

# **B.M.S COLLEGE OF ENGINEERING BENGALURU**

Autonomous Institute, Affiliated to VTU



## **LAB REPORT**

**23CS3PCOOJ**

Submitted in partial fulfilment of the requirements for Lab

Bachelor of Engineering

in

Computer Science and Engineering

Submitted by:

**NIKHILESH C**

**(1BM22CS181)**

Department of Computer Science and Engineering,

B.M.S College of Engineering,

Bull Temple Road, Basavanagudi, Bangalore, 560 019

2023-2024.

## **Lab Program 1:**

Q) Write a Java program to display the roots of a quadratic equation.

Solution:

```
import java.util.Scanner;

class Quadratic{

    int a,b,c;

    double r1,r2,d;

    void getd(){

        Scanner s = new Scanner(System.in);

        System.out.println("Enter the coefficients a,b,c");

        a=s.nextInt();

        b=s.nextInt();

        c=s.nextInt();

    }

    void compute(){

        while(a==0){

            System.out.println("Not a quadratic equation");

            System.out.println("Enter a non-zero value for a: ");

            Scanner s=new Scanner(System.in);

            a=s.nextInt();

        }

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

        if(d==0){

            r1=(-b)/(2*a);

            System.out.println("Roots are real and equal");

            System.out.println("Root 1 = Root 2 = "+r1);

        }

        else if (d>0){

            r1=(((-b)+(Math.sqrt(d)))/(double)(2*a));

            r2=(((-b)-(Math.sqrt(d)))/(double)(2*a));

        }

    }

}
```

```

        System.out.println("Roots are real and distinct");

        System.out.println("Root 1 = "+r1+" Root 2 = "+r2);

    }

    else if (d<0){

        System.out.println("Roots are imaginary");

        r1=(-b)/(2*a);

        r2=Math.sqrt(-d)/(2*a);

        System.out.println("Root 1 =" +r1+"+i"+r2);

        System.out.println("Root 1 =" +r1+"-i"+r2);

    }

}

class QuadraticMain{

    public static void main(String args[]){

        Quadratic q = new Quadratic();

        q.getd();

        q.compute();

    }

}

```

Output:

```

Enter the coefficients a,b,c
0 1 2
Not a quadratic equation
Enter a non zero value for a:
1
Roots are imaginary
Root 1 =0.0+i1.3228756555322954
Root 1 =0.0-i1.3228756555322954

```

```

Enter the coefficients a,b,c
1 -2 1
Roots are real and equal
Root 1 = Root 2 = 1.0

```

**Lab Program 2:**

Q) Write a java program to calculate SGPA of a student.

Solution:

```

import java.util.Scanner;

class Subject{

```

```

int subjectMarks;
int credits;
int grade;
}

class Student{
    Subject subject[];
    String name;
    String usn;
    double SGPA;
    Scanner s;
    Student(){
        int i;
        subject=new Subject[9];
        for(i=0;i<9;i++){
            subject[i]=new Subject();
        }
        s=new Scanner(System.in);
    }

    void getStudentDetails(){
        System.out.println("Enter your name: ");
        name=s.next();
        System.out.println("Enter your USN: ");
        usn=s.next();
    }

    void getMarks(){
        for(int i=0;i<9;i++){
            System.out.println("Enter marks for subject "+(i+1)+": ");
        }
    }
}

```

```

        subject[i].subjectMarks=s.nextInt();
        System.out.println("Enter the credits for subject "+(i+1)+": ");
        subject[i].credits=s.nextInt();
        subject[i].grade=(subject[i].subjectMarks/10)+1;
        if(subject[i].grade==11)
            subject[i].grade=10;
        if(subject[i].grade<=4)
            subject[i].grade=0;
    }
}

```

```

void computeSGPA(){
    int effectiveScore=0;
    int totalCredits=0;
    for(int i=0;i<9;i++){
        effectiveScore+=(subject[i].grade*subject[i].credits);
        totalCredits+=subject[i].credits;
    }
    SGPA=(double)effectiveScore/(double)totalCredits;
}
}

```

```

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

```

```

        System.out.println("Name of Student: "+s1.name);
        System.out.println("USN of student: "+s1.usn);
        System.out.println("SGPA of the student: "+s1.SGPA);
    }
}

```

Output:

```

Enter your name:
ABC
Enter your USN:
1BM22CS001
Enter marks for subject 1:
98
Enter the credits for subject 1:
4
Enter marks for subject 2:
92
Enter the credits for subject 2:
4
Enter marks for subject 3:
86
Enter the credits for subject 3:
3
Enter marks for subject 4:
92
Enter the credits for subject 4:
3
Enter marks for subject 5:
97
Enter the credits for subject 5:
3
Enter marks for subject 6:
98

```

```

Enter the credits for subject 6:
1
Enter marks for subject 7:
99
Enter the credits for subject 7:
1
Enter marks for subject 8:
95
Enter the credits for subject 8:
1
Enter marks for subject 9:
87
Enter the credits for subject 9:
1
Name of Student: ABC
USN of student: 1BM22CS001
SGPA of the student: 9.80952380952381

```

### Lab Program 3:

Q) Write a Java Program to display details of books.

Solution:

```

import java.util.Scanner;
class Books{
    private String name;
    private String author;
    private int price;
    private int num_pages;
}

```

```

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

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

class BookMain{
    public static void main(String args[]){
        Scanner s=new Scanner(System.in);
        int n;
        System.out.println("Enter the number of books: ");
        n=s.nextInt();
        Books b[]=new Books[n];
        for(int i=0;i<n;i++){
            System.out.println("Enter details for book "+(i+1)+":");
            System.out.print("Name: ");
            String name1=s.next();
            System.out.print("Author: ");
            String author=s.next();
            System.out.print("Price: ");
        }
    }
}

```

```

        int price=s.nextInt();
        System.out.print("Number of pages: ");
        int num_pages=s.nextInt();
        b[i]=new Books(name1,author,price,num_pages);
    }
    System.out.println("\nDetails of the Books:");
    for(int i=0;i<n;i++){
        System.out.println("\nBook "+(i+1)+":\n"+b[i]);
    }
    s.close();
}

```

Output:

```

Enter the number of books:
2
Enter details for book 1:
Name: LittleWomen
Author: Name1
Price: 360
Number of pages: 320
Enter details for book 2:
Name: Thumbolina
Author: Name2
Price: 500
Number of pages: 540

```

```

Details of the Books:

Book 1:
Book name:LittleWomen
Author name:Name1
Price:360
Number of pages:320

Book 2:
Book name:Thumbolina
Author name:Name2
Price:500
Number of pages:540

```

### Lab Program 4:

Q) Write a java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Three classes named Rectangle, Triangle and Circle extend Shape and print their respective areas based on the user input for dimensions.

Solution:

```

import java.util.Scanner;
class InputScanner{
    Scanner s=new Scanner(System.in);

```

```

double getInput(String prompt){
    System.out.println(prompt);
    return s.nextDouble();
}

}

abstract class Shape extends InputScanner{
    double s1,s2;
    abstract void area();
}

class Rectangle extends Shape{

    Rectangle(){
        s1=getInput("Enter the length of the rectangle:");
        s2=getInput("Enter the breadth of the rectangle:");
    }

    void area(){
        double area=s1*s2;
        System.out.println("Area of the rectangle="+area);
    }
}

class Triangle extends Shape{

    Triangle(){
        s1=getInput("Enter the base of the triangle:");
        s2=getInput("Enter the height of the triangle:");
    }

    void area(){
        double area=s1*s2/2;
    }
}

```

```

        System.out.println("Area of the triangle="+area);
    }

}

class Circle extends Shape{
    Circle(){}
    s1=getInput("Enter the radius of the circle:");
}

void area(){
    double area=Math.PI*s1*s1;
    System.out.println("Area of the circle="+area);
}

}

class Main1{
    public static void main(String args[]){
        Rectangle r=new Rectangle();
        Triangle t=new Triangle();
        Circle c=new Circle();
        r.area();
        t.area();
        c.area();
    }
}

```

Output:

```
Enter the length of the rectangle:  
4  
Enter the breadth of the rectangle:  
3  
Enter the base of the triangle:  
6  
Enter the height of the triangle:  
3  
Enter the radius of the circle:  
4  
Area of the rectangle=12.0  
Area of the triangle=9.0  
Area of the circle=50.26548245743669
```

## **Lab Program 5:**

Q) Write a java program to work with bank system.

Solution:

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

```
class Account {  
    String customerName;  
    int accountNumber;  
    String accountType;  
    double balance;  
  
    // Constructor  
    public Account(String name, int number, String type, double initialBalance) {  
        customerName = name;  
        accountNumber = number;  
        accountType = type;  
        balance = initialBalance;  
    }  
  
    // Method to accept deposit and update the balance  
    public void deposit(double amount) {  
        balance += amount;  
    }
```

```

        System.out.println("Deposit successful. Updated balance: " + balance);
    }

// Method to display the balance
public void displayBalance() {
    System.out.println("Current Balance: " + balance);
}

}

class CurrAcct extends Account {
    double minBalance;
    double serviceCharge;

    // Constructor
    public CurrAcct(String name, int number, double initialBalance) {
        super(name, number, "Current", initialBalance);
        minBalance = 500.0; // Set the minimum balance for current account
        serviceCharge = 50.0; // Set the service charge for falling below minimum balance
    }

    // Method to withdraw and update balance, checking for minimum balance
    public void withdraw(double amount) {
        if ((balance - amount) >= minBalance) {
            balance -= amount;
            System.out.println("Withdrawal successful. Updated balance: " + balance);
        } else {
            System.out.println("Insufficient funds. Service charge of " + serviceCharge + " imposed.");
            balance -= serviceCharge;
            System.out.println("Updated balance after service charge: " + balance);
        }
    }
}

```

```

    }

}

class SavAcct extends Account {
    double interestRate;

    // Constructor
    public SavAcct(String name, int number, double initialBalance, double rate) {
        super(name, number, "Savings", initialBalance);
        interestRate = rate;
    }

    // Method to compute and deposit interest
    public void depositInterest() {
        double interest = balance * (interestRate / 100);
        balance += interest;
        System.out.println("Interest of " + interest + " deposited. Updated balance: " + balance);
    }

    // Method to withdraw and update balance
    public void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
            System.out.println("Withdrawal successful. Updated balance: " + balance);
        } else {
            System.out.println("Insufficient funds for withdrawal.");
        }
    }
}

```

```

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

        // Sample usage
        CurrAcct currentAccount = new CurrAcct("John Doe", 123456, 1000.0);
        SavAcct savingsAccount = new SavAcct("Jane Smith", 789012, 2000.0, 5.0);

        int choice;
        do {
            System.out.println("\nMenu:");
            System.out.println("1. Deposit");
            System.out.println("2. Display Balance");
            System.out.println("3. Deposit Interest (for Savings Account)");
            System.out.println("4. Withdraw");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: ");
            choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter deposit amount: ");
                    double depositAmount = scanner.nextDouble();
                    currentAccount.deposit(depositAmount);
                    break;
                case 2:
                    currentAccount.displayBalance();
                    break;
                case 3:
                    if (savingsAccount instanceof SavAcct) {

```

```

((SavAcct) savingsAccount).depositInterest();
} else {
    System.out.println("This option is applicable only for Savings Account.");
}
break;

case 4:
    System.out.print("Enter withdrawal amount: ");
    double withdrawalAmount = scanner.nextDouble();
    currentAccount.withdraw(withdrawalAmount);
    break;

case 5:
    System.out.println("Exiting the program. Goodbye!");
    break;

default:
    System.out.println("Invalid choice. Please enter a valid option.");
}

} while (choice != 5);

}

```

Output:

```

Menu:
1. Deposit
2. Display Balance
3. Deposit Interest (for Savings Account)
4. Withdraw
5. Exit
Enter your choice: 1
Enter deposit amount: 2500
Deposit successful. Updated balance: 3500.0

```

```
Menu:  
1. Deposit  
2. Display Balance  
3. Deposit Interest (for Savings Account)  
4. Withdraw  
5. Exit  
Enter your choice: 3  
Interest of 100.0 deposited. Updated balance: 2100.0
```

```
Menu:  
1. Deposit  
2. Display Balance  
3. Deposit Interest (for Savings Account)  
4. Withdraw  
5. Exit  
Enter your choice: 4  
Enter withdrawal amount: 1000  
Withdrawal successful. Updated balance: 2500.0
```

## **Lab Program 6:**

Q) Write a java program to implement packages.

Solution:

*File 1: Main.java*

```
import SEE.Externals;  
  
class Main{  
  
    public static void main(String args[]){  
  
        int numOfStudents=2;  
  
        Externals finalMarks[]=new Externals[numOfStudents];  
  
        for(int i=0;i<numOfStudents;i++){  
  
            finalMarks[i]=new Externals();  
  
            finalMarks[i].inputStudentDetails();  
  
            System.out.println("Enter CIE marks");  
  
            finalMarks[i].inputCIEmarks();  
  
            System.out.println("Enter SEE marks");  
  
            finalMarks[i].inputSEEmarks();  
    }  
}
```

```

    }

    System.out.println("Displaying data:\n");

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

        finalMarks[i].calculateFinalMarks();

        finalMarks[i].displayFinalMarks();

    }

}

}

```

*File 2: Internals.java (Inside package CIE)*

```

package CIE;

import java.util.Scanner;

public class Internals extends Student{

    protected int marks[] = new int[5];

    public void inputCIEmarks(){

        Scanner s = new Scanner(System.in);

        System.out.println("Enter the marks for 5 subjects");

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

            System.out.println("Enter the marks for subject "+(i+1)+": ");

            marks[i] = s.nextInt();

        }

    }

}

```

*File 3: Student.java (Inside package CIE)*

```

package CIE;

import java.util.Scanner;

public class Student{

    protected String usn = new String();

    protected String name = new String();

    protected int sem;

    public void inputStudentDetails(){

}

```

```

System.out.println("Enter Details of students:");
Scanner s=new Scanner(System.in);
System.out.println("Enter USN:");
usn=s.nextLine();
System.out.println("Enter Name:");
name=s.nextLine();
System.out.println("Enter Semester:");
sem=s.nextInt();
}

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

```

*File 4: Externals.java (Inside package SEE)*

```

package SEE;
import CIE.Internals;
import java.util.Scanner;
public class Externals extends Internals{
    protected int marks[];
    protected int finalMarks[];
    public Externals(){
        marks=new int[5];
        finalMarks=new int[5];
    }
    public void inputSEEmarks(){
        Scanner s=new Scanner(System.in);
        for(int i=0;i<5;i++){
            System.out.print("Subject"+(i+1)+" marks:");
        }
    }
}

```

```

        marks[i]=s.nextInt();
    }

}

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

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

```

Output:

```

Enter Details of students:
Enter USN:
1BM22CS001
Enter Name:
ABC
Enter Semester:
3
Enter CIE marks
Enter the marks for 5 subjects
Enter the marks for subject 1:
39
Enter the marks for subject 2:
40
Enter the marks for subject 3:
38
Enter the marks for subject 4:
40
Enter the marks for subject 5:
36
Enter SEE marks
Subject1 marks:97
Subject2 marks:92
Subject3 marks:95
Subject4 marks:89
Subject5 marks:78

```

```

Displaying data:

USN: 1BM22CS001
Name: ABC
Semester: 3
Subject1: 87
Subject2: 86
Subject3: 85
Subject4: 84
Subject5: 75

```

## **Lab Program 7:**

Q) Write a java program to demonstrate Exception Handling.

Solution:

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

```
class WrongAge extends Exception{
```

```
    public WrongAge(){  
        super("Age Error");  
    }
```

```
    public WrongAge(String message){
```

```
        super(message);  
    }
```

```
}
```

```
class InputScanner{
```

```
    public static int getIntInput(String prompt){  
        Scanner scanner=new Scanner(System.in);  
        System.out.print(prompt);  
        return scanner.nextInt();  
    }
```

```
}
```

```
class Father extends InputScanner{
```

```
    public int fatherAge;
```

```
    public Father() throws WrongAge{
```

```
        fatherAge=getIntInput("Enter Father's age: ");
```

```
        if(fatherAge<0){
```

```
            throw new WrongAge("Age cannot be negative");
```

```
}
```

```

    }

    public void display(){
        System.out.println("Father's Age: "+fatherAge);
    }

}

class Son extends Father{

    private int sonAge;

    public Son() throws WrongAge{
        super();
        sonAge=getIntInput("Enter Son's age: ");
        if(sonAge > super.fatherAge){
            throw new WrongAge("Son's age cannot be greater than father's age");
        }
        else if(sonAge<0){
            throw new WrongAge("Age cannot be negative");
        }
        else if(sonAge==super.fatherAge){
            throw new WrongAge("Son's age cannot be equal to father's age");
        }
    }

    public void display(){
        super.display();
        System.out.println("Son's Age: "+sonAge);
    }

}

public class ExceptionHandling{

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

```

```

        Son son=new Son();
        son.display();
    }

    catch(WrongAge e){
        System.err.println("Error: "+e.getMessage());
    }
}
}
}

```

Output:

```

Enter Father's age: 54
Enter Son's age: 19
Father's Age: 54
Son's Age: 19

Enter Father's age: 19
Enter Son's age: 54
Error: Son's age cannot be greater than father's age

Enter Father's age: -2
Error: Age cannot be negative

Enter Father's age: 12
Enter Son's age: 12
Error: Son's age cannot be equal to father's age

Enter Father's age: 45
Enter Son's age: -3
Error: Age cannot be negative

```

## Lab Program 8:

Q) Write a java program to work with threads.

Solution:

```

class MessageThread extends Thread{
    private final String message;
    private final long interval;

```

```

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

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

public class TwoThreadDemo{
    public static void main(String[] args){
        DisplayMessageThread thread1=new DisplayMessageThread("BMS College of
Engineering", 10000);
        DisplayMessageThread thread2=new DisplayMessageThread("CSE", 2000);
        thread1.setName("Thread 1");
        thread2.setName("Thread 2");
        thread1.start();
        thread2.start();
        try{
            Thread.sleep(20000);
        } catch(InterruptedException e){
            System.out.println("Main thread interrupted.");
        }
    }
}

```

```

    }
    thread1.interrupt();
    thread2.interrupt();
    System.out.println("Main thread exiting.");
}

```

Output:

```

BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
Main thread exiting.
Thread 2 interrupted.
Thread 1 interrupted.

```

### Lab Program 9:

Q) Write a java program to demonstrate events.

Solution:

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class SwingDemo{
    SwingDemo(){
        //create jframe container
        JFrame jfrm=new JFrame("Divider App");
        jfrm.setSize(275,150);
        jfrm.setLayout(new FlowLayout());
        // to terminate on close
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

```

```

// text label
JLabel jlab=new JLabel("Enter the divider and divident:");
// add text field for both numbers
JTextField ajtf=new JTextField(8);
JTextField bjtf=new JTextField(8);
// calc button
JButton button=new JButton("Calculate");
// labels
JLabel err=new JLabel();
JLabel alab=new JLabel();
JLabel blab=new JLabel();
JLabel anslab=new JLabel();
// add in order
jfrm.add(err);//to display error bois
jfrm.add(jlab);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);
ActionListener l=new ActionListener(){
    public void actionPerformed(ActionEvent evt){
        System.out.println("Action event from a text field");
    }
};
ajtf.addActionListener(l);
bjtf.addActionListener(l);
button.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent evt){

```

```

try{
    int a=Integer.parseInt(ajtf.getText());
    int b=Integer.parseInt(bjtf.getText());
    int ans=a/b;
    alab.setText("\nA = "+a);
    blab.setText("\nB = "+b);
    anslab.setText("\nAns = "+ ans);
}

catch(NumberFormatException e){
    alab.setText("");
    blab.setText("");
    anslab.setText("");
    err.setText("Enter Only Integers!");
}

catch(ArithmaticException e){
    alab.setText("");
    blab.setText("");
    anslab.setText("");
    err.setText("B should be NON zero!");
}

}

});

// display frame
jfrm.setVisible(true);
}

public static void main(String args[]){
    // create frame on event dispatching thread
    SwingUtilities.invokeLater(new Runnable(){

        public void run(){
            new SwingDemo();
        }
    });
}

```

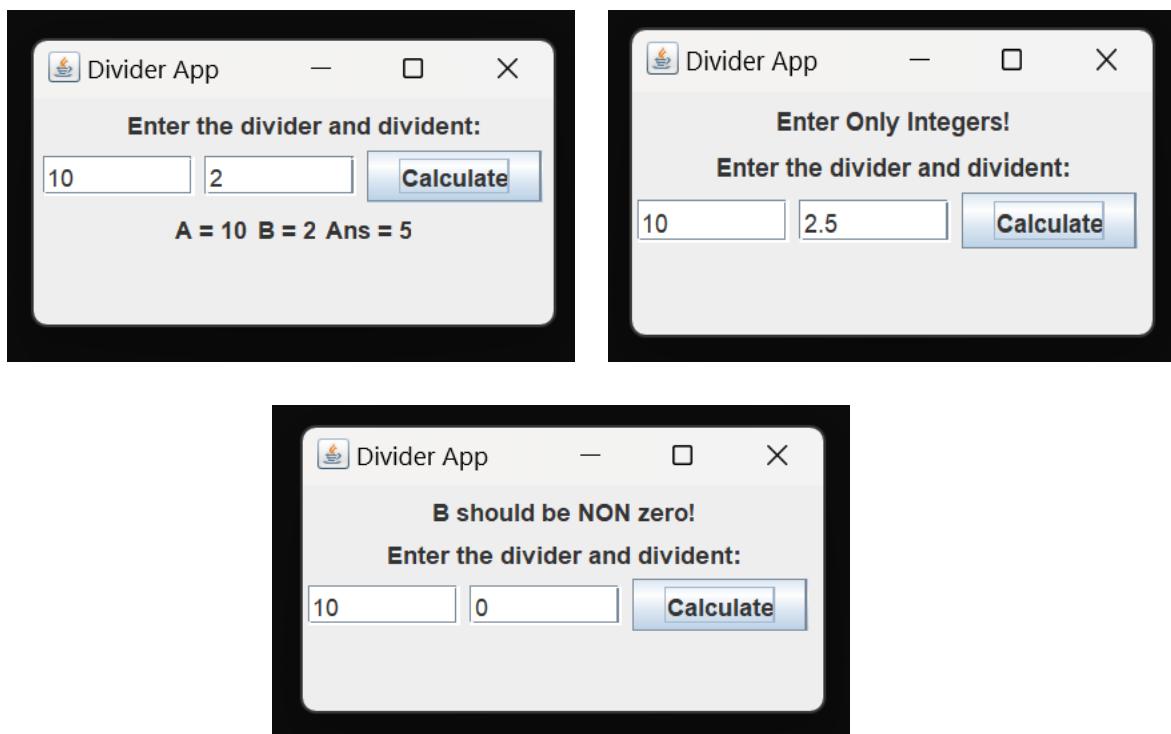
```

    }
});

}
}

```

Output:



### Lab Program 10 [Part 1]:

Q) Write a java program to demonstrate inter process communication.

Solution:

```

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<5){
            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<5){
            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.");
    }
}

```

Output:

```

Press Control-C to stop.
Put: 0

Intimate Consumer

Producer waiting

Got: 0

Intimate Producer

Put: 1
consumed:0

Intimate Consumer

Producer waiting

Got: 1

Intimate Producer

consumed:1
Put: 2

Intimate Consumer

Producer waiting

```

```

Got: 2

Intimate Producer

consumed:2
Put: 3

Intimate Consumer

Producer waiting

Got: 3

Intimate Producer

consumed:3
Put: 4

Intimate Consumer

Got: 4

Intimate Producer

consumed:4

```

## Lab Program 10 [Part 2]:

Q) Write a java program to demonstrate deadlock.

Solution:

```

class A{
    synchronized void foo(B b){
        String name=Thread.currentThread().getName();

```

```

System.out.println(name + " entered A.foo");
try{
    Thread.sleep(1000);
}
catch(Exception e){
    System.out.println("A Interrupted");
}
System.out.println(name+" trying to call B.last()");
b.last();
}

void last(){
    System.out.println("Inside A.last");
}
}

class B{
    synchronized void bar(A a){
        String name=Thread.currentThread().getName();
        System.out.println(name + " entered B.bar");
        try{
            Thread.sleep(1000);
        }
        catch(Exception e){
            System.out.println("B Interrupted");
        }
        System.out.println(name + " trying to call A.last()");
        a.last();
    }

    void last(){
        System.out.println("Inside A.last");
    }
}

```

```

}

class Deadlock implements Runnable{

    A a=new A();
    B b=new B();

    Deadlock(){
        Thread.currentThread().setName("MainThread");
        Thread t=new Thread(this,"RacingThread");
        t.start();
        a.foo(b); //get lock on a in this thread.
        System.out.println("Back in main thread");
    }

    public void run(){
        b.bar(a); //get lock on b in other thread.
        System.out.println("Back in other thread");
    }
}

public static void main(String args[]){
    new Deadlock();
}

```

Output:

```

MainThread entered A.foo
RacingThread entered B.bar
RacingThread trying to call A.last()
MainThread trying to call B.last()
Inside A.last
Inside A.last
Back in other thread
Back in main thread

```



## Lab Program 1:

- Q. Program to display the roots of quadratic equation.

Ans:

```
import java.util.Scanner  
class Quadratic {  
    int a, b, c;  
    double r1, r2, d;  
    void getd() {  
        Scanner s = new Scanner(System.in);  
        System.out.println("Enter the coefficients a,b,c");  
        a = s.nextInt();  
        b = s.nextInt();  
        c = s.nextInt();  
    }  
    void compute() {  
        while (a == 0) {  
            System.out.println("Not a quadratic equation");  
            System.out.println("Enter a non-zero value for a:");  
            Scanner s = new Scanner(System.in);  
            a = s.nextInt();  
        }  
    }  
}
```

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

if ( $d == 0$ ) {

$$r1 = (-b) / (2 * a) ;$$

System.out.println ("Roots are real and equal");

System.out.println ("Root 1 = Root 2 = " + r1);

}

else if ( $d > 0$ ) {

$$r1 = ((-b) + (\text{Math.sqrt}(d))) / (\text{double})(2 * a) ;$$

$$r2 = ((-b) - (\text{Math.sqrt}(d))) / (\text{double})(2 * a) ;$$

System.out.println ("Roots are real and distinct");

System.out.println ("Root 1 = " + r1 + "Root 2 = " + r2);

}

else if ( $d < 0$ ) {

System.out.println ("Roots are imaginary");

$$r1 = (-b) / (2 * a) ;$$

System.out.println ("Root 1 = " + r1 + " + i " + r2);

System.out.println ("Root 2 = " + r1 + " - i " + r2);

}

}

}

class QuadraticMain {

public static void main (String args[]) {

Quadratic q = new Quadratic();

q.getd();

q.compute();

}

}

Output:

Enter the coefficients a, b, c :

~~Roots are Imaginary~~

~~Roots are Real~~

(i) 0 1 2

Not a quadratic equation

Enter a non-zero value of a:

(ii) 1 -2 1

Roots are real and equal

The roots are +1 and +1

~~Roots are Real~~

(iii) 1 2 10

Roots are imaginary

$$\text{Root 1} = -1.0 + i\sqrt{18}$$

$$\text{Root 2} = -1.0 - i\sqrt{18}$$

~~Roots are Imaginary~~

## Lab Program 2:

- Q. Program in Java to calculate SGPA of a student.

Ans:

```
import java.util.Scanner;
class Subject {
    int subjectMarks;
    int credits;
    int grade;
}
```

```
class Student {
    Subject subject[];
    String name;
    String usn;
    double SGPA;
    Scanner s;
    Student () {
        int i;
        subject = new Subject[9];
        for (i=0; i<9; i++)
            subject[i] = new Subject();
        s = new Scanner(System.in);
    }
}
```

```
void getStudentDetails () {
    System.out.println ("Enter your name: ");
    name = s.next();
    System.out.println ("Enter your USN: ");
    usn = s.next();
}
```

```
void getMarks () {  
    for (int i=0 ; i < 9 ; i++) {  
        System.out.println ("Enter marks for subject " + (i+1)  
                           + ": ");  
        subject[i].subjectMarks = s.nextInt ();  
        System.out.println ("Enter the credits for  
                           subject " + (i+1) + "=" );  
        subject[i].credits = s.nextInt ();  
        subject[i].grade = (subject[i].subjectMarks / 10)  
                           + 1;  
        if (subject[i].grade == 11)  
            subject[i].grade = 10;  
        if (subject[i].grade <= 4)  
            subject[i].grade = 0;  
    }  
}
```

```
void computeSGPA () {  
    int effectiveScore = 0;  
    int totalCredits = 0;  
    for (int i=0 ; i < 9 ; i++) {  
        effectiveScore += (subject[i].grade *  
                           subject[i].credits);  
        totalCredits += subject[i].credits;  
    }  
    SGPA = (double) effectiveScore / (double) totalCredits;  
}
```

~~3~~

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

```
sl.getMarks();  
sl.computeSGPA();  
System.out.println("Name of Student: " + sl.name);  
System.out.println("USN of Student: " + sl.usn);  
System.out.println("SGPA of the Student: " + sl.SGPA);
```

3.

### Output :

Enter your Name:

Name

Enter your USN:

IBM 22CS599

Enter marks for subject 1:

98

Enter the credits for subject 1:

4

Enter marks for subject 2:

82 92

Enter the credits for subject 2:

4

Enter marks for subject 3:

86

Enter the credits for subject 3:

3

Enter ~~86~~ marks for subject 4:

92

Enter the credits for ~~subject 4~~ subject 4:

3

Enter marks for subject 5:

97

Enter the credits for subject 5:

3

Enter marks for subject 6:

98

Enter the credits for subject 6:

1

Enter marks for subject 7:

99

Enter the credits for subject 7:

1

Enter marks for subject 8:

95

Enter the credits for subject 8:

1

Enter marks for subject 9:

87

Enter the credits for subject 9:

1

Name of Student : ~~Nikhil~~ Name

USN of Student : IBMA2CS599

SGPA of the student : 9.80952380952381

02/02/2023  
10/12/2023

26/12/2023

Lab Program 3:

Q. Program in Java to display book details.

Ans. import java.util.Scanner;  
 class Books {

```
private String name;
private String author;
private int price;
private int numPages;
```

```
public Books (String name, String author, int price,
int numPages) {
```

```
this.name = name;
```

```
this.author = author;
```

```
this.price = price;
```

```
this.numPages = numPages;
```

{}

```
public String toString () {
```

```
String name, author, price, numPages;
```

```
name = "Book name : " + this.name + "\n";
```

```
author = "Author name : " + this.author + "\n";
```

```
price = "Price : " + this.price + "\n";
```

```
numPages = "Number of pages : " + this.numPages + "\n";
```

```
return name + author + price + numPages;
```

{}

~~class BookMain {~~

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

~~Scanner s = new Scanner (System.in);~~

~~int n;~~

~~System.out.println ("Enter the number of books : ");~~

~~int~~

```
n = s.nextInt();  
Books b[] = new Books[n];  
for(int i=0; i<n; i++) {  
    System.out.println("Enter details for book " + (i+1) + ":");  
    System.out.print("Name : ");  
    String name1 = s.next();  
    System.out.print("Author: ");  
    String author = s.next();  
    System.out.print("Price ");  
    int price = s.nextInt();  
    System.out.print("Number of pages : ");  
    int num_pages = s.nextInt();  
    b[i] = new Books(name1, author, price, num_pages);  
}
```

```
System.out.println("In Details of the Books:");  
for(int i=0; i<n; i++) {  
    System.out.println("In Book " + (i+1) + "\n" + b[i]);  
}
```

s.close();

## Output:

Enter the number of books :

32

~~Enter details for book 1 :~~

Name : LittleWomen

Author : Namel

Price : 360

Number of pages : 320

~~Enter details for book 2 :~~

Name : Thumkolina

Author : Name2

Price : 500

Number of Pages : 540

### Details of the books :

#### Book 1 :

Book name : Little Woman

Author ~~name~~ name : Name1

Price : 360

Number of Pages : 320

#### Book 2 :

Book name : Thumliolina

Author name : Name2

Price : 500

Number of Pages : 540

Ans.  
26/12/2023

02/01/2023

## Lab Program 4:

- Q. Java program to create an abstract class named shape that contains two integers and an empty method named printArea()...

Ans: import java.util.Scanner;

~~abstract class Shape {  
protected int d1;  
protected int d2;  
public abstract void printArea();  
public void setDimensions(int d1, int d2);}~~

class InputScanner {  
Scanner s = new Scanner(System.in);  
double getInput (String prompt) {  
System.out.println(prompt);  
return s.nextDouble();  
}

~~abstract class Shape extends InputScanner {  
double d1, d2;  
abstract void area();}~~

class Rectangle extends Shape {  
Rectangle () {

~~d1 = getInput ("Enter the length of the rectangle : ");  
d2 = getInput ("Enter the breadth of the rectangle : ");~~

void area () {

    double area = s1 \* s2 ;

    System.out.println ("Area of the rectangle = " + area);

}

3  
class Triangle extends Shape {

Triangle () {

    s1 = getInPut ("Enter the base of the triangle : ");

    s2 = getInPut ("Enter the height of the triangle : ");

}

void area () {

    double area = s1 \* s2 / 2 ;

    System.out.println ("Area of the triangle = " + area);

3

class Circle extends Shape {

Circle () {

    s1 = getInPut ("Enter the radius of the circle : ");

3

void area () {

    double area = Math.PI \* s1 \* s1 ;

    System.out.println ("Area of the circle = " + area);

3

class Main {

    public static void main (String args []) {

        Rectangle r1 = new Rectangle ();

        Triangle t = new Triangle ();

        Circle c = new Circle ();

        r1.area ();

t.area();  
c.area();

3  
3

Output :

Enter the length of the rectangle :

24

Enter the breadth of the rectangle :

3

Enter the base of the triangle :

6

Enter the height of the triangle :

3

Enter the radius of the circle :

4

Area of the Rectangle = 12.0

Area of the Triangle = 9.0

Area of the circle = 50.26548245



## Lab Programs :-

- Program to demonstrate string constructors.

Ans. ~~class StringConstructors {~~  
~~public static void main (String args [ ]) {~~  
~~String s1 = "Hello";~~  
~~System.out.println (s1);~~  
~~char chars [ ] = { 'a', 'b', 'c', 'd' };~~  
~~Str~~

Output :

Hello  
abc  
Hello  
Hell

- Program to demonstrate string length, string literal and string concat.

Ans. Output :

5

Hello, Nikhilesh !  
Hello Nikhilesh

- Program to demonstrate `toString()`.

Ans. Output :

Dimensions are 10.0 by 14.0 by 12.0  
box b : Dimensions are 10.0 by 14.0 by 12.0

4. Program to extract a substring using getchars()

Ans: Output:

Extracted substring : Bmsce

5. Program to demonstrate getbytes() and toCharArray()

Ans: Output:

Byte Array : 72 101 108 108 111 114 108 100 33

Char Array : J a v a P r o g r a m m i n g

6. Program to demonstrate equals.

Ans: Output:

Bmsce equals Bmsce → true

Bmsce equals College → false

Bmsce equals BMSCE → false

Bmsce equalsIgnoreCase BMSCE → true.

7. Program to find substring using regionMatches().

Ans: Output:

Substring is matched

Substring is not matched.

8. Program to demonstrate startWith()

Ans: Output:

Starts with 'Hello' : true  
 Starts with 'hi' : false.

9. Program to demonstrate endsWith().

Ans. Output :

Ends with "World" : true  
 Ends with 'Java' : false

10. Program to demonstrate equals() versus == .

Ans. Hello equals Hello → true  
 Hello == Hello → false

of alphabets

11. Program to perform sorting, using compareTo().

Ans. Sorted alphabets :

apple, ball, cat, dog, ent, free, gun, hen,  
 ice, jug, kite, lift, man, net, orange, haricot,  
 queen, ring, star, tree, ~~umbrella~~, van, watch,  
 xmas, yatch, zel.

12. Program to sort # numbers from 10 to 1 using  
 compareTo(). 17/01/2024

Verified: 1 10 2 3 4 5 6 7 8 9

Ans. Output :

Output

1 10 2

x 1 2 3 4 5 6 7 8 9 10

13. Program to demonstrate usage of substring(),  
 indexOf(), + for replacing "was" to "is" in  
 "Thwas was a test. Thwas was, too."

Ans: Output:

Modified String: This is a test. This is too.

14. Java program to demonstrate concat().

Ans: Output:

helloworld

15. Program to demonstrate replace()

Ans: Output:

Modified string: Welcome to BMSCE commge.

16. Program to demonstrate trim()

Ans: Output:

Trimmed String: [Hello Friends]

17. Enter the details for student 1:

Registration number: 10

Full name: Navya Billalar

Semester: 3

CGPA: 8.9

Enter details for student 2:

Registration number: 102

Full name: Nikita B

Semester: 3

CGPA: 8.8

Enter details for student 3:

✓ pending  
Re  
16/11/2022

Registration number : 103

Full name : Nikhilesh

Semester : 3

CGPA : 9.3

continued after (20)

18. Program to demonstrate string buffer functions.

Ans: Output :

After setlength (5) : Hello

Character at index 1 : e

After setcharat (7, 'M') : Hello, Morld !

F

Extracted characters : Hello

After append (" , World!") : Hello, World !

After insert (5, " , World!") : Hello, World !

After reverse : ! drow , olleh

After delete (7,12) : Hello !

After deleteCharAt (5) : Hello World !

After replace (7,12, "universe") : Hello, universe

Extracted substring (7,12) : Universe

19. Program to create an abstract class bird . . .

Ans: Output :

Eagle : Eagle flies high in the sky  
Eagle screeches loudly

Hawk : Hawk soars gracefully through the air,  
Hawk emits piercing

20. Program to create an abstract class shape.

Ans: Output :

Circle - Area : 78.5398 ,  
Perimeter : 31.415

Triangle - Area : 6.0  
Perimeter : 12.0

17. Continuation of Output :

Enter details for student 4:

Registration Number : 104

Full Name : Ankit

Semester : 3

CGPA : 8.5

Enter details for student 5:

Registration number : 105

Full Name : Ben

Semester : 3

CGPA : 8.6

Student records :

Registration number : 10

Full Name : Navya Billalor

Semester : 3

Q8 Program to demonstrate Setlength(), Charat(), Setcharat(),  
getchar(), append(), Insert(), reverse(), delete(),  
Deletecharat(), Replace(), substring().

Ans:

Output:

CGPA : 8.9

Registration number : 102

Student records (sorted by CGPA) :

Registration Number : 103

Full Name : Nikhilash

Registration Number : 10

Registration Number : 102

Registration Number : 105

Registration Number : 104

Student Records (sorted by Name) :

Registration Number : 104

Full Name : Ankit

Full Name : Ben

Full name : Navya Billalas

Full name : Nikhilesh

Full name : Nikita

By  
22/11/2020

23/01/2024

Lab 6 :

Q. Program to implement packages.

Ans (i) Student.java:

```
package CIE;
import java.util.Scanner;
public class Student {
    protected String usn = new String();
    protected String name = new String();
    protected int sem;
    public void inputStudentDetails() {
        System.out.println("Enter Details of Students:");
        Scanner s = new Scanner(System.in);
        System.out.println("Enter USN:");
        usn = s.nextLine();
        System.out.println("Enter name:");
        name = s.nextLine();
        System.out.println("Enter Semester:");
        sem = s.nextInt();
    }
}
```

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

(ii) Internals.java:

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

```

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

```

### (iii.) Externals.java :

```

package SEE;
import CIE.Internals;
import java.util.Scanner;
public class Externals extends Internals {
    protected int marks[];
    protected int finalMarks[];
    public Externals() {
        marks = new int[5];
        finalMarks = new int[5];
    }
    public void inputSEEmarks() {
        Scanner s = new Scanner(System.in);
        for (int i = 0; i < 5; i++) {
            System.out.println("Subject " + (i + 1) + " marks:");
            marks[i] = s.nextInt();
        }
    }
    public void calculateFinalMarks() {
}
}

```

```
for (int i=0; i<5; i++)  
    finalMarks[i] = marks[i]/2 + super_marks[i];  
  
public void displayFinalMarks () {  
    displayStudentDetails ();  
    for (int i=0; i<5; i++)  
        System.out.println ("Subject " + (i+1) + ":" +  
                           finalMarks[i]);  
}
```

(iv.) Main.java:

```
import SEE.Externals;  
class Main {  
    public static void main (String args []) {  
        int numofStudents = 5;  
        Externals finalMarks[] = new Externals [numofStudents];  
        for (int i=0; i<numofStudents; i++) {  
            finalMarks[i] = new Externals ();  
            finalMarks[i].inputStudentDetails ();  
            System.out.println ("Enter CIE marks");  
            finalMarks[i].inputCIEmarks ();  
            System.out.println ("Enter SEE marks");  
            finalMarks[i].inputSEEmarks ();  
  
            System.out.println ("Displaying data:\n");  
            for (int i=0; i<numofStudents; i++) {  
                finalMarks[i].calculateFinalMarks ();  
                finalMarks[i].displayFinalMarks ();  
            }  
        }  
    }  
}
```

Output:

Enter details of student:

Enter USN:

IBM22 CS181

Enter Name:

Nikhilesh C

Enter Semester:

3

Enter CIE marks

Enter the marks for subject 1:

39

Enter the marks for subject 2:

40

Enter the marks for subject 3:

38

Enter the marks for subject 4:

40

Enter the marks for subject 5:

36

Enter SEE marks

Subject 1 marks: 97

Subject 2 marks: 92

Subject 3 marks: 95

Subject 4 marks: 89

Subject 5 marks: 78

~~Enter details of student~~ Displaying Data:

USN: IBM22 CS181

Name: Nikhilesh C

Semester: 3

Subject 1 : 87  
Subject 2 : 86  
Subject 3 : 85  
Subject 4 : 84  
Subject 5 : 75

R  
22/11/2021

30/01/2021

## Lab Program 7

Q. Java Program to demonstrate Exception handling.

Ans. import java.util.Scanner;

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

```
class InputScanner {  
    public static int getInput(String prompt) {  
        Scanner s = new Scanner(System.in);  
        System.out.print(prompt);  
        return s.nextInt();  
    }  
}
```

```
class Father extends InputScanner {  
    public int fatherAge;  
    public Father() throws WrongAge {  
        fatherAge = getInput("Enter Father's age : ");  
        if (fatherAge < 0) {  
            throw new WrongAge("Age cannot be negative");  
        }  
    }  
    public void display() {  
        System.out.println("Father's age: " + fatherAge);  
    }  
}
```

class Son extends Father {

private int sonAge;

public Son() throws WrongAge {  
super();

sonAge = getInPut ("Enter son's age: ");

if (sonAge > super.fatherAge) {

throw new WrongAge ("Son's age cannot be greater  
than father's age");

{

else if (sonAge < 0) {

{

throw new WrongAge ("Age cannot be negative");

{

{

public void display() {

super.display();

System.out.println ("Son's age: " + sonAge);

{

public class ExceptionHandling {

public static void main (String args[]) {

try {

Son son = new Son();

son.display();

{

catch (WrongAge e) {

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

{

{

## Output :

- (i) Enter Father's age : 54  
Enter Son's age : 19  
Father's age : 54  
Son's age : 19
- (ii) Enter Father's age : 19  
Enter Son's age : 54  
Error : Son's age cannot be greater than father's age
- (iii) Enter Father's age : -2  
Error : Age cannot be negative
- (iv) Enter Father's age : 12  
Enter Son's age : 12  
Error : Son's age cannot be equal to father's age
- (v) Enter Father's age : 45  
Enter Son's age : -3  
Error : Age cannot be negative

90  
30 1/2m

## Lab Program 8

Q. # Java program to work with threads.

Ans: ~~class extends Thread~~

```
class MessageThread extends Thread {
```

```
    private final String message;
```

```
    private final long interval;
```

```
    display
```

```
MessageThread (String message, long interval) {
```

```
    this.message = message;
```

```
    this.interval = interval;
```

```
}
```

```
public void run () {
```

```
    try {
```

```
        while (true) {
```

```
            System.out.println(message);
```

```
            Thread.sleep(interval);
```

```
}
```

```
}
```

```
catch (InterruptedException e) {
```

```
    System.out.println(Thread.currentThread().getName() +
```

```
        " interrupted.");
```

```
}
```

```
}
```

~~public class TwoThreadDemo {~~

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

~~MessageThread t1 = new MessageThread ("BMS College of  
Engineering", 10000);~~

~~MessageThread t2 = new MessageThread ("CSE", 2000);~~

```
t1.setName ("Thread 1");
t2.setName ("Thread 2");
t1.start;
t2.start;
try {
    Thread.sleep(20000);
}
catch (InterruptedException e) {
    System.out.println ("Main thread interrupted");
}
t1.interrupt();
t2.interrupt();
System.out.println ("Main thread exiting");
}
```

### Output:

BMS College of Engineering  
CSE → After 2 seconds

CSE → "

CSE → "

CSE → "

BMS College of Engineering → After 10 seconds  
CSE → After 2 seconds

CSE → "

CSE → "

CSE → "

CSE → "

Main thread exiting → After 20 seconds

Thread 2 interrupted

Thread 1 interrupted.

06/02/2024

Lab Program 10

- Java Program to demonstrate inter process communication.

Ans.

```

class Q {
    int n;
    boolean valueSet = false;
    synchronized int get() {
        while (!valueSet)
            try {
                System.out.println("In Consumer Waiting\n");
                wait();
            } catch (InterruptedException e) {
                System.out.println("InterruptedException caught");
            }
        System.out.println("Got: " + n);
        valueSet = false;
        System.out.println("In Intimate Producer\n");
        notify();
        return n;
    }
    synchronized void put(int n) {
        while (valueSet)
            try {
                System.out.println("In Producer 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 {
```

```
    Queue q;
```

```
    Producer(Q q) {
```

```
        this.q = q;
```

```
        new Thread(this, "Producer").start();
```

{}

```
    public void run() {
```

```
        int i = 0;
```

```
        while (i < 5) {
```

```
            q.put(i++);
```

{}

{}

{}

```
class Consumer implements Runnable {
```

```
    Queue q;
```

```
    Consumer(Q q) {
```

```
        this.q = q;
```

```
        new Thread(this, "Consumer").start();
```

{}

```
    public void run() {
```

```
        int i = 0;
```

```
        while (i < 5) {
```

```
            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");

}

}

Output :

Put : 0

Intimate Consumer

Producer waiting

Press Control-C to stop

Got : 0

Intimate Producer

Consumed : 0

Put : 1

Intimate consumer

Producer waiting

Got : 1

Intimate Producer

consumed : 1

Put : 2

Intimate Consumer

Producer waiting

Got : 2

Intimate Producer

consumed : 2

Put : 3

Intimate Consumer

Producer waiting

Got : 3

Intimate Producer

Put : 4

Intimate Consumer

Consumed : 3

Got : 4

Intimate Producer

consumed : 4

By  
6/2/2020

19/02/2024

## Lab Program 9

Q. Java Program to demonstrate events.

Ans.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class SwingDemo {
    SwingDemo() {
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JLabel jl1 = new JLabel("Enter the divider and dividend:");
        JTextField jt1 = new JTextField(8);
        JTextField jt2 = new JTextField(8);
        JButton button = new JButton("Calculate");
        JLabel err = new JLabel();
        JLabel alab = new JLabel();
        JLabel blab = new JLabel();
        JLabel anslab = new JLabel();
        jfrm.add(err);
        jfrm.add(jl1);
        jfrm.add(jt1);
        jfrm.add(jt2);
        jfrm.add(button);
        jfrm.add(alab);
        jfrm.add(blab);
        jfrm.add(anslab);
        ActionListener l = new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                System.out.println("Action event from a text field");
            }
        };
        jt1.addActionListener(l);
        jt2.addActionListener(l);
    }
}
```

```
ajtf.addActionListener(l);  
bjtf.addActionListener(l);  
button.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent evt) {  
        try {  
            int a = Integer.parseInt(ajtf.getText());  
            int b = Integer.parseInt(bjtf.getText());  
            int ans = a/b;  
            anslab  
            alab.setText("In A = " + a);  
            blab.setText("In B = " + b);  
            anslab.setText("In Ans = " + ans);  
        }  
    }  
});
```

```
catch (NumberFormatException e) {  
    alab.setText(" ");  
    blab.setText(" ");  
    anslab.setText(" ");  
    err.setText("Enter Only Integers!");  
}
```

```
catch (ArithmaticException e) {  
    alab.setText(" ");  
    blab.setText(" ");  
    anslab.setText(" ");  
    err.setText("B should be NON zero!");  
}
```

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

```
public static void main (String args[]) {  
    SwingUtilities.invokeLater(new Runnable() {  
        public void run () {  
            new SwingDemo();  
        }  
    });
```

? );  
?

Output :

Divider App

Enter the divider and dividend

10	2
calculate	

Upon Clicking Calculate Button.

Divider App

Enter the divider and dividend

10	2
calculate	

A = 10   B = 2   Ans = 5

error label  
comes here in  
case of catch blank

Report on Functions Used :

1. JFrame :

- \* Part of the Java Swing library.
- \* Provides a set of components for building desktop applications.

Re=2012/24

2. setSize :

- \* A method in the Java Swing library.
- \* Used to set the dimensions (width and height) of a component.

### 3. setLayout:

- \* Used to set the layout manager for a container.

### 4. setDefaultCloseOperation:

- \* Used to set the default operation that should happen when the user closes a window.

### 5. Label:

- \* Used to display a single line of non-editable text or an image.

### 6. JTextField:

- \* A component that allows the user to enter and edit a single line of text.

### 7. addFrame:

- \* Method in the 'Container' class.
  - JFrame extends this class
- \* Adds a component to the ~~container~~.

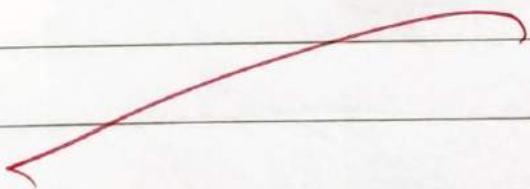
### 8. addActionListener:

- \* An interface that responds to events, such as button clicks.

- \* JButton extends this.
  - It adds an Action Listener to a button.

9. setText:

- \* Method which sets the text that is displayed by the label.



Pur  
setText

13/02/2024

## 2. Java program to demonstrate deadlock.

Ans.

```
class A {
    synchronized void demo (B b) {
        String name = Thread.currentThread().getName();
        System.out.println(name + " entered A.demo");
        try {
            Thread.sleep(1000);
        } catch (Exception e) {
            System.out.println("A interrupted");
        }
        System.out.println(name + " trying to call B.last()");
        b.last();
    }
    void last() {
        System.out.println("Inside A.last");
    }
}
```

✓

```
class B {  
    synchronized void bar(A a) {  
        String name = Thread.currentThread().getName();  
        System.out.println("name + " entered B.bar");  
        try {  
            Thread.sleep(1000);  
        } catch (Exception e) {  
            System.out.println("B interrupted");  
        }  
        System.out.println(name + " trying to call A.last()");  
        a.last();  
    }  
    void last() {  
        System.out.println("Inside A.last()");  
    }  
}
```

```
class Deadlock implements Runnable {  
    A a = new A();  
    B b = new B();  
    Deadlock() {  
        Thread.currentThread().setName("MainThread");  
        Thread t = new Thread(this, "RacingThread");  
        t.start();  
        a.demo(b); // get lock on 'a' in this thread  
        System.out.println("Back in main thread");  
    }  
}
```

```
public void run() {  
    b.bar(a); // get lock on b in the other thread  
    System.out.println("Back in other thread");  
}
```

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

{}

Output:

~~RacingThread entered B. bar~~  
~~MainThread entered A. demo~~  
~~Inside A. last()~~  
~~Back in other thread~~  
~~Inside~~

MainThread entered A. demo  
RacingThread entered B. bar  
MainThread trying to call B. last()  
RacingThread trying to call A. last()  
Inside B. last()  
Back in main thread  
Inside A. last()  
Back in other thread

By  
13/2/2020