner.nextDouble();
        double c = scanner.nextDouble();

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

        if (d > 0) {
            System.out.println("Roots are real and distinct.");
            double r1 = (-b + Math.sqrt(d)) / (2 * a);
            double r2 = (-b - Math.sqrt(d)) / (2 * a);
            System.out.println("First Root: " + r1 + "\nSecond Root: " + r2);
        } else if (d == 0) {
            System.out.println("Roots are real and equal.");
            double r1 = -b / (2 * a);
            System.out.println("First and Second Root: " + r1);
        } else {
            System.out.println("Roots are imaginary.");
            double realPart = -b / (2 * a);
            double imaginaryPart = Math.sqrt(Math.abs(d)) / (2 * a);
            System.out.println("First Root: " + realPart + " + i" + imaginaryPart);
            System.out.println("Second Root: " + realPart + " - i" + imaginaryPart);
        }
    }
}
```

OUTPUT



A screenshot of a Windows Terminal window titled "Windows PowerShell". The window displays the output of a Java application named "Quadratic". The application prompts the user for three coefficients (Roots) and then calculates the roots based on the discriminant. The terminal shows three separate runs of the application with different input values (1 2 3, 1 2 1, and 1 4 1), each resulting in either imaginary or real roots.

```
Windows PowerShell X + v

Windows Terminal can be set as the default terminal application in your settings. Open Settings X

Enter the Roots:
1 2 3
Roots are Imaginary .
First Root:-1.0 + iNaN
Second Root:-1.0 - iNaN
PS C:\Users\BMSCE\Desktop\1BM22CS159> javac Quadratic.java
PS C:\Users\BMSCE\Desktop\1BM22CS159> java Quadratic
Enter the Roots:
1 2 3
Roots are Imaginary .
First Root:-1.0 + i1.4142135623730951
Second Root:-1.0 - i1.4142135623730951
PS C:\Users\BMSCE\Desktop\1BM22CS159> javac Quadratic.java
>>
PS C:\Users\BMSCE\Desktop\1BM22CS159> java Quadratic
Enter the Roots:
1 2 1
Roots are real and Equal.
First And Second Root :-1.0
PS C:\Users\BMSCE\Desktop\1BM22CS159> javac Quadratic.java
PS C:\Users\BMSCE\Desktop\1BM22CS159> java Quadratic
Enter the Roots:
1 4 1
Roots are real and distinct.
First Root:-0.2679491924311228
Second Root:-3.732050807568877
PS C:\Users\BMSCE\Desktop\1BM22CS159> |
```

Lab Program : 1:

Program to find roots of a quadratic equation

```
import java.util.Scanner;  
class Quadratic {  
    public static void main(String args[]) {  
        Scanner s1 = new Scanner(System.in);  
        System.out.println("Enter the Roots:");  
        Double a = s1.nextDouble();  
        Double b = s1.nextDouble();  
        Double c = s1.nextDouble();  
  
        Double r1, r2, d = b * b - 4 * a * c;  
        if (d > 0) {  
            System.out.println("Roots are real and  
            distinct.");  
            r1 = (-b + Math.sqrt(d)) / (2 * a);  
            r2 = (-b - Math.sqrt(d)) / (2 * a);  
            System.out.println("First Root: " + r1 + "\n"  
                "Second Root: " + r2);  
        }  
    }  
}
```

```
else if (d == 0) {  
    System.out.println("Roots are real and  
    Equal.");  
    r1 = (-b) / (2 * a);  
    System.out.println("First And Second Root: " + r1);  
}  
else {  
    System.out.println("Roots are Imaginary.");  
    r1 = (-b) / (2 * a);  
    r2 = Math.sqrt(Math.abs(d)) / (2 * a);  
    System.out.println("First Root: " + r1 + " + i " + r2);  
    System.out.println("Second Root: " + r1 + " - i " + r2);  
}  
}
```

Output:-

Enter the Roots:
1 2 3
Roots are Imaginary.
First Root: -1.0 + 1.4142
Second Root: -1.0 - 1.4142

Enter the Roots:

1 2 1
Roots are real and Equal.
First and Second Root: -1.0

Enter the Roots:

1 4 1
Roots are real and distinct.
First Root: -0.267949
Second Root: -3.73205

LABORATORY 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 Student {
    void display(String usn, String name, int cr[], int mrks[]) {
        System.out.println("USN: " + usn + "\n" + "NAME:" + name);
        for (int i = 0; i < 5; i++) {
            System.out.println("Credits: " + cr[i] + " Marks: " + mrks[i]);
        }
    }

    void sgpa(int cr[], int mrks[]) {
        int maxCreMarks = 0, creMarks = 0;
        for (int i = 0; i < 5; i++) {
            maxCreMarks += cr[i] * 10;
            creMarks += ((mrks[i] / 10) + 1) * cr[i];
        }

        double sgpa = (double) (creMarks * 10) / (double) maxCreMarks;
        System.out.println("SGPA: " + sgpa);
    }
}

public class Main {
    public static void main(String[] args) {
        String usn, name;
        int credits[] = new int[5];
        int marks[] = new int[5];
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter usn:");
        usn = scanner.next();
        System.out.println("Enter Name:");
        name = scanner.next();

        System.out.println("Enter Credits for the five subjects:");
        for (int i = 0; i < 5; i++) {
            credits[i] = scanner.nextInt();
        }

        System.out.println("Enter the marks for all subjects:");
        for (int i = 0; i < 5; i++) {
            marks[i] = scanner.nextInt();
        }
    }
}
```

```

Student student = new Student();
student.display(usn, name, credits, marks);
student.sgpa(credits, marks);
}
}

```

Bafna Gold
Date: 1-1-24
Page:

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

→ import java.util.Scanner;

```

class Student {
    void display(String usn, String name,
    int credits[], int marks[]) {
        System.out.println("USN: " + usn + "\n" + "name: "
                           + name);
        for (int i = 0; i < 5; i++) {
            System.out.println("Credits: " +
                               credits[i] + " Marks: " + marks[i]);
        }
    }

    void sgpa(int credits[], int marks[]) {
        int maxCgMarks = credits[0] * 10;
        for (int i = 1; i < 5; i++) {
            if (maxCgMarks < credits[i] * 10) {
                maxCgMarks = credits[i] * 10;
            }
        }
        double sgpa = (maxCgMarks * 10) / maxCgMarks;
        System.out.println("SGPA: " + sgpa);
    }

    public static void main(String[] args) {
        String usn, name;
        int credits[] = new int[5];
        int marks[] = new int[5];
        Scanner s1 = new Scanner(System.in);
        System.out.println("Enter usn:");
        usn = s1.next();
        System.out.println("Enter Credits for the
                           five subjects:");
    }
}

```

Version to be issued by Board of Secondary Education, Maharashtra. No part of this publication may be reproduced without written permission from the Board.

```
for (int i = 0; i < 5; i++) {  
    credits[i] = s1.nextInt();
```

}

```
System.out.println("Enter the marks for all  
Subjects : ");
```

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

```
    marks[i] = s1.nextInt();
```

}

```
Student student = new Student();
```

```
student.display(usn, name, credits, marks);
```

```
student.sgpai(credits, marks);
```

```
}
```

```
}
```

Output :-

```
Enter usn: 1BM22CS159
```

```
Enter Name: UZAIR
```

```
Enter credits for the five subjects:
```

```
4 4 3 3 2
```

```
Enter the marks for all subjects:
```

```
91 96 88 86 93
```

```
USN: 1BM22CS159
```

```
NAME: UZAIR
```

```
Credits: 4 Marks: 91
```

```
Credits: 4 Marks: 96
```

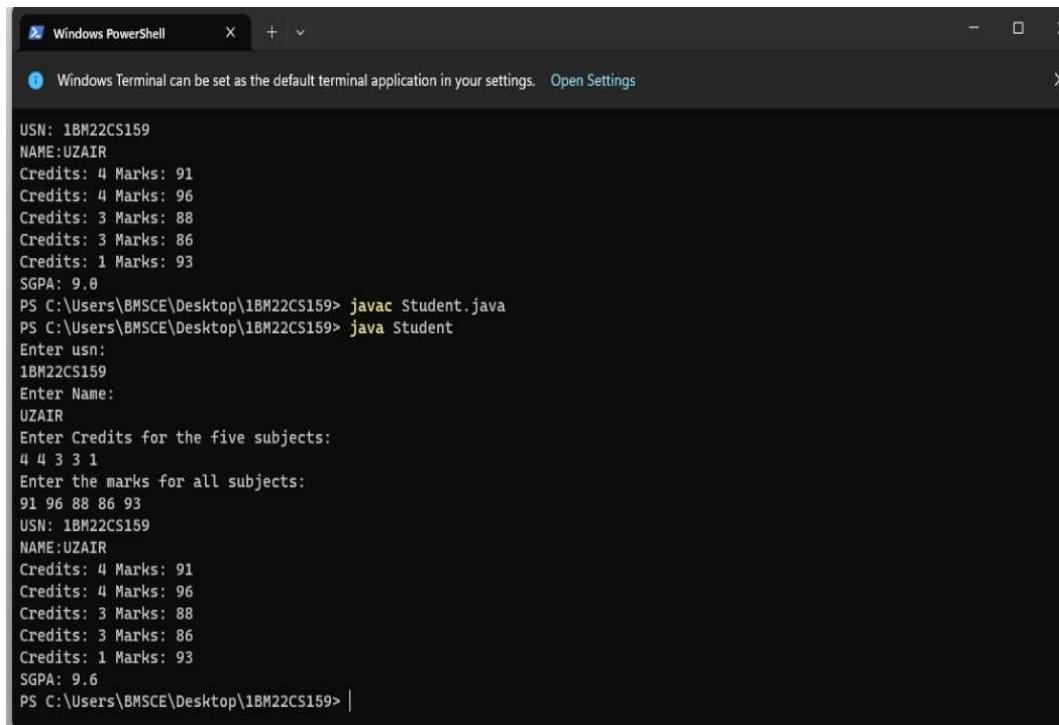
```
Credits: 3 Marks: 88
```

```
Credits: 3 Marks: 86
```

```
Credits: 1 Marks: 93
```

```
SGPA: 9.6
```

OUTPUT



```
Windows PowerShell X + v
Windows Terminal can be set as the default terminal application in your settings. Open Settings X

USN: 1BM22CS159
NAME:UZAIR
Credits: 4 Marks: 91
Credits: 4 Marks: 96
Credits: 3 Marks: 88
Credits: 3 Marks: 86
Credits: 1 Marks: 93
SGPA: 9.0
PS C:\Users\BMSCE\Desktop\1BM22CS159> javac Student.java
PS C:\Users\BMSCE\Desktop\1BM22CS159> java Student
Enter usn:
1BM22CS159
Enter Name:
UZAIR
Enter Credits for the five subjects:
4 4 3 3 1
Enter the marks for all subjects:
91 96 88 86 93
USN: 1BM22CS159
NAME:UZAIR
Credits: 4 Marks: 91
Credits: 4 Marks: 96
Credits: 3 Marks: 88
Credits: 3 Marks: 86
Credits: 1 Marks: 93
SGPA: 9.6
PS C:\Users\BMSCE\Desktop\1BM22CS159> |
```

LABORATORY 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 String toString() {
        return "Book Details:\nName: " + name + "\nAuthor: " + author +
               "\nPrice: Rs." + price + "\nNumber of Pages: " + numPages;
    }
}

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

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

        Book[] books = new Book[n];

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

            System.out.print("Author: ");
            String author = scanner.next();

            System.out.print("Price: Rs.");
            double price = scanner.nextDouble();
        }
    }
}
```

```

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

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

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

```

Bafna Gold

Books

Create a class Book which contains four members: name, author, price, numPages. Include a constructor to set the values for the & get the details of object :-Include toString().

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 String toString() {
 return "Book Details: In Name: " + name + " In
 Author: " + author + " In Price: Rs. " + price +
 " In Number of Pages: " + numPages;
 }

 public class BookTest {

 public static void main(String args[]) {
 if (n < 0) {
 Scanner scanner = new Scanner(System.in);
 System.out.print("Enter the number of books: ");
 ("Entered input");
 int n = scanner.nextInt();
 }
 System.out.print(" Enter details for book"
 +(i+1)+": ");
 System.out.print("Name: ");
 }
 }
}

```
String name = scanner.nextLine();
```

```
System.out.println("Author : ");
```

```
String author = scanner.nextLine();
```

```
System.out.print("Number of Pages: ");
```

```
int numPages = scanner.nextInt();
```

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

```
}
```

```
System.out.println("In Details of the Books: ");
```

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

```
System.out.println("In Book " + (i+1) + ":" + "
```

```
books[i]);
```

```
}
```

```
3
```

Output :- Enter the number of books: 2
Enter the details for Book 1:

Name: maths

Author: RD

Price: Rs.200

Number of Pages: 400

Enter details for Book 2:

Name: Science

Author: HC

Price: Rs.566

Number of Pages: 675

Details of the Books:

Book 1:

Book Details:

Name: math

Author : RN

Price : Rs. 200.0

Number of Pages : 400

Book 2:

Book Details:

Name: Science

Author : HC

Price : Rs. 566

Number of Pages : 675

Enter the number of books : -2

Invalid input

~~Run~~
~~117~~

OUTPUT

```
Windows PowerShell      X + | ^

i Windows Terminal can be set as the default terminal application in your settings. Open Settings

Enter the number of books: 2

Enter details for Book 1:
Name: maths
Author: rd
Price: Rs.200
Number of Pages: 400

Enter details for Book 2:
Name: gtfdh
Author: gfdhg
Price: Rs.566
Number of Pages: 675

Details of the Books:

Book 1:
Book Details:
Name: maths
Author: rd
Price: Rs.200.0
Number of Pages: 400

Book 2:
Book Details:
Name: gtfdh
Author: gfdhg
Price: Rs.566.0
Number of Pages: 675
PS C:\Users\BMSCE\Desktop\1BM22CS159> |
```

LABORATORY 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.

```
abstract class Shape {  
    int dim1, dim2;  
  
    abstract void printArea();  
}  
  
class Rectangle extends Shape {  
    Rectangle(int l, int b) {  
        dim1 = l;  
        dim2 = b;  
    }  
  
    void printArea() {  
        System.out.println("Area of Rectangle = " + (dim1 * dim2));  
    }  
}  
  
class Triangle extends Shape {  
    Triangle(int base, int height) {  
        dim1 = base;  
        dim2 = height;  
    }  
  
    void printArea() {  
        System.out.println("Area of Triangle = " + (0.5 * dim1 * dim2));  
    }  
}  
  
class Circle extends Shape {  
    Circle(int r) {  
        dim1 = r;  
    }  
  
    void printArea() {  
        System.out.println("Area of Circle = " + (3.142 * dim1 * dim1));  
    }  
}  
  
public class ShapeDemo {  
    public static void main(String[] args) {  
        Rectangle r = new Rectangle(10, 20);  
        Triangle t = new Triangle(10, 20);  
    }  
}
```

```
        Circle c = new Circle(20);

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

OUTPUT

```
PS C:\Users\BMSCE\Desktop\1BM22CS159> javac ShapeDemo.java
PS C:\Users\BMSCE\Desktop\1BM22CS159> java ShapeDemo
Area of Rectangle=200
Area of Triangle=100.0
Area of Circle=1256.8
PS C:\Users\BMSCE\Desktop\1BM22CS159> |
```

8-17-24

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.

→ abstract class Shape {

 int dim1, dim2;

 abstract void printArea();

}

class Rectangle extends Shape {

 Rectangle(int l, int b) {

 dim1 = l;

 dim2 = b;

}

 void printArea() {

 System.out.println("Area of Rectangle = "
 + (dim1 * dim2));

}

class Triangle extends Shape {

 Triangle(int l, int b) {

 dim1 = l; dim2 = b;

}

 void printArea() {

 System.out.println("Area of Triangle = "
 + (0.5 * dim1 * dim2));

}

```
class Circle extends Shape {  
    Circle (int r) {  
        dim1 = r;  
    }  
    void printArea () {  
        System.out.println ("Area of Circle = "  
            + (3.142 * dim1 * dim1));  
    }  
}
```

```
class ShapeDemo {  
    public static void main (String args[]) {  
        Rectangle r = new Rectangle (10, 20);  
        Triangle t = new Triangle (10, 20);  
        Circle c = new Circle (20);  
        r.printArea ();  
        t.printArea ();  
        c.printArea ();  
    }  
}
```

Output:-
Area of Rectangle = 200 in
Area of Triangle = 100.0
Area of Circle = 1256.8

~~Session~~

LABORATORY 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 customerName;
    int accountNumber;
    String accountType;
    double balance;

    public Account(String customerName, int accountNumber, String accountType) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = 0;
    }

    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposit of " + amount + " successful.");
    }

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

class SavingsAccount extends Account {
    double interestRate;

    public SavingsAccount(String customerName, int accountNumber) {

```

```

super(customerName, accountNumber, "Savings");
this.interestRate = 0.05; // 5% interest rate
}

public void depositInterest() {
    double interest = balance * interestRate;
    deposit(interest);
    System.out.println("Interest deposited: " + interest);
}

public void withdraw(double amount) {
    if (balance >= amount) {
        balance -= amount;
        System.out.println("Withdrawal of " + amount + " successful.");
    } else {
        System.out.println("Insufficient balance for withdrawal.");
    }
}

class CurrentAccount extends Account {
    double minimumBalance;
    double serviceCharge;

    public CurrentAccount(String customerName, int accountNumber) {
        super(customerName, accountNumber, "Current");
        this.minimumBalance = 1000; // Minimum balance required
        this.serviceCharge = 50; // Service charge if balance falls below minimum
    }

    public void withdraw(double amount) {
        if (balance - amount >= minimumBalance) {
            balance -= amount;
            System.out.println("Withdrawal of " + amount + " successful.");
        } else {
            System.out.println("Insufficient balance for withdrawal, Service charge of " + serviceCharge + " will be applied.");
            balance -= serviceCharge;
        }
    }
}

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

        System.out.println("Enter name for savings account: ");
        String savingsName = scanner.nextLine();
        System.out.println("Enter account number for savings account: ");
        int savingsNumber = scanner.nextInt();
        SavingsAccount savingsAccount = new SavingsAccount(savingsName, savingsNumber);

        System.out.println("Enter name for current account: ");
        scanner.nextLine(); // Consume newline
        String currentName = scanner.nextLine();
        System.out.println("Enter account number for current account: ");
    }
}

```

```

int currentNumber = scanner.nextInt();
CurrentAccount currentAccount = new CurrentAccount(currentName, currentNumber);

System.out.println("Enter amount to deposit into savings account: ");
double savingsDeposit = scanner.nextDouble();
savingsAccount.deposit(savingsDeposit);
savingsAccount.displayBalance();

System.out.println("Enter amount to deposit into current account: ");
double currentDeposit = scanner.nextDouble();
currentAccount.deposit(currentDeposit);
currentAccount.displayBalance();

System.out.println("Enter amount to withdraw from savings account: ");
double savingsWithdraw = scanner.nextDouble();
savingsAccount.withdraw(savingsWithdraw);
savingsAccount.displayBalance();

System.out.println("Enter amount to withdraw from current account: ");
double currentWithdraw = scanner.nextDouble();
currentAccount.withdraw(currentWithdraw);
currentAccount.displayBalance();
}

}

```

OUTPUT

```

PS C:\Users\BMSCE\Desktop\1BM22CS159> javac Bank.java
PS C:\Users\BMSCE\Desktop\1BM22CS159> java Bank
Enter name for savings account:
uzair
Enter account number for savings account:
123
Enter name for current account:
yaseen
Enter account number for current account:
2345
Enter amount to deposit into savings account:
1234
Deposit of 1234.0 successful.
Current balance: 1234.0
Enter amount to deposit into current account:
300
Deposit of 300.0 successful.
Current balance: 300.0
Enter amount to withdraw from savings account:
13
Withdrawal of 13.0 successful.
Current balance: 287.0
Enter amount to withdraw from current account:
100
Insufficient balance for withdrawal. Service charge of 50.0 will be applied.
Current balance: 250.0

```

19-2

5 Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called saving account & the other current account. The SA provides compound interest Eg withdrawal facilities but no cheque book facility. The Current account provides cheque book facility but no interest. Current account holder 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-act & Sav-act to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

- a) Accept deposit from customer & update the balance.
- b) Display the balance.
- c) Compute and deposit interest.
- d) Permit withdrawal and update the balance.

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

```
class Account
```

```
String customerName;
```

```
int accountNumber;
```

```
String accountType;
```

```
double balance;
```

```
public Account (String customerName, int  
accountNumber, String accountType) {  
this.customerName = customerName;  
this.accountNumber = accountNumber;  
this.accountType = accountType;  
this.balance = 0;
```

```
}  
public void deposit (double amount) {  
balance += amount;  
System.out.println ("Deposit " +  
amount + " successful.");
```

```
}  
public void displayBalance () {
```

```
System.out.println ("Current balance:  
" + balance);
```

```
}
```

```
class SavingsAccount extends Account
```

```
double interest; // balance * interestRate
```

```
public SavingsAccount (String customerName,  
int accountNumber)
```

```
super (customerName, accountNumber,  
"Savings");
```

```
}  
this.interest = 0.05;
```

```
System.out.println ("Interest deposited:");
```

```
public void depositInterest()  
{  
    double interestA = balance * interest;  
    deposit(interestA);  
    System.out.println("Interest deposited :"  
        + interestA);  
}
```

3

```
public void withdraw(double amount)  
{  
    if (balance >= amount){  
        balance -= amount;  
        System.out.println("Withdrawal "+  
            amount + " successful.");  
    } else  
        System.out.println("Insufficient  
            balance for withdrawal.");  
}
```

3

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

```
public CurrentAccount extends Account {  
    double minimumBalance;  
    double serviceCharge;
```

```
public CurrentAccount (String customerName,  
    int accountNumber,  
    this.minimumBalance = 1000;  
    this.serviceCharge = 50;
```

3

```
public void withdraw (double amount)  
{  
    if (balance - amount >= minimumBal  
        - ance){  
        balance -= amount;
```

System.out.println ("Withdrawal of "+ amount +
"Successful.");

}

else {

System.out.println ("Insufficient balance
for withdrawal. Service charge "+
+ "service charge "+ " will be applied
balance - = service charge;

}

}

public class Bank {

public static void main (String [] args) {
Scanner sc = new Scanner (System.in);
SavingsAccount sa = new SavingsAccount
("John Doe", 123456);

CurrentAccount ca = new CurrentAccount
("Uzair", 654321);

Saving sa.deposit (5000);

sa.displayBalance();

sa.depositInterest();

sa.displayBalance();

ca.deposit (3000);

ca.displayBalance();

ca.withdraw (2000);

ca.displayBalance();

}

}

O/P:-

Deposit of 5000.0 successful.

Current balance: 5000.0

Deposit of 250.0 successful.

Interest deposited: 250.0

Current balance: 5250.0

Deposit of 300.0 successful.

Current balance: 300.0

Insufficient balance for withdrawal.

Service charge of 50.0 will be applied

Current balance: 250.0

Our
19/2/14

LABORATORY PROGRAM - 6

Create a package CIE which has two classes- Student and Internals. The class Student has members like usn, name, sem. The class Internals which is a derived class of Student and 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.

```
// CIE/Student.java
package CIE;

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

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

// CIE/Internals.java
package CIE;

public class Internals extends Student {
    protected int[] internalMarks;

    public Internals(String usn, String name, int sem, int[] internalMarks) {
        super(usn, name, sem);
        this.internalMarks = internalMarks;
    }
}
```

```

// SEE/External.java
package SEE;
import CIE.Student;

public class External extends Student {
    protected int[] seeMarks;

    public External(String usn, String name, int sem, int[] seeMarks) {
        super(usn, name, sem);
        this.seeMarks = seeMarks;
    }
}

// MainFile.java
import CIE.Internals;
import SEE.External;

public class MainFile {
    public static void main(String[] args) {
        // Example data
        Internals studentCIE = new Internals("123", "John", 5, new int[]{80, 75, 90, 85, 88});
        External studentSEE = new External("123", "John", 5, new int[]{85, 78, 92, 88, 90});

        // Displaying final marks
        displayFinalMarks(studentCIE, studentSEE);
    }

    public static void displayFinalMarks(Internals internals, External external) {
        System.out.println("Student Details:");
        System.out.println("USN: " + internals.usn);
        System.out.println("Name: " + internals.name);
        System.out.println("Semester: " + internals.sem);

        System.out.println("\nFinal Marks:");
        for (int i = 0; i < internals.internalMarks.length; i++) {
            int finalMarks = internals.internalMarks[i] + external.seeMarks[i];
            System.out.println("Course " + (i + 1) + ": " + finalMarks);
        }
    }
}

```

OUTPUT

```

PS C:\Users\muzai\OneDrive\Desktop> javac CIE/*.java SEE/*.java result/*.java
PS C:\Users\muzai\OneDrive\Desktop> java result/test
usn:1BM22CS159 name: Uzair sem: 3
internal marks:
internal marks 1 : 43.000000internal marks 2 : 45.000000internal marks 3 : 47.000000internal marks 4 : 44.000000internal
marks 5 : 41.000000
external marks:
external marks 1 : 90.000000external marks 2 : 87.000000external marks 3 : 65.000000external marks 4 : 98.000000external
marks 5 : 43.000000

```

6. Create a package CIE which has two classes - Student & internals. The class internal marks has an array that stores the internal marks scored in five courses of the current sem 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 sem 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 Student {
    public String un;
    public String name;
    public int sem;
    public Student (String th, String nm, int s) {
        this.un = th;
        this.name = nm;
        this.sem = s;
    }
}
```

```
package CIE;
public class internals extends (IE.Student) {
    public double marks[];
    public internals (String u, String n,
                     int s, double m[]) {
        super (u, n, s);
        +
        marks = m;
    }
}
```

this. imarks = m;

3 5

package SEE;

import CIE.Student;

public class Externals extends CIE.Student {

public double smarks[];

public Externals (String u, String n, int s, double m[]) {

Super(u, n, s);

this. smarks = m;

3

package result;

import CIE.Student;

import CIE.internals;

import SEE.externals;

public class Test {

public static void main (String args[]) {

double internal[] = {43, 45, 47, 44, 41};

double external[] = {90, 87, 65, 98, 43};

Student s1 = new Student ("IBM22CS159", "Uzair", 3, internal);

internals i1 = new internals ("IBM22CS159", "Uzair", 3, internal);

externals e1 = new externals ("IBM22CS159", "Uzair", 3, external);

System.out.println ("Utn: " + s1.usn + " name: " + s1.name + " Sem: " + s1.sem);

```

System.out.println("internal marks:");
for(int i = 0; i < 5; i++) {
    System.out.printf ("internal marks %d : %.f", i + 1, s1.internalMarks[i]);
}
}

System.out.println();
System.out.println("external marks : ");
for(int i = 0; i < 5; i++) {
    System.out.printf ("external marks %d : %.f", i + 1, e1.externalMarks[i]);
}
}

User : 1BM22-CS159 name: Uzair sem: 3
internal marks:
internal marks 1: 43 internal marks 2: 45. internal marks
47.000 internal marks 4: 44.
internal marks 5: 41.

external marks:
external marks 1: 90 external marks 2: 87
external marks 3: 65 external marks 4: 98
external marks 5: 43.

```

LABORATORY 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 MyException extends Exception {  
    int detail;  
    public MyException(int age, String exe) {  
        this.detail=age;  
        System.out.println(exe + " given age is: " + age + " please enter again");  
    }  
    public String getMessage() {  
        return "Exception: " + detail;  
    }  
}  
class Father{  
    int age;  
    public Father(int age) throws MyException {  
        if (age < 0) throw new MyException(age , "Age cannot be lesser than 0");  
        this.age = age ;  
    }  
}  
class Son extends Father{  
    int age;  
    public Son(int fatherAge, int sonAge) throws MyException{  
        super(fatherAge);  
        this.age = sonAge;  
        if (this.age > super.age) throw new MyException(age , "Age of son cannot be more  
than father");  
    }  
}  
public class father_son {  
    public static void main(String[] args) {  
        try {  
            Father f1 = new Father(-1);  
            Son s1 = new Son(30 , 31);  
        }  
        catch (MyException e) {  
            System.out.println("Exception caught: " + e.getMessage());  
        }  
    }  
}
```

```
PS C:\Users\muzai\OneDrive\Desktop\java> java father_son  
Age cannot be lesser than 0 given age is: -1 please enter again  
Exception caught: Exception: -1
```

7. Write a program that demonstrates handling of exceptions in C++ inheritance tree.
 Create a base class "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 that takes both father & sons age & throws an exception if Son's age is > = father's age.

```
class MyException : public Exception {
    int detail;
public:
    MyException(int age, string ex) {
        this.detail = age;
        cout << ex << " given age is: " << age << " please enter again ";
    }
}
```

```
public string getMessage() {
    return "Exception: " + detail;
}
```

~~Class Father {~~

```
int age;
public Father(int age) throws MyException {
    if (age < 0) throw new MyException(
        age, "Age cannot be lesser than 0");
    this.age = age;
}
```

~~Class Son extends Father {~~

```
int age;
```

```
public Son(int fatherAge, int sonAge)
throws MyException {
    Super(fatherAge);
    this.age = sonAge;
    if (this.age > Super.age) throw
        new MyException("Age of son
            cannot be more than father");
```

{ }
3

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

}{

Father f = new Father(-1);

Son s1 = new Son(30, 31);

}

catch (MyException e) {

System.out.println ("Exception
caught: " + e.getMessage());

}

Output:

Age of son cannot be more than
Father. given age is 31 please enter
again

Exception caught: Exception: 31,

R/1/18/20

LABORATORY 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 NewThread implements Runnable{
    Thread t;
    NewThread(){
        t = new Thread(this , "NThread");
        System.out.println("Child thread: "+t);
        t.start();
    }
    public void run() {
        try {
            for(int n=5 ;n>0 ; n--) {
                System.out.println("CSE:"+n);
                Thread.sleep(2000);
            }
        } catch (InterruptedException ie) {
            System.out.println("main thread interrupted");
        }
    }
}
public class Thread2 {
    public static void main(String[] args) {
        new NewThread();
        System.out.println("Back in Main");
        try {
            for(int n=5 ; n>0 ; n--) {
                System.out.println("BMSCE:"+n);
                Thread.sleep(10000);
            }
        } catch (InterruptedException ie) {
            System.out.println("main thread interrupted");
        }
    }
}
```

OUTPUT

```
PS C:\Users\BMSCE\Desktop\1BM22CS159> javac Thread2.java
PS C:\Users\BMSCE\Desktop\1BM22CS159> java Thread2
CT:Thread[#20,NThread,5,main]
Back in main
CSE:5
BMSCE:3
CSE:4
CSE:3
CSE:2
CSE:1
BMSCE:2
Child Thread quitting
BMSCE:1
Main Thread quitting
```

5/2

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

```
class NewThread implements Runnable
{
    Thread t;
    NewThread()
    {
        t = new Thread(this, "NThread");
        System.out.println("C7 :" + t);
        t.start();
    }
    public void run()
    {
        try
        {
            for(int n=5; n>0; n--)
            {
                System.out.println("CSE :: " + n);
                Thread.sleep(2000);
            }
        }
        catch(InterruptedException e)
        {
            System.out.println("Child Thread Interrupted");
            System.out.println("Child Thread quitting");
        }
    }
}
class Thread2
{
    public static void main(String ss[])
    {
        new NewThread();
        System.out.println("Back in main");
    }
}
```

try

for (int n=3; n>0; n--)

System.out.println("BMSCE"+n);

Thread.sleep(10000);

}

}

catch (InterruptedException e)

{

System.out.println("Main Thread interrupted");

}

}

C:\> Thread [#20, NThread, S, main]

Back in main

CSE:5

BMSCE:3

CSE:4

CSE:3

CSE:2

CSE:1

BMSCE:2

Child Thread quitting

BMSCE:1

Main Thread quitting

SP/1202

LABORATORY 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 DivisionMain1 extends Frame implements ActionListener
{
    TextField num1,num2;
    Button dResult;
    Label outResult;
    String out="";
    double resultNum;
    int flag=0;
    public DivisionMain1()
    {
        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)
{
    int n1,n2;
    try
    {
        if (ae.getSource() == dResult)
        {
            n1=Integer.parseInt(num1.getText());
            n2=Integer.parseInt(num2.getText());

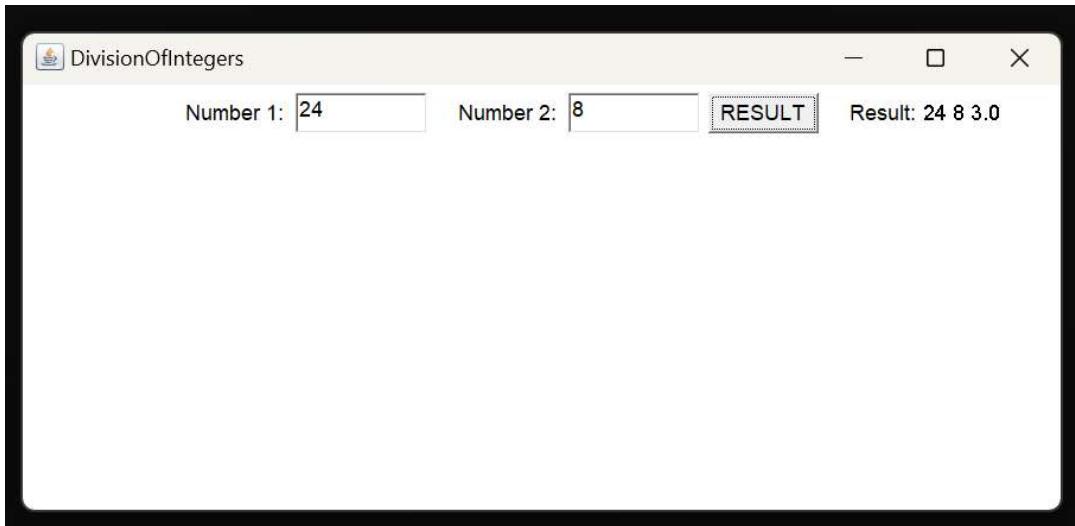
            /*if(n2==0)
                throw new ArithmeticException();*/
            out=n1+" "+n2+" ";
            resultNum=n1/n2;
            out+=String.valueOf(resultNum);
            repaint();
        }
    }
    catch(NumberFormatException e1)
    {
        flag=1;
        out="Number Format Exception! "+e1;
        repaint();
    }
    catch(ArithmeticException e2)
    {
        flag=1;
        out="Divide by 0 Exception! "+e2;
        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)
{
    DivisionMain1 dm=new DivisionMain1();
    dm.setSize(new Dimension(800,400));
    dm.setTitle("DivisionOfIntegers");
    dm.setVisible(true);
}
```

OUTPUT



9 Create Java UI for Integer Division.
Handle Exceptions & Display results.

9. import java.awt.*
import java.awt.event.*;

public class DivisionMain1 extends Frame
implements ActionListener

{
 TextField num1, num2;
 Button dResult;
 Label outResult;
 String out = "";
 double resultNum;
 int flag = 0;

 public DivisionMain1()
 {

 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);
 outResult = new Label("Result:",
 Label.RIGHT);
 add(number1);
 add(num1);
 add(number2);
 add(num2);
 add(dResult);
 add(outResult);

```
num1.addActionListener(this);  
num2.addActionListener(this);  
cResult.addActionListener(this);  
addWindowListener(new WindowAdapter()
```

```
}  
public void windowClosing(WindowEvent we)
```

```
}  
System.exit(0);
```

```
});
```

```
}  
public void actionPerformed(ActionEvent ae)
```

```
int n1, n2;
```

```
try
```

```
if(ae.getSource() == cResult)
```

```
n1 = Integer.parseInt(num1.getText());
```

```
n2 = Integer.parseInt(num2.getText());
```

```
out = n1 + " * " + n2 + " = "
```

```
resultNum = n1 / n2;
```

```
out += String.valueOf(resultNum);
```

```
repaint();
```

```
}
```

```
}
```

```
Catch(NumberFormatException e1)
```

```
{ flag = 1;
```

```
out = "Number Format Exception! " + e1;
```

```
repaint();
```

```
}
```

```
Catch(ArithmeticException e2)
```

```
{ flag = 1; out = "Divide by 0 Exception! "
```

```
+ e2; 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)
```

```

Division Main1 dm = new DivisionMain1();
dm.setSize(new Dimension(800, 400));
dm.setTitle("Division Of Integers");
dm.setVisible(true);

```

Output:

Division Of Integers				-	X
Number1:	24	Number2:	8	RESULT	Result: 24 8 30

Laboratory - 10

Report

The given program utilize Java's AWT and Swing libraries to create GUI Applications. These programs showcase various event handling in Java.

i) ButtonDemo: It is an applet that demonstrates event handling in Java AWT. It contains three buttons labeled "Yes", "No" & "Undecided". Clicking on each button triggers an action event, and the corresponding message is displayed on the applet.

ii) Button List: It is another frame-based Java application that demonstrates event handling and consists of three buttons similar to ButtonDemo program. Clicking on any button updates a message indicating the button pressed.

iii) buttondrag: It is a frame-based Java app that implements a puzzle game. Here, players rearrange numbered buttons in ascending order by swapping their position.

iv) DivisionMain: It is a frame-based Java app that allows users to input 2 numbers and calculate their division. It includes error handling for

Scenarios such as division by zero & invalid input formats

v) Division Main1: It's another frame-based Java app that performs division operations similar to DivisionMain. However, it both handles exceptions in a different order compared to DivisionMain.

vi) TextField Demo: Demonstrates the usage of the text fields in Java AWT. It provides a simple GUI interface where users can input their name & password. Upon pressing Enter in either text field, the program repaints the window to display the entered name & password.

vii) ButtonList1: It is a frame-based Java app that demonstrates event handling & dialog creation. Clicking on any of the buttons ("Yes", "No", "Undecided") opens a dialog window displaying the button label.

Report on Mouse Events Demo Program.

This showcases the implementation of mouse event handling in Java. It provides a simple graphical UI, where user can interact with the mouse & the program responds to various mouse events.

In conclusion, this program, through its intuitive interface & dynamic feedback user can understand & interact with different mouse actions.

Ran
20/12/20