

B.M.S. COLLEGE OF ENGINEERING

Basavanagudi, Bengaluru- 560019

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



LABORATORY RECORD

On

*“Object-Oriented Java Programming
(22CS3PCOOJ)”*

Submitted By :

K. ADITYA VARMA

1BM22CS120

In partial fulfilment of

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

2023-24

INDEX

1	18.12.2023	Laboratory Program – 1
2	01.01.2024	Laboratory Program – 2
3	01.01.2024	Laboratory Program – 3
4	08.01.2024	Laboratory Program – 4
5	08.01.2024	Laboratory Program – 5
6	22.01.2024	Laboratory Program – 6
7	22.01.2024	Laboratory Program – 7
8	22.01.2024	Laboratory Program – 8
9	19.02.2024	Laboratory Program – 9 (And Report on few AWT program)

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.

Program:

```
import java.util.Scanner;  
import java.lang.Math;  
  
class disc  
{  
    public static void main(String xx[])  
    {  
        System.out.println("Enter values of a,b,c");  
        Scanner s = new Scanner(System.in);  
        int a = s.nextInt();  
        int b = s.nextInt();  
        int c = s.nextInt();  
        int D = (b*b) - (4*a*c);  
        if (a==0 || b==0 || c==0)  
        {  
            System.out.println("Invalid coefficients");  
        }  
        else if(D>0)  
        {  
            System.out.println("It has two distinct real roots");  
            double r1 = (-b+Math.sqrt(D))/(2*a);  
            double r2 = (-b-Math.sqrt(D))/(2*a);  
            System.out.println("Roots are " + r1 + " and " + r2);  
        }  
        else if(D==0)  
        {  
            System.out.println("It has one real root");  
            double r = -b/(2*a);  
            System.out.println("Root is " + r);  
        }  
        else  
        {  
            System.out.println("It has no real roots");  
        }  
    }  
}
```

```
double 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 one real root");
    double r1 = -b/(2*a);
    System.out.println("The root is "+r1);
}

else if(D<0)
{
    System.out.println("It has real and imaginary roots ");
    double r1 = -b/(2*a);
    double r2 = (Math.sqrt(Math.abs(D)))/(2*a);
    System.out.println("The roots are " +r1+ " and " +r2);
}

}
```

OUTPUT:

```
C:\Users\k Aditya Varma\Desktop\MY JAVA>java disc
Enter values of a,b,c
1
3
5
It has real and imaginary roots
The roots are -1.0 and 1.6583123951777
```

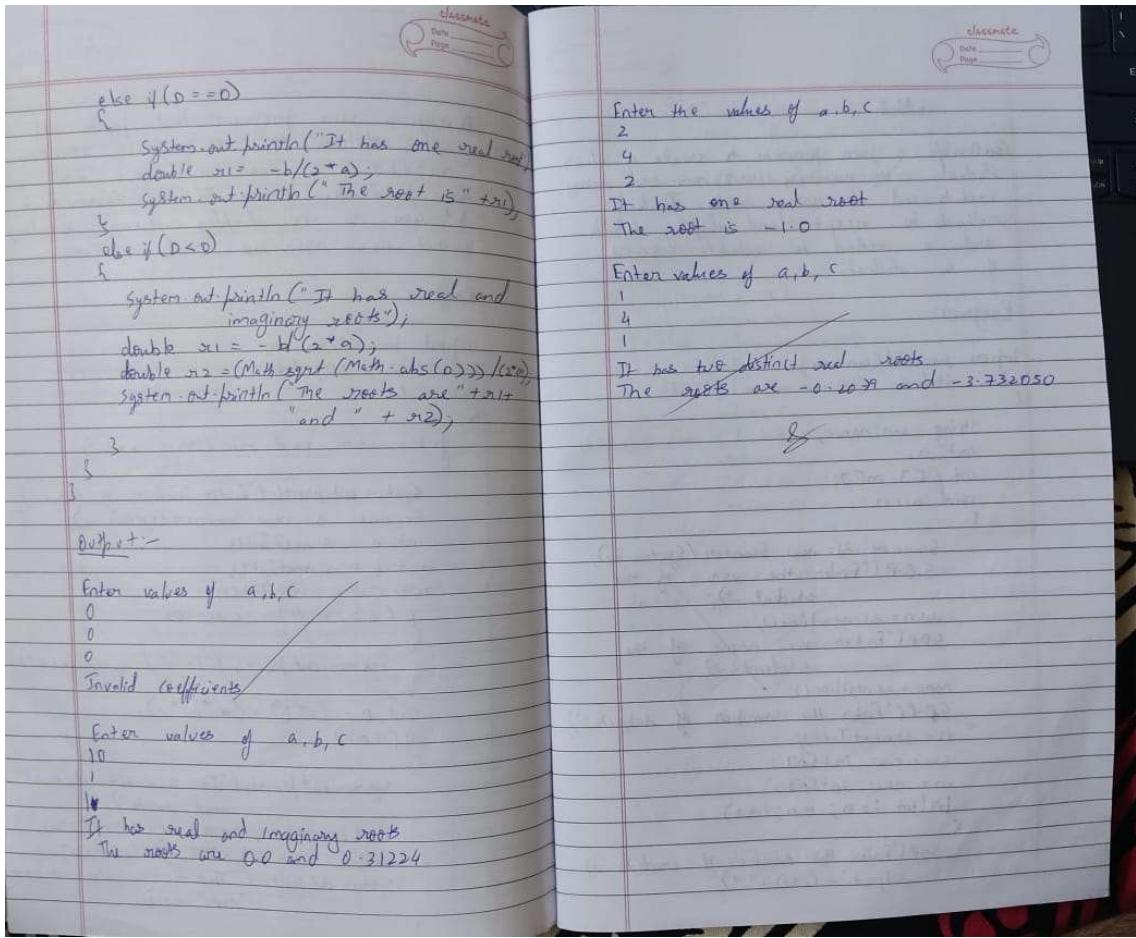
```
C:\Users\k Aditya Varma\Desktop\MY JAVA>java disc
Enter values of a,b,c
6
8
0
Invalid coefficients
```

```
C:\Users\k Aditya Varma\Desktop\MY JAVA>java disc
Enter values of a,b,c
23
34
67
It has real and imaginary roots
The roots are 0.0 and 1.5384179141699246
```

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 -ve, display message stating that there are no real solutions.

Program :-

```
import java.util.Scanner;
import java.lang.Math;
class disc
{
    public static void main (String xx[])
    {
        System.out.print ("Enter values of a,b,c ");
        Scanner s = new Scanner (System.in);
        int a = s.nextInt();
        int b = s.nextInt();
        int c = s.nextInt();
        if (a == 0 || b == 0 || c == 0)
        {
            System.out.println ("Invalid coefficients");
        }
        int D = (b * b) - (4 * a * c);
        if (D > 0)
        {
            System.out.println ("It has two distinct
real roots");
            double r1 = (-b + Math.sqrt(D)) / (2 * a);
            double r2 = (-b - Math.sqrt(D)) / (2 * a);
            System.out.println ("The roots are " + r1 +
" and " + r2);
        }
    }
}
```



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
{
    String usn;
    String name;
    int credits[];
    int marks[];
    int n;
    void acc()
    {
        Scanner s1 = new Scanner(System.in);
        System.out.print("Enter USN: ");
        usn = s1.nextLine();
        System.out.print("Enter Name: ");
        name = s1.nextLine();
        System.out.println("Enter the number of subjects : ");
        n = s1.nextInt();
        credits = new int[n];
        marks = new int[n];
        for (int i = 0; i < n; i++) {
            System.out.print("Enter credits for subject " + (i + 1) + ": ");
            credits[i] = s1.nextInt();
        }
    }
}
```

```
        System.out.print("Enter marks for subject " + (i + 1) + ": ");
        marks[i] = s1.nextInt();
    }
}

void disp()
{
    System.out.println("USN: " + usn);
    System.out.println("Name: " + name);
    System.out.println("Credits:");
    for (int i = 0; i < n; i++)
    {
        System.out.println(" Subject " + (i + 1) + ": " + credits[i]);
    }
    System.out.println("Marks:");
    for (int i = 0; i < n; i++)
    {
        System.out.println(" Subject " + (i + 1) + ": " + marks[i]);
    }
}
double sgpa()
{
    int totalCredits = 0;
    double weightedSum = 0;

    for (int i = 0; i < n; i++)
    {
        totalCredits += credits[i];
        weightedSum += calculateGradePoint(marks[i]) * credits[i];
    }
}
```

```
    }

    return weightedSum / totalCredits;
}

double calculateGradePoint(int marks)

{
    if (marks >= 90) return 10.0;

    else if (marks >= 80) return 9.0;

    else if (marks >= 70) return 8.0;

    else if (marks >= 60) return 7.0;

    else if (marks >= 50) return 6.0;

    else return 0.0;
}

}

class Main

{
    public static void main(String[] args)

    {
        Student s = new Student();

        s.acc();

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

        s.disp();

        double sgpa = s.sgpa();

        System.out.println("\nSGPA: " + sgpa);
    }
}
```

OUTPUT:

```
Enter USN: 120
Enter Name: Aditya
Enter the number of subjects :
2
Enter credits for subject 1: 1
Enter marks for subject 1: 23
Enter credits for subject 2: 4
Enter marks for subject 2: 99

Student Details:
USN: 120
Name: Aditya
Credits:
    Subject 1: 1
    Subject 2: 4
Marks:
    Subject 1: 23
    Subject 2: 99

SGPA: 8.0
```

2) Create a package CIF which has two classes - Student and Internals. The class Student has members like USN, name, etc. personal has members like The class Internals has an array that stores 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 a derived class of Student. This class has an array that stores the SEC marks scored in 5 courses of current sem of the student. Inherit the two packages in a file main that declares the final marks of n student in all five courses.

Program:-

```
package CIF;
public class Student {
    int usn;
    String name;
    int sem;
    public void accept() {
        Scanner s1 = new Scanner(System.in);
        S.O.P("Enter name : ");
        name = s1.nextLine();
        S.O.P("Enter usn : ");
        usn = s1.nextInt();
    }
}
```

```
S.O.P("Enter sem : ");
sem = s.nextInt();
```

3 void display()

```
S.O.P("USN = " + usn);
S.O.P("Name = " + name);
S.O.P("Sem = " + sem);
```

3

```
package CIF;
import java.util.Scanner;
class Internals extends Student
```

```
{ public int marks[] = new int[5];
void acceptMarks() {
```

```
Scanner s1 = new Scanner(System.in);
for (int i = 0; i < 5; i++)
    marks[i] = s1.nextInt();
3
```

void displayMarks()

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

```
S.O.P("Marks in CIF subject" + (i + 1) + ":" + marks[i]);
3
```

```
package SEE;
import CIF.Student;
import java.util.Scanner;
```

```
public class External extends CIF.Student
```

```
{ int secMarks[] = new int[5];
public void secAccept() {
    Scanner s1 = new Scanner(System.in);
    for (int i = 0; i < 5; i++)
        secMarks[i] = s1.nextInt();
3
```

void secDisplay()

```
{ for (int i = 0; i < 5; i++)
    S.O.P("Marks in SEE subject" + (i + 1) + ":" + secMarks[i]);
3
```

import java.util.Scanner;

```
import CIF.Student;
import CIF.Internals;
import CIF.External;
```

public class Data

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

```
int n;
Scanner s1 = new Scanner(System.in);
S.O.P("Enter no of students : ");
n = s1.nextInt();
3
```

```
CIF.Student data = new CIF.Student[n];
3
```

```
CIF.Internals marks[] = new CIF.Internals[n];
3
```

```
SEE.External smarks[] = new SEE.External[n];
3
```

```
for (int i = 0; i < n; i++)
3
```

```
data[i] = new CIF.Student();
marks[i] = new CIF.Internals();
smarks[i] = new SEE.External();
S.O.P("Enter details : ");
data[i].accept();
3
```

```
S.O.P("Enter CIF marks : ");
marks[i].acceptMarks();
3
```

```
S.O.P("Enter SEE marks : ");
smarks[i].secAccept();
3
```

```
S.O.P("Student Details : ");
for (int j = 0; j < n; j++)
3
```

```
data[i].display();
marks[i].displayMarks();
smarks[i].secDisplay();
3
```

```

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

public class External extends CIE.student
{
    int see_marks[] = new int[5];
    public void seeMarks()
    {
        Scanner si = new Scanner(System.in);
        for(int i=0; i<5; i++)
        {
            see_marks[i] = si.nextInt();
        }
    }

    void seeDisplay()
    {
        for(int i=0; i<5; i++)
        {
            System.out.println("Marks in SEE subjects " + (i+1) + " : " + see_marks[i]);
        }
    }

    import java.util.Scanner;
    import CIE.student;
    import CIE.Internal;
    import CIE.External;
}

public class Data
{
    public static void main(String args[])
    {
        int n;
        Scanner si = new Scanner(System.in);
        System.out.println("Enter no of students : ");
        n = si.nextInt();

        CIE.student data[] = new CIE.student[n];
        CIE.Internal marks[] = new CIE.Internal[n];
        SEE.External smarks[] = new SEE.External[n];

        for(int i=0; i<n; i++)
        {
            data[i] = new CIE.student();
            marks[i] = new CIE.Internal();
            smarks[i] = new SEE.External();
            data[i].accept();
            marks[i].accept();
            smarks[i].accept();
        }

        System.out.println("Student Details : ");
        for(int j=0; j<n; j++)
        {
            data[j].display();
            marks[j].display();
            smarks[j].seeDisplay();
        }
    }
}

```

Output :-

Enter no of students : 1
 Enter student details:
 Enter name: Aditya
 Enter vsn: 120
 Enter sem: 3

Enter CIE marks:
 40
 31
 29
 35
 30

Enter SEE marks:
 88
 61
 48
 91
 99

Student Details:

Aditya
 USN: 120
 Name: Aditya
 Sem: 3

Marks in CIE subject 1 : 40
 Marks in CIE subject 2 : 31
 Marks in SEE subject 3 : 29
 Marks in SEE subject 4 : 35
 Marks in SEE subject 5 : 30

86
 29/01/2023

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 {  
    Scanner s1 = new Scanner(System.in);  
  
    String name, author;  
    int price, num_pages;  
  
    Book(String name, String author, int price, int num_pages) {  
        this.name = name;  
        this.author = author;  
        this.price = price;  
        this.num_pages = num_pages;  
    }  
  
    Book() {}  
  
    void accept() {  
        System.out.println("Enter the name : ");  
        name = s1.nextLine();  
        System.out.println("Enter the name of the author : ");  
        author = s1.nextLine();  
        System.out.println("Enter the price of the book : ");  
        price = s1.nextInt();  
        System.out.println("Enter the number of pages : ");  
        num_pages = s1.nextInt();  
    }  
}
```

```
}

public String toString() {
    return "Book Details : \n name : " + name + "\n author : " + author + "\n Price : " + price
        + "\n Number of pages : " + num_pages;
}

public static void main(String args[]) {
    int n;
    Scanner s = new Scanner(System.in);
    System.out.println("Enter the number of entries : ");
    n = s.nextInt();
    Book b[] = new Book[n];
    for (int i = 0; i < n; i++) {
        b[i] = new Book();
        b[i].accept();
    }
    for (int i = 0; i < n; i++) {
        System.out.println(b[i].toString());
    }
}
```

OUTPUT:

```

Enter the number of entries :
1
Enter the name :
Aditya
Enter the name of the author :
Jeevan
Enter the price of the book :
150
Enter the number of pages :
50
Book Details :
name : Aditya
author : Jeevan
Price : 150
Number of pages : 50

```

LAB-3

Q Date
Page

S.O.P("Enter the name of the author");
author = s1.nextLine();
S.O.P("Enter the price of book : ");
price = s1.nextInt();
S.O.P("Enter the number of pages");
num.pages = s1.nextInt();

public String toString()
{
 return "Book Details :\nName " + name +
 "\nAuthor : " + author + "\nPrice : " + price +
 "\nNumber of pages : " + num.pages;
}

public static void main(String args[]){
 int n;
 Scanner s1 = new Scanner(System.in);
 S.O.P("Enter the number of entries");
 n = s1.nextInt();
 Book b[] = new Book[n];
 for(int i=0; i<n; i++)
 {
 b[i] = new Book();
 b[i].accept();
 }
 for(int i=0; i<n; i++)
 {
 S.O.P(b[i]);
 }
}

import java.util.Scanner;
class Book
{
 Scanner s1 = new Scanner(System.in)
 String name, author;
 int price, num.pages;
 Book(String name, String author, int price,
 int num.page)
 {
 this.name = name;
 this.author = author;
 this.price = price;
 this.num.pages = num.pages;
 }
 Book()
 {
 }
 void accept()
 {
 S.O.P("Enter the name : ");
 name = s1.nextLine();
 }
}

Enter the number of entries :

1

Enter the name : Aditya

Enter the name of the author : Jeewan

Enter the price of the Book :

253

Enter the number of pages :

425

Book Details :

Name : Aditya

Author : Jeewan

Price : 253

Number of pages : 425

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 d1;
```

```
    int d2;
```

```
    abstract void printArea();
```

```
}
```

```
class Rectangle extends Shape
```

```
{
```

```
    Rectangle(int x, int y)
```

```
{
```

```
    d1 = x;
```

```
    d2 = y;
```

```
}
```

```
    void printArea()
```

```
{
```

```
        System.out.println("Area of the rectangle: " + (d1*d2));
```

```
}
```

```
}
```

```
class Triangle extends Shape
```

```
{  
    Triangle(int x, int y)  
    {  
        d1 = x;  
        d2 = y;  
    }  
    void printArea()  
    {  
        System.out.println("Area of the Triangle: " + (0.5*d1*d2));  
    }  
}  
class Circle extends Shape  
{  
    Circle(int x)  
    {  
        d1 = x;  
  
    }  
    void printArea()  
    {  
        System.out.println("Area of the Circle: " + (3.14*d1*d1));  
    }  
}
```

```
class Main
{
    public static void main (String[] args) {
        Rectangle r1 = new Rectangle(5,3);
        Triangle t1 = new Triangle(4,8);
        Circle c1 = new Circle(5);

        r1.printArea();
        t1.printArea();
        c1.printArea();
    }
}
```

OUTPUT

```
Area of the rectangle: 15
Area of the Triangle: 16.0
Area of the Circle: 78.5
```

CAB-3

classmate

Date _____
Page _____

2) Abstract class Program

abstract class Shape

{

 int d1;

 int d2;

 abstract void printArea();

}

class Rectangle extends Shape

{

 Rectangle(int x, int y)

 {

 d1 = x;

 d2 = y;

 }

 void printArea()

{

 S.O.P ("Area of the rectangle " +
 (d1 * d2));

}

class Triangle extends Shape

{

 Triangle(int x, int y)

{

 d1 = x;

 d2 = y;

}

 void printArea()

{

 S.O.P ("Area of the Triangle: " +
 (0.5 * d1 * d2));

}

{

class circle extends Shape

{

 circle(int x)

{

 d1=x;

}

 void printArea()

{

 S.O.P("Area of the circle :"
 + (3.14 * d1 * d2));

}

}

class Main

{

 public static void main(String args[])

{

 Rectangle r1 = new Rectangle(5,3),

 Triangle t1 = new Triangle(4,3),

 Circle c1 = new Circle(5);

 r1.printArea();

 t1.printArea();

 c1.printArea();

}

}

Output :-

Area of the rectangle : 15.0

Area of the Triangle : 16.0

Area of the Circle : 78.5

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;
    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("Deposit successful. Updated balance: " + balance);
    }
}
```

```
public void displayBalance() {  
    System.out.println("Account Number: " + accountNumber);  
    System.out.println("Customer Name: " + customerName);  
    System.out.println("Account Type: " + accountType);  
    System.out.println("Balance: " + balance);  
}  
}  
  
class SavAcct extends Account {  
    public SavAcct(String customerName, long accountNumber, double balance) {  
        super(customerName, accountNumber, "Savings", balance);  
    }  
  
    public void computeAndDepositInterest(double rate) {  
        double interest = balance * rate / 100;  
        balance += interest;  
        System.out.println("Interest computed and deposited. Updated balance: "  
            + balance);  
    }  
  
    public void withdraw(double amount) {  
        if (amount <= balance) {  
            balance -= amount;  
            System.out.println("Withdrawal successful. Updated balance: " +  
                balance);  
        } else {  
    }
```

```
        System.out.println("Insufficient funds. Withdrawal failed.");
    }
}

}

class CurrAcct extends Account {
    double minimumBalance;
    double serviceCharge;

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

    private void checkMinimumBalance() {
        if (balance < minimumBalance) {
            balance -= serviceCharge;
            System.out.println("Minimum balance not maintained. Service charge
imposed. Updated balance: " + balance);
        }
    }

    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
        }
    }
}
```

```
        System.out.println("Withdrawal successful. Updated balance: " +  
balance);  
  
        checkMinimumBalance();  
  
    } else {  
  
        System.out.println("Insufficient funds. Withdrawal failed.");  
  
    }  
}  
  
}  
  
public class Bank {  
  
    public static void main(String[] args) {  
  
        Scanner s1 = new Scanner(System.in);  
  
        System.out.print("Enter customer name for Savings Account: ");  
        String SCN = s1.nextLine();  
        System.out.print("Enter account number for Savings Account: ");  
        long SAN = s1.nextLong();  
        System.out.print("Enter initial balance for Savings Account: ");  
        double SIB = s1.nextDouble();  
        SavAcct SA = new SavAcct(SCN, SAN, SIB);  
  
        System.out.print("Enter customer name for Current Account: ");  
        String CCN = s1.next();  
        System.out.print("Enter account number for Current Account: ");  
        long CAN = s1.nextLong();  
        System.out.print("Enter initial balance for Current Account: ");  
        double CIB = s1.nextDouble();
```

```
System.out.print("Enter minimum balance for Current Account: ");
double MB = s1.nextDouble();
System.out.print("Enter service charge for Current Account: ");
double SC = s1.nextDouble();
CurrAcct CA = new CurrAcct(CCN, CAN, CIB, MB, SC);
```

```
System.out.print("Enter deposit amount for Savings Account: ");
double SDA = s1.nextDouble();
SA.deposit(SDA);
```

```
System.out.print("Enter interest rate for Savings Account: ");
double SIR = s1.nextDouble();
SA.computeAndDepositInterest(SIR);
```

```
System.out.print("Enter withdrawal amount for Savings Account: ");
double SWA = s1.nextDouble();
SA.withdraw(SWA);
```

```
System.out.print("Enter deposit amount for Current Account: ");
double CDA = s1.nextDouble();
CA.deposit(CDA);
```

```
System.out.print("Enter withdrawal amount for Current Account: ");
double CWA = s1.nextDouble();
CA.withdraw(CWA);
```

```

        System.out.println("\nFinal Balances:");
        System.out.println("Savings Account:");
        SA.displayBalance();
        System.out.println("\nCurrent Account:");
        CA.displayBalance();
    }
}

```

OUTPUT:

```

Enter customer name for Savings Account: Aditya
Enter account number for Savings Account: 123
Enter initial balance for Savings Account: 150
Enter customer name for Current Account: Aditya
Enter account number for Current Account: 123
Enter initial balance for Current Account: 450
Enter minimum balance for Current Account: 345
Enter service charge for Current Account: 56
Enter deposit amount for Savings Account: 76
Deposit successful. Updated balance: 226.0
Enter interest rate for Savings Account: 32
Interest computed and deposited. Updated balance: 298.32
Enter withdrawal amount for Savings Account: 100
Withdrawal successful. Updated balance: 198.32
Enter deposit amount for Current Account: 50
Deposit successful. Updated balance: 500.0
Enter withdrawal amount for Current Account: 670
Insufficient funds. Withdrawal failed.

Final Balances:
Savings Account:
Account Number: 123
Customer Name: Aditya
Account Type: Savings
Balance: 198.32

Current Account:
Account Number: 123
Customer Name: Aditya
Account Type: Current
Balance: 500.0

C:\Users\k Aditya Varma\Desktop\MY JAVA>

```

classmate
Date _____
Page _____

Lab Program - 5

Bank Account problem

```

import java.util.Scanner;
class Account {
    String custName;
    double accNum;
    String accType;
    double balance;
    Account(String name, double num, String type) {
        custName = name;
        accNum = num;
        accType = type;
        this.balance = 0.0;
    }
    public void deposit(double amount) {
        balance = balance + amount;
        System.out.println("Deposit successful");
        System.out.println("Balance : " + balance);
    }
    public void dispBalance() {
        System.out.println("Account Balance : " + balance);
    }
}

```

class SavingsAcc extends Account

```

double interestRate;
SavingsAcc(String name, int num) {
    super(name, num, "savings");
    this.interestRate = 0.05;
}
public void dep() {
    double interest = balance * interestRate;
    deposit(interest);
}
public void withdraw(double amount) {
    if (amount <= balance) {
        balance = balance - amount;
        System.out.println("withdrawl successful");
    } else {
        System.out.println("Insufficient balance for withdrawl");
    }
}

```

class CurrentAcc extends Account

```

double minBal;
double serviceCharge;
CurrentAcc(String name, double num) {
    super(name, num, "current");
    this.minBal = 1000.0;
}

```

Date _____
Page _____

```

this.serviceCharge = 10.0;
public void withdrawl(double amount) {
    if (amount <= balance) {
        balance -= amount;
        System.out.println("Withdrawl Successful");
        checkMinBal();
    } else {
        System.out.println("Insufficient funds for withdrawl");
    }
}
void checkMinBal() {
    if (balance < minBal) {
        balance -= serviceCharge;
        System.out.println("Service charge imposed");
    }
}

class Bank {
    public static void main(String args[]) {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter customer name:");
        String name = s.next();
    }
}

```

S.O.P (" Enter account number");
double accNo = s.nextInt();
SavingsAcc s1 = new SavingsAcc(name, accNo);
CurrentAcc c1 = new CurrentAcc(name, accNo);

s1.deposit(1000);
s1.dep();
s1.dispBalance();

c1.deposit(15.00);
c1.withdrawl(200);
c1.dispBalance();

Output :-

Enter customer name : Aditya
Enter account number : 987654321
Deposit Successful
Balance : 1000.0
Interest deposited
Account Balance : 1050.0

Deposit Successful
Balance : 1500.0
withdrawl Successful
Service charge imposed
Account Balance : 1290.0

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.

```
package CIE;
```

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

```
public class Internals extends Student {
```

```
    public int[] internalMarks;
```

```
    public Internals(String usn, String name, int sem, int[] internalMarks) {
```

```
        super(usn, name, sem);
```

```
        this.internalMarks = internalMarks;
```

```
    }
```

```
}
```

```
package CIE;
```

```
public class Student {
```

```
    public String usn;
```

```
    public String name;
```

```
    public int sem;
```

```
    public Student(String usn, String name, int sem) {
```

```
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }
}

package SEE;

import CIE.Student;

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

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

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

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

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

Internals[] internalsArray = getInputForCIEStudents(scanner, n);
External[] externalsArray = getInputForSEESTudents(scanner, n);

int[][] finalMarks = calculateFinalMarks(internalsArray, externalsArray, n);

displayFinalMarks(internalsArray, finalMarks, n);

}

private static Internals[] getInputForCIEStudents(Scanner scanner, int n) {
    Internals[] internalsArray = new Internals[n];
    for (int i = 0; i < n; i++) {
        System.out.println("Enter details for CIE student " + (i + 1));
        System.out.print("USN: ");
        String usn = scanner.next();
        System.out.print("Name: ");
        String name = scanner.next();
        System.out.print("Semester: ");
        int sem = scanner.nextInt();
        System.out.println("Enter internal marks for 5 courses:");
        int[] internalMarks = new int[5];
        for (int j = 0; j < 5; j++) {
            System.out.print("Course " + (j + 1) + ": ");
            internalMarks[j] = scanner.nextInt();
        }
    }
}
```

```

        }

        internalsArray[i] = new Internals(usn, name, sem, internalMarks);

    }

    return internalsArray;
}

private static External[] getInputForSEEStudents(Scanner scanner, int n) {

    External[] externalsArray = new External[n];

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

        System.out.println("Enter details for SEE student " + (i + 1));

        System.out.print("USN: ");

        String usn = scanner.next();

        System.out.print("Name: ");

        String name = scanner.next();

        System.out.print("Semester: ");

        int sem = scanner.nextInt();

        System.out.println("Enter SEE marks for 5 courses:");

        int[] seeMarks = new int[5];

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

            System.out.print("Course " + (j + 1) + ": ");

            seeMarks[j] = scanner.nextInt();

        }

        externalsArray[i] = new External(usn, name, sem, seeMarks);

    }

    return externalsArray;
}

```

```
private static int[][] calculateFinalMarks(Internals[] internalsArray, External[]
externalsArray, int n) {
    int[][] finalMarks = new int[n][5];
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < 5; j++) {
            finalMarks[i][j] = internalsArray[i].internalMarks[j] +
externalsArray[i].seeMarks[j];
        }
    }
    return finalMarks;
}
```

```
private static void displayFinalMarks(Internals[] internalsArray, int[][][]
finalMarks, int n) {
    System.out.println("\nFinal Marks:");
    for (int i = 0; i < n; i++) {
        System.out.print("USN: " + internalsArray[i].usn + ", Name: " +
internalsArray[i].name +
", Semester: " + internalsArray[i].sem + ", Final Marks: ");
        for (int j = 0; j < 5; j++) {
            System.out.print(finalMarks[i][j] + " ");
        }
        System.out.println();
    }
}
```

OUTPUT:

```
Enter the number of students:  
1  
Enter details for CIE student 1  
USN: 120  
Name: Aditya  
Semester: 3  
Enter internal marks for 5 courses:  
Course 1: 35  
Course 2: 24  
Course 3: 34  
Course 4: 13  
Course 5: 24  
Enter details for SEE student 1  
USN: 120  
Name: Aditya  
Semester: 3  
Enter SEE marks for 5 courses:  
Course 1: 78  
Course 2: 65  
Course 3: 45  
Course 4: 88  
Course 5: 87  
  
Final Marks:  
USN: 120, Name: Aditya, Semester: 3, Final Marks: 113 89 79 101 111
```

2) Create a package CIF which has two classes - Student and Internals. The class Student has students like USN, name, marks, etc. The class Internals has members like The class Internals has an array that stores the internal marks scored in 5 courses of the current semester of the student. Create another package SFF which has the class External which is a derived class of student. This class has an array that stores the SFF marks scored in 5 courses of current sem of the student. Insert the two packages in a file that declares the final marks of a student in all five courses.

Program:-

```

package CIF;
public class Student {
    int usn;
    String name;
    int sem;
    public void accept() {
        Scanner s1 = new Scanner(System.in);
        S.O.P("Enter name : ");
        name = s1.nextLine();
        S.O.P("Enter usn : ");
        usn = s1.nextInt();
    }
}

```

```

S.O.P("Enter sem : ");
sem = s1.nextInt();

void display() {
    S.O.P(" USN = " + usn);
    S.O.P(" Name = " + name);
    S.O.P(" Sem = " + sem);
}

package CIF;
import java.util.Scanner;
class Internals extends Student {
    public int marks[] = new int[5];
    void acceptMarks() {
        Scanner s1 = new Scanner(System.in);
        for (int i=0; i<5; i++) {
            marks[i] = s1.nextInt();
        }
    }
}

void displayMarks() {
    for (int i=0; i<5; i++) {
        S.O.P(" Marks in CIF subject" + (i+1) + ":" + marks[i]);
    }
}

```

package SFF

```

import CIF.Student;
import java.util.Scanner;

public class External extends CIF.Student {
    int sec_marks[] = new int[5];
    public void secAccept() {
        Scanner s1 = new Scanner(System.in);
        for (int i=0; i<5; i++) {
            sec_marks[i] = s1.nextInt();
        }
    }
}

void secDisplay() {
    for (int i=0; i<5; i++) {
        S.O.P(" Marks in SFF Subject" + (i+1) + ":" + sec_marks[i]);
    }
}

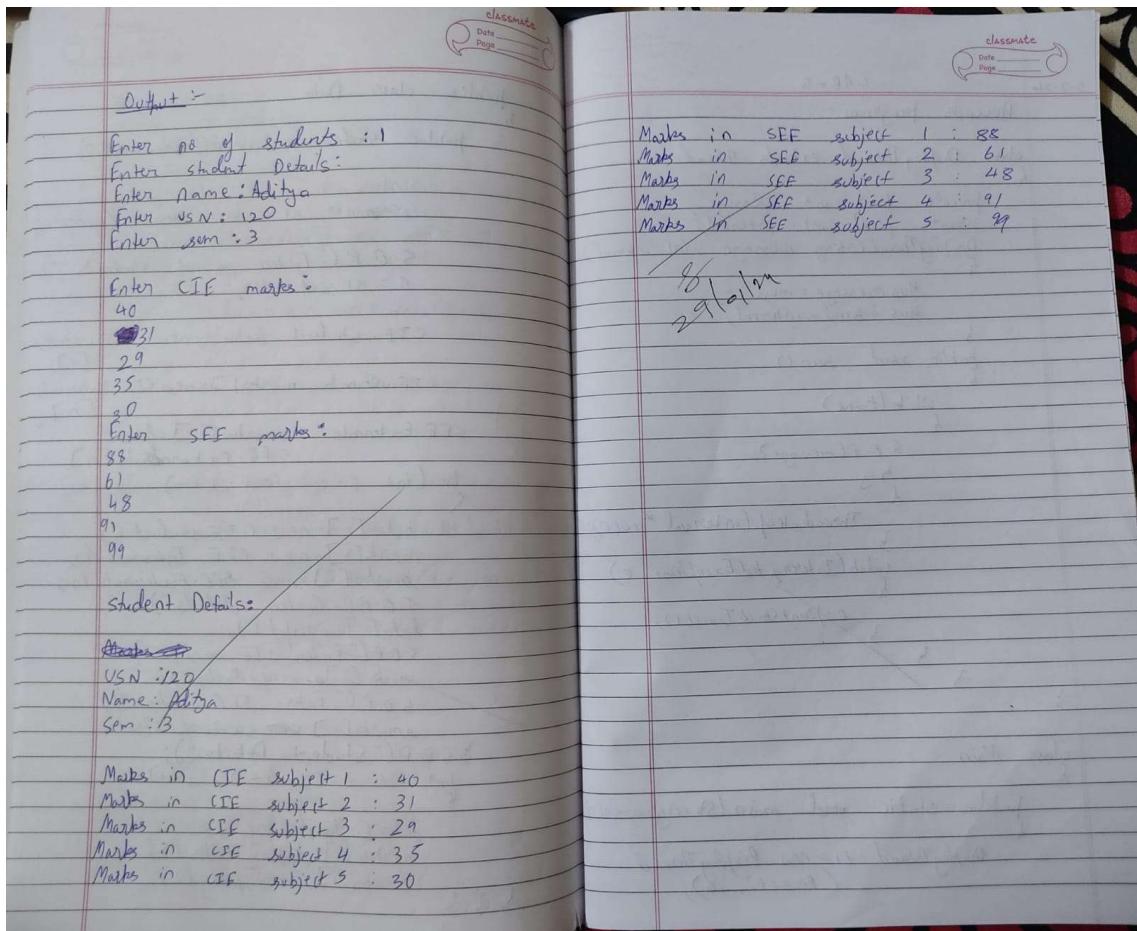
import java.util.Scanner;
import CIF.Student;
import CIF.Internals;
import CIF.Externals;

```

```

public class Data
{
    public static void main(String args[])
    {
        int n;
        Scanner s1 = new Scanner(System.in);
        S.O.P(" Enter no of students ");
        n = s1.nextInt();
        CIF.Student data[] = new CIF.Student[n];
        CIF.Internals marks[] = new CIF.Internals[n];
        SFF.Externals smarks[] = new SFF.Externals[n];
        for (int i=0; i<n; i++) {
            data[i] = new CIF.Student();
            marks[i] = new CIF.Internals();
            smarks[i] = new SFF.Externals();
            data[i].accept();
            marks[i].acceptMarks();
            smarks[i].secAccept();
            S.O.P(" Student Details : ");
            for (int j=0; j<n; j++) {
                data[i].display();
                marks[i].displayMarks();
                smarks[i].secDisplay();
            }
        }
    }
}

```



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=father’s age

```
import java.util.Scanner;

class WrongAge extends Exception {
    public WrongAge() {
        super("Invalid age! Age cannot be negative nor zero.");
    }

    public WrongAge(String message) {
        super(message);
    }
}

class Father {
    private int age;

    public Father(int age) throws WrongAge {
        if (age <= 0) {
            throw new WrongAge();
        }

        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 should be less than
Father's age.");

        }

        this.sonAge = sonAge;
    }

    public int getSonAge() {

        return sonAge;
    }
}

public class InheritanceException {

    public static void main(String[] args) {

        try {

            Scanner s1 = new Scanner(System.in);

            System.out.print("Enter Father's age: ");

            int fatherAge = s1.nextInt();
        }
    }
}
```

```
 Father f = new Father(fatherAge);
 System.out.print("Enter Son's age: ");
 int sonAge = s1.nextInt();
 Son s = new Son(f.getAge(), sonAge);
 System.out.println("Father's age: " + f.getAge());
 System.out.println("Son's age: " + s.getSonAge());
 } catch (WrongAge e) {
 System.out.println("Exception: " + e);
 } catch (Exception e) {
 System.out.println(
 "Exception: Invalid input. Please enter valid integer
 values.");
 }
 }
```

OUTPUT:

```
C:\Users\k Aditya Varma\Desktop\MY JAVA>java InheritanceException
Enter Father's age: 45
Enter Son's age: 55
Exception: WrongAge: Son's age should be less than Father's age.
```

LAB-4
Exception Handling

```
import java.util.Scanner;
class Exc extends Exception
{
    public Exc (String message)
    {
        super (message);
    }
}

class Father
{
    private int age;
    public Father (int age) throws Exc
    {
        if (age < 0)
            throw new Exc ("Age cannot
                           be negative");
    }

    public int getAge ()
    {
        return age;
    }
}

class Son extends Father
{
    private int sonAge;
    public Son (int fatherAge, int sonAge)
        throws Exc
    {
        super (fatherAge);
    }
}
```

classmate
Date _____
Page _____

```

if (sonAge >= fatherAge)
{
    throw new Exception("Son's age cannot
    be greater than or equal to
    Father's age");
}

public int getSonAge()
{
    return sonAge;
}

class Main
{
    public static void main(String args[])
    {
        try
        {
            Scanner s = new Scanner(System.in);
            S.O.P("Enter father's age : ");
            int fatherAge = s.nextInt();
            S.O.P("Enter son's age : ");
            int sonAge = s.nextInt();
            Son s1 = new Son(fatherAge, sonAge);
            S.O.P("Father's age : " + s1.getFatherAge());
            S.O.P("Son's age : " + s1.getSonAge());
        }
        catch (Exception e)
        {
            S.O.P("Error : " + e.getMessage());
        }
        catch (Exception e)
        {
            S.O.P("An error occurred : " +
                e.getMessage());
        }
    }
}

```

classmate
Date _____
Page _____

```

int positive(Scanner s)
{
    while (true)
    {
        try
        {
            int value = Integer.parseInt(s.nextLine());
            if (value < 0)
            {
                S.O.P("Please enter a
                non negative integer.");
            }
            else
            {
                return value;
            }
        }
        catch (NumberFormatException e)
        {
            S.O.P("Please enter a
            valid integer.");
        }
    }
}

```

Output :-

```

Enter father's age : 58
Enter Son's age : 25
Father's Age : 58
Son's Age : 25

```

Output :-

```

Enter father's age : 50
Enter Son's age : 51
Error!
Error: Son's age cannot be greater
than or equal to Father's Age.

```

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 DisplayThread extends Thread {  
    private final String message;  
    private final int interval;  
  
    DisplayThread(String message, int interval) {  
        this.message = message;  
        this.interval = interval;  
    }  
  
    public void run() {  
        while (true) {  
            System.out.println(message);  
            try {  
                Thread.sleep(interval * 1000);  
            } catch (InterruptedException e) {  
                e.printStackTrace();  
            }  
        }  
    }  
  
    public class Main {  
        public static void main(String[] args) {  
    }
```

```
    DisplayThread thread1 = new DisplayThread("BMS College of Engineering",
10);

    DisplayThread thread2 = new DisplayThread("CSE", 2);

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

}
```

OUTPUT:

```
C:\Users\k Aditya Varma\Desktop\MY JAVA>java Thr
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
^C
```

3-2-24

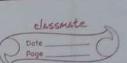
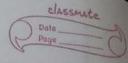
LAB - 5

Threads program:-

```

class DisplayThread extends Thread
{
    private final int message;
    private final int interval;
    DisplayThread (String message, int interval)
    {
        this.message = message;
        this.interval = interval;
    }
    public void run()
    {
        while (true)
        {
            System.out.println(message);
            try
            {
                Thread.sleep(interval * 1000);
            }
            catch (InterruptedException e)
            {
                e.printStackTrace();
            }
        }
    }
}
class Main
{
    public static void main (String args[])
    {
        DisplayThread t1 = new DisplayThread ("BMSCE", 10);
    }
}

```



DisplayThread t2 = new DisplayThread ("CSE", 2);

t1.start();
t2.start();

3

Output :-

```

BMSCE
CSE
CSE
CSE
CSE
CSE
BMSCE
CSE
CSE
CSE
CSE
CSE
BMSCE
CSE
CSE
CSE
CSE
CSE
CSE
CSE

```

05/02/24

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

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;
```

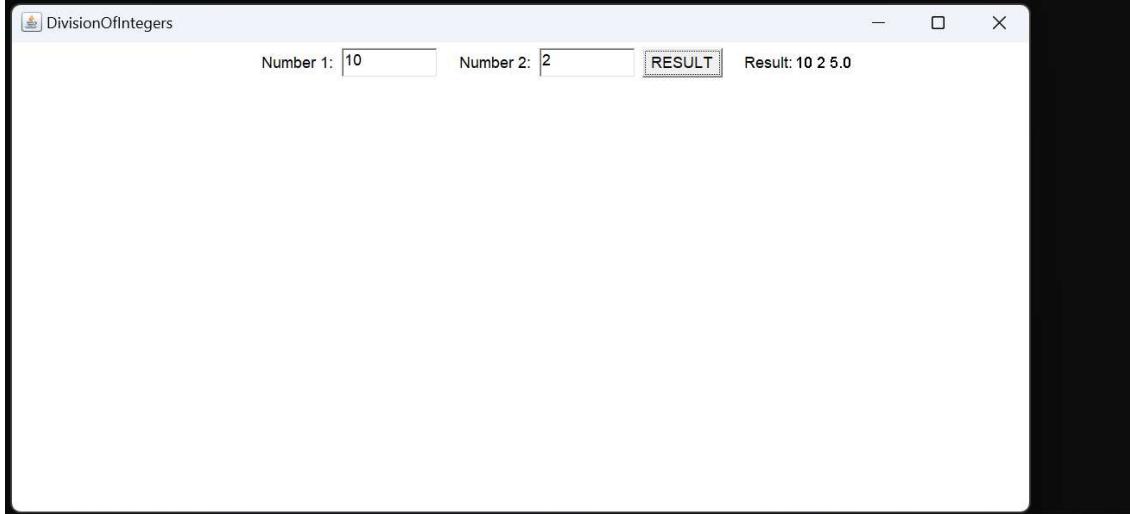
```
        out = "Number Format Exception! " + e1;
        repaint();
    } catch (ArithmetricException e2) {
        flag = 1;
        out = "Divide by 0 Exception! " + e2;
        repaint();
    }
}

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:

```
C:\Users\k Aditya Varma\Desktop\MY JAVA>javac DivisionMain1.java
C:\Users\k Aditya Varma\Desktop\MY JAVA>java DivisionMain1
```



Laboratory Program - 9

Write a program that creates a user interface to perform integer division. 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 integers the program would throw a NumberFormatException. If Num2 were zero, the program would throw an ArithmeticException displaying 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)
 }
}

```

classmate
Date _____
Page _____
```

```

n1=Integer.parseInt(num).getText();
n2=Integer.parseInt(num2).getText();

if(n2==0) throw new
ArithmaticException();
out=n1+" "+n2+":";
result=N/nm=n1/n2;
out+=String.valueOf(result+N/nm);
repaint();
}

catch(NumberFormatException e1)
{
flag=1;
out+="Number Format Exception! "+e1;
repaint();
}

catch(ArithmaticException 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
}

```

Output :-

Division of integers		
Number: [16]	Number2: [8]	Result: 20

```

classmate
Date _____
Page _____
```

```

n1=Integer.parseInt(num).getText();
n2=Integer.parseInt(num2).getText();

if(n2==0) throw new
ArithmaticException();
out=n1+" "+n2+":";
result=N/nm=n1/n2;
out+=String.valueOf(result+N/nm);
repaint();

catch(NumberFormatException e1)
{
flag=1;
out+="Number Format Exception! "+e1;
repaint();
}

catch(ArithmaticException 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
}

```

Output :-

Division of integers		
Number: [16]	Number2: [8]	Result: 20

Report On Mouse Events Demo Program

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

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