VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

on

OBJECT ORIENTED JAVA PROGRAMMING

Submitted by

SHUBHANSHU RAJ (1BM23CS325)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING

(Autonomous Institution under VTU)

BENGALURU-560019 Sep

2024-Jan 2025

B. M. S. College of Engineering,

Bull Temple Road, Bangalore 560019

(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "**OBJECT ORIENTED JAVA PROGRAMMING**" carried out by **SHUBHANSHU RAJ (1BM23CS325),** who is bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2024-

25. The Lab report has been approved as it satisfies the academic requirements in respect of **Object-Oriented Java Programming Lab - (23CS3PCOOJ)** work prescribed for the said degree.

Dr. Nandhini Vineeth

Associate Professor,
Department of CSE,
BMSCE, Bengaluru

Dr. Kavitha Sooda

Professor and Head,
Department of CSE
BMSCE, Bengaluru

INDEX

LABORATORY PROGRAM - 1

Develop a Java program that prints all real solutions to the quadratic equation ax2 +bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminate b2 -4ac is negative, display a message stating that there are no real solutions.

```
import java.util.Scanner;

class QuadraticEquation {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter the coefficient of a: ");
      int a = sc.nextInt();
      System.out.print("Enter the coefficient of b: ");
      int b = sc.nextInt();
      System.out.print("Enter the coefficient of c: ");
      int c = sc.nextInt();
      if (a == 0) {
         System.out.println("Invalid input: 'a' cannot be zero for a quadratic equation.");
         return;
      }
      double d = (b * b) - (4 * a * c);
      if (d > 0) {
         System.out.println("Real and Distinct Roots");
         double r1 = (-b + Math.sqrt(d)) / (2 * a);
         double r2 = (-b - Math.sqrt(d)) / (2 * a);
         System.out.println("The roots of the equation are:");
         System.out.println(r1 + " and " + r2);
      }
      else if (d == 0) {
         System.out.println("Real and Equal Roots");
         double r = -b / (2.0 * a);
         System.out.println("Root: " + r);
      }
      else {
         System.out.println("No Real Roots");
      }
```

```
Enter the Coefficient. of a:2
Enter the Coefficient of b:-8
Enter the Coefficient of c:3
Real and Distinct Roots
The roots of the Equation is:
3.58113883008419 and 0.41886116991581024
```

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.*; class Student {
   String name; String USN; int[] credits; int[] marks;
   Scanner sc = new Scanner(System.in);

   void getDetails() {
      System.out.print("Enter the Name:"); name = sc.next();
System.out.print("Enter the USN:"); USN = sc.next();
System.out.print("Enter the number of subjects:"); int n = sc.nextInt();
credits = new int[n]; marks = new int[n];
for (int i = 0; i < n; i++) {
         System.out.print("Enter the credits for:" + (i+1) + "Subject:"); credits[i] = sc.nextInt();
         System.out.print("Enter the marks for:" + (i+1) + "Subject:"); marks[i] = sc.nextInt();
      }
   }

   void showDetails() {
System.out.println("Name: " + name); System.out.println("USN: " + USN);
for (int i = 0; i < credits.length; i++) {
         System.out.println("Marks:" + i + "th Subject:" + marks[i]); System.out.println("Credits:"
         + i + "th Subject:" + credits[i]);
      }
   }

   double SGPACalc() { double sgpa=0;
int total_Credits=0;
      for (int i = 0; i < credits.length; i++) { sgpa += (marks[i]/10)* credits[i]; total_Credits +=
         credits[i];
      }
```

```
        return sgpa/total_Credits;
    }
  }
  class StudentSPGA{
public static void main(String[] args) { Student s= new Student(); s.getDetails();
  s.showDetails();
  System.out.println("SGPA:"+s.SGPACalc());
    }
  }
```

```
Enter the Name:Shreyas
Enter the USN:1BM23CS321
Enter the number of subjects:3
Enter the credits for:1Subject:2
Enter the marks for:1Subject:70
Enter the credits for:2Subject:3
Enter the marks for:2Subject:80
Enter the credits for:3Subject:4
Enter the marks for:3Subject:90
Name: Shreyas
USN: 1BM23CS321
Marks:0th Subject:70
Credits:0th Subject:2
Marks:1th Subject:80
Credits:1th Subject:3
Marks:2th Subject:90
Credits:2th Subject:4
SGPA:8.222222222222221
```

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, author;
   double price;
   int num_pages;

   // Constructor
   Book(String name, String author, double price, int num_pages) {
      this.name = name;
      this.author = author;
      this.price = price;
      this.num_pages = num_pages;
   }

   // Method to set book details
   void setDetails() {
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter the name of the book:");
      name = sc.nextLine();
      System.out.println("Enter name of the Author: ");
      author = sc.nextLine();
      System.out.println("Enter price of the book: ");
      price = sc.nextDouble();
      System.out.println("Enter the number of pages: ");
      num_pages = sc.nextInt();
   }

   // Method to get book details
   void getDetails() {
      System.out.println("Name of Book: " + name);
      System.out.println("Name of Author: " + author);
      System.out.println("Price: " + price);
      System.out.println("Number of Pages: " + num_pages);
   }

   // Display method (returning string)
   String Display() {
      return "Book Name: " + name + ", Author: " + author + ", Price: " + price + ", No. of
Pages: " + num_pages;
```

```
    }
}

class Library {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number of books:");
        int n = sc.nextInt();
        sc.nextLine();
        Book[] books = new Book[n];
        for (int i = 0; i < n; i++) {
            books[i] = new Book("", "", 0, 0);
            books[i].setDetails();
        }
        for (int i = 0; i < n; i++) {
            books[i].getDetails();
        }
    }
}
```

```
Enter the number of books:
2
Enter the name of the book:
Charlie and the Chocolate Factory
Enter name of the Author:
Roald Dahl
Enter price of the book:
450
Enter the number of pages:
200
Enter the name of the book:
Data Structures
Enter name of the Author:
Remma Thareja
Enter price of the book:
260
Enter the number of pages:
450
Name of Book: Charlie and the Chocolate Factory
Name of Author: Roald Dahl
Price: 450.0
Number of Pages: 200
Name of Book: Data Structures
Name of Author: Remma Thareja
Price: 260.0
Number of Pages: 450
```

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.

```
import java.util.Scanner; abstract
class Shape {
    int dim1;
    int dim2;

Shape(int dim1, int dim2) { this.dim1 =
    dim1;
    this.dim2 = dim2;
        }

        abstract void printArea();
    }
    class Rectangle extends Shape { Rectangle(int
        length, int width) {
    super(length, width);
        }

        void printArea() {
    int area = dim1 * dim2;
    System.out.println("Area of Rectangle: " + area);
        }
    }
    class Triangle extends Shape {
        Triangle(int base, int height) {
    super(base, height);
        }

        void printArea() {
    double area = 0.5 * dim1 * dim2;
    System.out.println("Area of Triangle: " + area);
        }
    }
    class Circle extends Shape { public
        Circle(int radius) {
```

```
super(radius, 0);

    public void printArea() {
double area = Math.PI * dim1 * dim1;
System.out.println("Area of Circle: " + area);
    }
}

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

System.out.println("Select a shape:");
System.out.println("1. Rectangle");
System.out.println("2. Triangle");
System.out.println("3. Circle");

int choice = scanner.nextInt(); Shape
shape=null;
        switch (choice) { case 1:
                System.out.println("Enter length of rectangle:"); int
                length = scanner.nextInt();
                System.out.println("Enter width of rectangle:"); int width
                = scanner.nextInt();
                shape = new Rectangle(length, width); break;
            case 2:
                System.out.println("Enter base of triangle:"); int base =
                scanner.nextInt();
                System.out.println("Enter height of triangle:"); int height
                = scanner.nextInt();
                shape = new Triangle(base, height); break;
            case 3:
```

```
            System.out.println("Enter radius of circle:"); int
            radius = scanner.nextInt();
            shape = new Circle(radius); break;
        default:
            System.out.println("Invalid choice"); return;
    }

shape.printArea();

        }
}
```

```
C:\1BM23CS321>java Calc
Select a shape:
1. Rectangle
2. Triangle
3. Circle
1
Enter length of rectangle:
12
Enter width of rectangle:
11
Area of Rectangle: 132

C:\1BM23CS321>java Calc
Select a shape:
1. Rectangle
2. Triangle
3. Circle
2
Enter base of triangle:
12
Enter height of triangle:
12
Area of Triangle: 72.0

C:\1BM23CS321>java Calc
Select a shape:
1. Rectangle
2. Triangle
3. Circle
3
Enter radius of circle:
12
Area of Circle: 452.3893421169302

C:\1BM23CS321>
```

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;

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

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

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

```
    }

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

    public abstract void withdraw(double amount);

    public abstract void updateBalance();
}

class CurAcct extends Account {
    private static final double MINIMUM_BALANCE = 1000;
    private static final double SERVICE_CHARGE = 50;

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

    public void withdraw(double amount) {
        if (balance - amount >= 0) {
            balance -= amount;
            System.out.println("Withdrawal successful. New balance: " + balance);
            updateBalance();
        } else {
            System.out.println("Insufficient balance.");
        }
    }

    public void updateBalance() {
        if (balance < MINIMUM_BALANCE) {
            balance -= SERVICE_CHARGE;
            System.out.println("Service charge imposed due to low balance. New balance: " +
balance);
        }
    }
}

class SavAcct extends Account {
    private static final double INTEREST_RATE = 0.05;

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

    public void computeInterest() {
```

```
        double interest = balance * INTEREST_RATE;
        balance += interest;
        System.out.println("Interest computed and added. New balance: " + balance);
    }

    public void withdraw(double amount) {
        if (balance - amount >= 0) {
            balance -= amount;
            System.out.println("Withdrawal successful. New balance: " + balance);
        } else {
            System.out.println("Insufficient balance.");
        }
    }

    public void updateBalance() {
        computeInterest();
    }
}

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

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

        System.out.print("Enter Account Number: ");
        String accNum = scanner.nextLine();

        System.out.print("Enter Account Type (Savings/Current): ");
        String type = scanner.nextLine();

        System.out.print("Enter Initial Balance: ");
        double balance = scanner.nextDouble();

        Account account;
        if (type.equalsIgnoreCase("Savings")) {
            account = new SavAcct(name, accNum, balance);
        } else if (type.equalsIgnoreCase("Current")) {
            account = new CurAcct(name, accNum, balance);
        } else {
            System.out.println("Invalid Account Type.");
            scanner.close();
            return;
        }
```

```
        boolean quit = false;
        while (!quit) {
            System.out.println("\n1. Deposit\n2. Withdraw\n3. Display Balance\n4. Update
Balance\n5. Quit");
            System.out.print("Enter your choice: ");
            int choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter amount to deposit: ");
                    double depositAmount = scanner.nextDouble();
                    account.deposit(depositAmount);
                    break;
                case 2:
                    System.out.print("Enter amount to withdraw: ");
                    double withdrawAmount = scanner.nextDouble();
                    account.withdraw(withdrawAmount);
                    break;
                case 3:
                    account.displayBalance();
                    break;
                case 4:
                    account.updateBalance();
                    break;
                case 5:
                    quit = true;
                    break;
                default:
                    System.out.println("Invalid choice. Please try again.");
            }
        }

        scanner.close();
    }
}
```

```
Enter Customer Name: Shreyas
Enter Account Number: 12234
Enter Account Type (Savings/Current): Savings
Enter Initial Balance: 1000000

1. Deposit
2. Withdraw
3. Display Balance
4. Update Balance
5. Quit
Enter your choice: 1
Enter amount to deposit: 10000
Deposit successful. New balance: 1010000.0

1. Deposit
2. Withdraw
3. Display Balance
4. Update Balance
5. Quit
Enter your choice: 3
Account Balance: 1010000.0

1. Deposit
2. Withdraw
3. Display Balance
4. Update Balance
5. Quit
Enter your choice: 4
Interest computed and added. New balance: 1060500.0

1. Deposit
2. Withdraw
3. Display Balance
4. Update Balance
5. Quit
Enter your choice: 5
Thank you for banking with us.
```

LABORATORY PROGRAM - 6

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

INTERNALS.JAVA

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

public class Internals extends Student {
  protected int[] marks = new int[5];  // Marks for 5 courses

  public void inputCIEmarks() {
    Scanner scanner = new Scanner(System.in);
    System.out.println("Enter Internal marks for 5
    courses:"); for (int i = 0; i < 5; i++) {
      System.out.print("Enter marks for Course " + (i + 1) +
      ": "); marks[i] = scanner.nextInt();
    }
  }

  public void displayCIEmarks() {
    System.out.println("Internal Marks for 5
    courses:"); for (int i = 0; i < 5; i++) {
      System.out.println("Course " + (i + 1) + ": " + marks[i]);
    }
  }
}
```

STUDENT.JAVA

```
package CIE;
import
java.util.Scanner;
public class Student
{
```

```
  protected String usn;
  protected String name;
  protected int sem;

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

  public void displayStudentDetails() {
    System.out.println("USN: " + usn);
    System.out.println("Name: " + name);
    System.out.println("Semester: " + sem);
  }
}
```

EXTERNALS.JAVA

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

public class Externals extends
  Internals { protected int[]
  externalMarks = new int[5];
  protected int[ ] finalMarks = new
  int[5];

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

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

```
    System.out.println("Enter External marks for 5
    courses:"); for (int i = 0; i < 5; i++) {
      System.out.print("Enter marks for Course " + (i + 1) +
      ": "); externalMarks[i] = scanner.nextInt();
    }
  }

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

  public void displayFinalMarks() {
    displayStudentDetails();
    displayCIEmarks();
    System.out.println("Final Marks (Internal + External) for 5
    courses:"); for (int i = 0; i < 5; i++) {
      System.out.println("Course " + (i + 1) + ": " + finalMarks[i]);
    }
  }
```

MAIN.JAVA

```
import
SEE.Externals;
import
java.util.Scanner;

public class Main {
  public static void main(String[] args) {
    Scanner scanner = new
    Scanner(System.in);
    System.out.print("Enter the number of
    students: "); int n = scanner.nextInt();

    Externals[] students = new Externals[n];

    for (int i = 0; i < n; i++) { students[i] = new Externals();

      System.out.println("Enter details for student "
      + (i + 1)); students[i].inputStudentDetails();
      students[i].inputCIEmarks();
      students[i].inputSEEmarks();
      students[i].calculateFinalMarks();
    }
```

```
D:\1BM23CS321\Package>java Main.java
Enter the number of students: 1
Enter details for student 1
Enter USN: 1BM23CS321
Enter Name: Shreyas
Enter Semester: 3
Enter Internal marks for 5 courses:
Enter marks for Course 1: 45
Enter marks for Course 2: 43
Enter marks for Course 3: 48
Enter marks for Course 4: 50
Enter marks for Course 5: 39
Enter External marks for 5 courses:
Enter marks for Course 1: 45
Enter marks for Course 2: 50
Enter marks for Course 3: 44
Enter marks for Course 4: 34
Enter marks for Course 5: 41
USN: 1BM23CS321
Name: Shreyas
Semester: 3
Internal Marks for 5 courses:
Course 1: 45
Course 2: 43
Course 3: 48
Course 4: 50
Course 5: 39
Final Marks (Internal + External) for 5 courses:
Course 1: 90
Course 2: 93
Course 3: 92
Course 4: 84
Course 5: 80
```

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.

```
class WrongAge extends
  Exception { String message;

  WrongAge(String message) {
    this.message = message;
  }

  public String toString() {
    return "WrongAge Exception: " + message;
  }
}

class Father
  { int fAge;
  Father(int age) throws
    WrongAge { if (age < 0) {
      throw new WrongAge("Father's age cannot be negative!");
    }
    fAge = age;
  }
}

class Son extends
  Father { int sAge;

  Son(int fAge, int sAge) throws
    WrongAge { super(fAge);

    if (sAge < 0) {
      throw new WrongAge("Son's age cannot be negative!");
    }
    if (sAge >= fAge) {
      throw new WrongAge("Son's age cannot be greater than or equal to Father's age!");
    }

    this.sAge = sAge;
  }
}

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

```
      Father father1 = new
      Father(40); Son son1 = new
      Son(40, 20);
      System.out.println("Father's age: " + father1.fAge + ", Son's age: " + son1.sAge);

      Father father2 = new Father(-5);
    }
    catch (WrongAge e) {
      System.out.println(e);
    }

    try {
      Son son2 = new Son(35, 40);
    }
    catch (WrongAge e) {
      System.out.println(e);
    }

    try {
      Son son3 = new Son(50, -10);
    }
    catch (WrongAge e) {
      System.out.println(e);
    }
  }
}
```

```
D:\1BM23CS321>java FatherSon.java
Father's age: 40, Son's age: 20
WrongAge Exception: Father's age cannot be negative!
WrongAge Exception: Son's age cannot be greater than or equal to Father's age!
WrongAge Exception: Son's age cannot be negative!
```

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 Threads extends Thread{
   String s; int time;
Threads(String s, int time){ this.s
= s;
this.time = time;
   }
public void run(){ try{
        while(true){
          System.out.println(s);
          Thread.sleep(time * 1000);
        }
      }
catch(InterruptedException ie){
        System.out.println("Thread occurs: " + ie);
      }
public class LP8{
  public static void main(String[] args) {
    Threads t1 = new Threads("BMS College of Engineering", 10);
    Threads t2 = new Threads("CSE", 2);

t1.start();
t2.start();
  }
```

```
C:\Users\shrey\.jdks\openjdk-23.0.1\
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
```

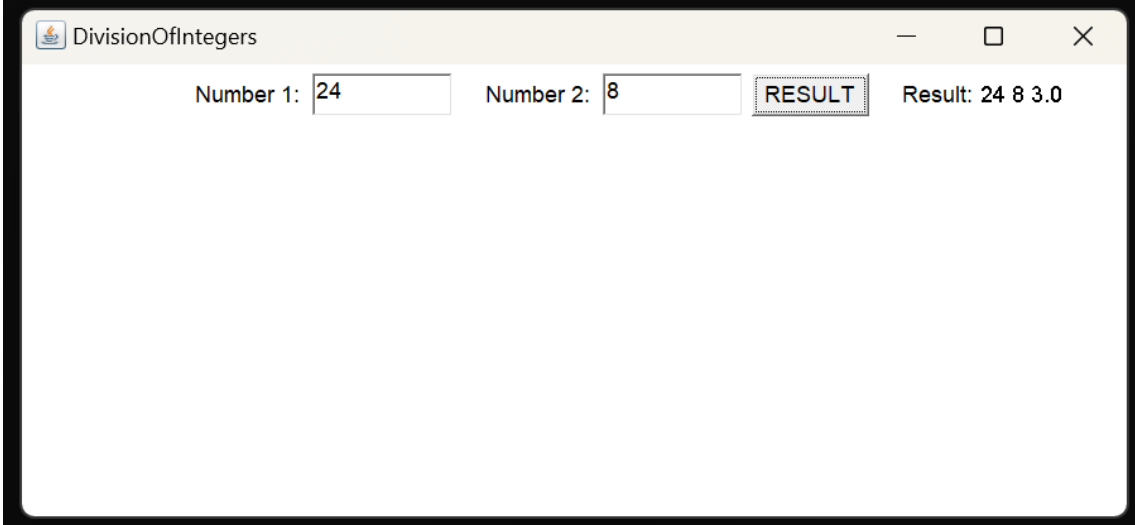
LABORATORY PROGRAM - 9

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 Arithmetic Exception 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;
}
```

```
D:\NotePad++\Java>javac DivisionMain1.java

D:\NotePad++\Java>java DivisionMain1
```



LABORATORY PROGRAM - 10

Demonstrate Interprocess communication and deadlock

```
class Q {
int n;
boolean valueSet = false;

synchronized int get() {
while(!valueSet)
try {
System.out.println("\nConsumer waiting\n");
wait();
} catch(InterruptedException e) {
System.out.println("InterruptedException caught");
}
System.out.println("Got: " + n);
valueSet = false;
System.out.println("\nIntimate Producer\n");
notify();
return n;
}

synchronized void put(int n) {
while(valueSet)
try {
System.out.println("\nProducer waiting\n");
wait();
} catch(InterruptedException e) {
System.out.println("InterruptedException caught");
}
this.n = n;
valueSet = true;
System.out.println("Put: " + n);
System.out.println("\nIntimate Consumer\n");
notify();
}
}

class Producer implements Runnable {
Q q;
Producer(Q q) {
this.q = q;
new Thread(this, "Producer").start();
}
public void run() {
```

```
int i = 0;
while(i<15) {
q.put(i++);
}
}
}

class Consumer implements Runnable {
Q q;
Consumer(Q q) {
this.q = q;
new Thread(this, "Consumer").start();
}
public void run() {
        int i=0;
while(i<15) {
int r=q.get();
System.out.println("consumed:"+r);
i++;
}
}
}

class PCFixed {
public static void main(String args[]) {
Q q = new Q();
new Producer(q);
new Consumer(q);
System.out.println("Press Control-C to stop.");
}
}
```