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:

NAVYA BILLALAR

(1BM22CS175)

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("Rootsd 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
Rootsd 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.

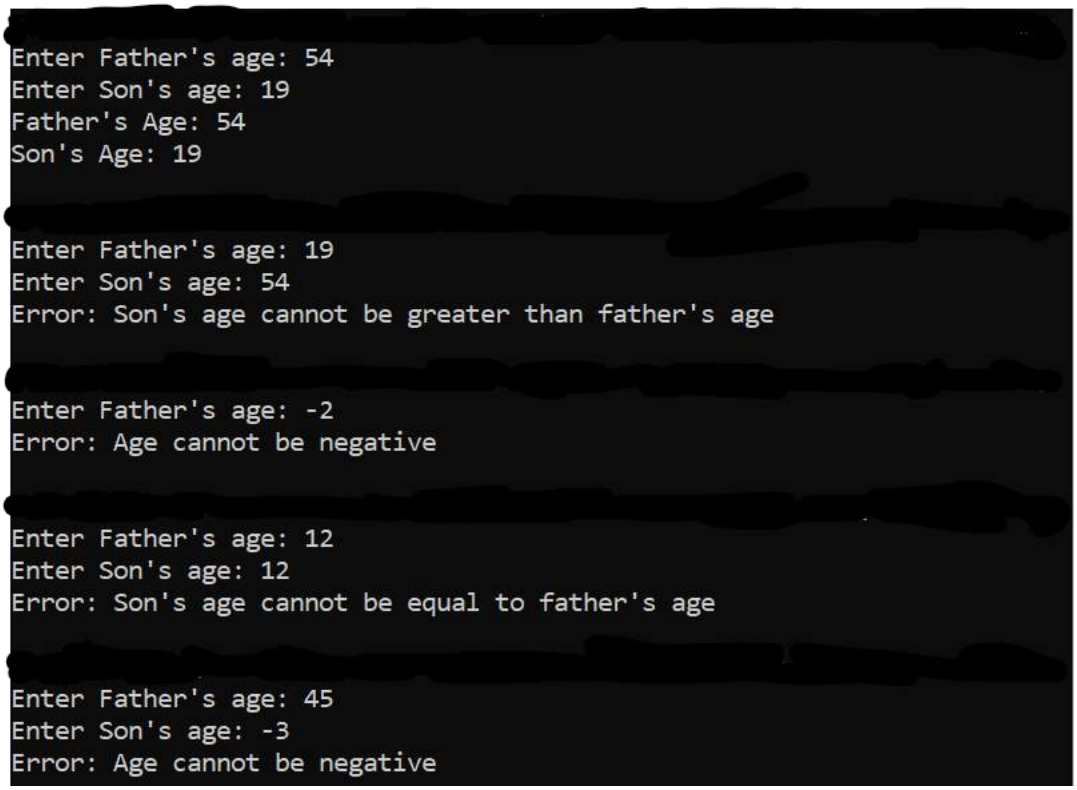
Solution:

```
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:



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
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:

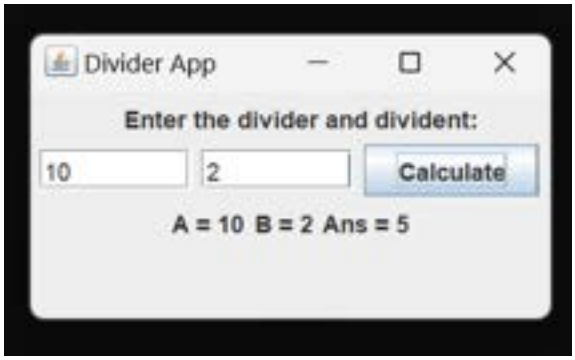
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class SwingDemo{
        SwingDemo(){
                //create jframe container
                JFrame jfrm=new JFrame("Divider App");
                jfrm.setSize(275,150);
                jfrm.setLayout(new FlowLayout());
      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

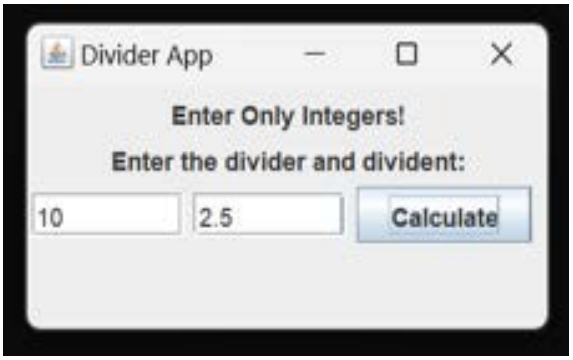
```
// text label
JLabel jlab=new JLabel("Enter the divider and divident:");
// add text field for both numbers
JTextField ajtf=new JTextField(8);
JTextField bjtf=new JTextField(8);
// calc button
JButton button=new JButton("Calculate");
// labels
JLabel err=new JLabel();
JLabel alab=new JLabel();
JLabel 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){
```

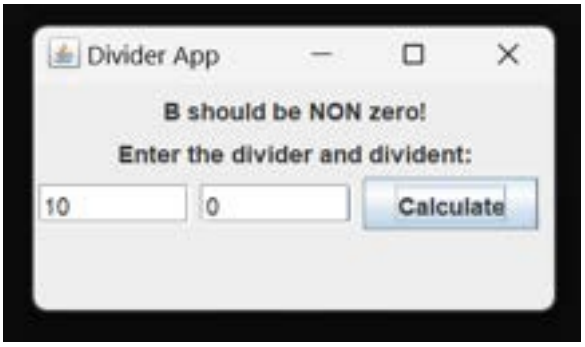
```
                    try{
                        int a=Integer.parseInt(ajtf.getText());
                        int b=Integer.parseInt(bjtf.getText());
                        int ans=a/b;
                        alab.setText("\nA = "+a);
                        blab.setText("\nB = "+b);
                        anslab.setText("\nAns = "+ ans);
                    }
                    catch(NumberFormatException e){
                        alab.setText("");
                        blab.setText("");
                        anslab.setText("");
                        err.setText("Enter Only Integers!");
                    }
                    catch(ArithmeticException e){
                        alab.setText("");
                        blab.setText("");
                        anslab.setText("");
                        err.setText("B should be NON zero!");
                    }
                }
            });
            // display frame
            jfrm.setVisible(true);
    }
    public static void main(String args[]){
            // 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 - 01

Program to find the area of a rectangle

~~Same~~

```
class RectangleArea {
public static void main (String args[]) {
int length, breadth;
length = Integer.parseInt (args[0]);
breadth = Integer.parseInt (args[1]);
int area = length * breadth;
System.out.println ("length of rectangle=" + length);
System.out.println ("Breadth of rectangle = " + breadth);
System.out.println (" Area of rectangle =" + area);
}}
```

2. Scanner:

```
import java.util.Scanner;
class HelloWorld {
public static void main (String args[])
 {
 int a; float b; String s;
 Scanner in = new Scanner (System.in);
 System.out.println ("Enter a string");
 s = in.nextLine();
 System.out.println ("You entered string "+s);
 System.out.println ("Enter an integer");
 a = in.nextInt();
 System.out.println ("You entered integer "+a);
 System.out.println ("Enter a float");
 b = in.nextFloat();
 System.out.println ("You entered float "+b);
}
}
```

3. Write a program to find factorial of a given number.

```
import java.util.Scanner;
class factorial {
public static void main(String args[])
{
    int fac = 1;
    System.out.println("Enter a number: ");
    Scanner sc = new Scanner(System.in);
    int n = sc.nextInt();
    for(int i=1; i<=n; i++){
        fac = fac * i;
}
 System.out.println("The factorial: " + fac);
}
}
```

Output:

4. Arrays

```
Import 'java.util.Scanner';
Class AutoArray {
public static void main (Strings args [ ]) {
int month_days [ ] = { 31, 28, 31, 30, 31, 30, 31, 31,
                       30, 31, 30, 31 };
System.out.println ("April has " + month_days [
  " days. " ); } }
```

Output:

April has 30 days.

5. Program to find the given 5 digit int is a palindrome or not.

```
Import java.util.Scanner;
class palindrome {
public static void (String args [ ])
{
int n , t, rem, rev = 0;
Scanner sc = new Scanner (System.in);
System.out.println (" Enter a 5 digit number:
n = sc.nextInt ();
t = n;
while (t > 0) {
rem = t % 10;
rev = rev * 10 + rem;
t = t / 10;
}
if (rev == n) {
```

```
System.out.println("Palindrome");
}
else {
System.out.println("not palindrome");
}
}
}
```

6. Program to find the sum of digits in 5 digit number

```
import java.util.Scanner;
class Sumofdigits {
public static void main(String args[]){
    long number, sum;
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter a 5-digit
      number:");
    number = sc.nextlong();
    for(sum = 0; number != 0; number =
            number/10){
        sum = sum + number%10;
    }
    System.out.println("Sum of digits:" + sum
}
}
```

7. Conversion:

```
Import java.util.Scanner;
class Conversion{
public ~~void m~~ static void main(String args[])
{
byte b=1; short s1=1000, s2;
int i1=100000, i2;
long l1=1000000, l2;
char c='+';
float f1=25.69f, f2;
double d1=536987.125, d2;
System.out.println(b+" "+s1+" "+i1+" "+
l1+" "+c+" "+f1+" "+d1);
s2=b;
i2=s1;
l2=i1;
System.out.println(s2+" "+i2+" "+l2);
}
}
```

8. Quadratic Equation:

```
Import java.util.Scanner
Class Quadratic
{
  int a,b,c;
  double r1,r2,d;
  void getd().
{
  Scanner sc = new scanner (System.in);
  System.out.println ("Enter the coefficients
        of 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:");
  System Scanner sc = new scanner (System.in);
    a = S.nextInt();
  }
  d=b*b-4*a*
  if (d==0)
  r1 = (-b)/(2*a);
  System.out.println("Roots are real & equal");
  System.out.println("Root1 = Root2 = "+r2);
}
```

```
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 = +"r1+" Root2
{                          = "+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 2 = +"r1+"
          +i" + r2");
    }
  }
}
Class Quadratic Main
  {
    public static void main(String args[])
  {
    Quadratic q = new quadratic();
    q.getd();
    q.compute();
  }
  }
```

Output:

Enter the coefficient ~~equation~~ a,b,c

Roots are real & distinct
Root1 = -0.381966 Root2 = -2.618

1 2 1
Roots are imaginary

0 0 0
Not a quadratic equation

1 0 0
Roots are real and equal
Root1 = Root2 0.0

(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

(iii) 1 2 10
Roots are imaginary
Root1 = -1.0 + i 18
Root2 = -1.0 - i 18

12/12/2023

2.23

LAB - 02

1) 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 subject {
        int subjectMark, credits, grades;
        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.next
    Int();
    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;
    }
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:
Navya
Enter your USN:
1BM22CS175
Enter marks for subject 1:
98
Enter credits for subject 1:
4
Enter marks for subject 2:
92
Enter credits for subject 2:
4
Enter marks for subject 3:
86
Enter credits for subject 3:
3
Enter marks for subject 4:
92
Enter credits for subject 4:
3
Enter marks for subject 5:
97
Enter credits for subject 5:
3
```

Enter marks for subject 6:
98
Enter ~~marks~~ credits four subject 6:
1
Enter marks four subject 7:
99
Enter credit four subject 7:
1
Enter marks four subject 8:
95
Enter credits four subject 8:
1
Enter marks four subject 9:
87
Enter credits four subject 9:
1
Name of student : Navya
USN of student : 1BH22CS175
SGPA of student : 9.88952

15/2/2023

Lab-03

Create a 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. Develop a Java program to create n book objects.

```
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.println ("Name: ");
            String name1 = s.nextLine ();
            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("\n Details of the
          Book: ");
        for (int i=0 ;i<n ;i++){
            System.out.println ("\n Book "+
          (i+1) +" : \n" +b[i]);
        }
        s.close();
    }
}
```

Output:

Enter the no. of books:
2
Enter details for book 1:
Name: littleWoman
Author: Name1
Price: 345
Number of pages: 500
Enter details for book 2:
Name: ghostvidu
Author: Name2
Price: 500
Number of pages: 600

Details of the books:

Book1:
Book name: littleWomen
Author: Name1
Price: 345
Number of pages: 500

Book2:
Book name: ghostvidu
Author: Name2.
Price: 500
Number of pages: 600

26/2/2023

Lab - 04

```
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 side1, side2;
        Abstract void area ();
Class Rectangle extends Shape {
        Rectang () {
          side1 = getInput ("enter length of
                  rectange: ");
          side2 = getInput ("enter breadth of
                  rectangle: ");
        void area () {
          double area = side1 * side2 ;
          System.out.println ("Area of the
                  rectange =" +area );
          }
        }
Class Triangle extends Shape {
          triangle () {
            side1 = getInput ("Enter the
            base of triangle: ");
            side2 = getInput ("Enter the
              height of the triangle: ");
            void area () {
              double area = side1 * side2 /
```

```
        System.out.println("Area of triangle: "
                + area);
    }
}
class Circle extends Shape {
    Circle() {
        side1 = getInput("Enter the
        radius of circle: ");
    void area() {
        double area = Math.PI *
            side1 * side2;
        System.out.println("The area
            of circle = "+area);
    }
}
class Main {
    public static void main (String args[])
    Rectangle rectangle = new Rectangle();
    Triangle triangle = new Triangle();
    Circle circle = new Circle();
    r.area();
    t.area();
    c.area();
    }
}
```

Output:

Enter length of the triangle:
3
Enter breadth of the triangle:
4
~~Area of rectangle =~~
~~12~~

enter
Base of triangle:
3
Enter height of triangle:
4
~~Area of triangle~~
6
Enter radius of circle:
4
Area of ~~circle~~ Rectangle:
12.0
Area of triangle:
6.0
Area of circle: 6
50.26548245

2/1/2024

Lab-05

```
Import Java.util.Scanner;
class Account {
    String customerName;
    int accno;
    String acctype;
    double balance;
    public Account (String name, int number,
    String type, double inHalBalance) {
    customerName = name;
    accno = number;
    acctype = type;
    balance = inHalBalance;
}
public void deposit (double amount) {
        balance += amount;
        System.out.println ("Deposit successful.
    Updated balance " + balance);
}
public void displayBalance() {
    System.out.println ("current Balance".
        + balance);
    }
}
class CurrAcct extends Account {
    double miBalance;
    double serviceCharge;
    public CurrAcct (String name, int
number, double inHalBalance) {
        super (name, number, "Current",
            inHalBalance);
```

```
        minBalance = 500.0;
        serviceCharge = 50.0;
    }
    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. Sevice charge of " + serviceCharge
            + " imposed.");
            balance -= serviceCharge;
            System.out.println("Updated
            balance after service charge: " +
                balance);
        }
    }
}
class SavAcct extends Account {
    double interestRate;
    public SavAcct(String name, int numberH
    double initialBalance, double rate) {
        super(name, number, "Savings",
            initialBalance);
                interestRate = rate;
    }
    public void depositInterest() {
        double interest = balance *
            (interestRate / 100);
        balance += interest;
        System.out.println("interest of " +
        interest + " deposited. Updated balance:
                " + 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.");
    }
}
}
Class BankMain {
    public static void main(String args[]) {
        Scanner s = new Scanner(System.in);
        CurrAcct currentAccount = new
            CurrAcct("Name1", 12345, 1000.0);
        SavAcct savingsAccount = new
            SavAcct("Name2", 789012, 2000.0, 5.0);
int choice;
    do {
        System.out.println("Menu");
        System.out.println("1. Deposit");
        System.out.println("2. Display Balance");
        System.out.println("3. Deposit Interest");
        System.out.println("4. Withdraw");
        System.out.println("5. Exit");
        System.out.println("Enter your choice:");
        choice = s.nextInt();
    switch(choice) {
        case 1: System.out.println("Enter
        deposit amount:");
```

```
            double depositAmount = s.nextDouble();
            currentAccount.deposit(
                depositAmount);
            break;
        Case 2:
            currentAccount.displayBalance();
            break;
        Case 3:
            if (savingsAccount instance of SavAcct)
{
            (ISavAcct) savingsAccount).
        depositInterest();
            }
            else {
                System.out.println("This option is
        applicable only for Savings Account"); }
            break;
        Case 4:
            System.out.println("Enter withdrawal
            amount :");
            double withdrawalAmount = s.nextDouble();
            currentAccount.withdraw(
                withdrawalAmount);
            break;
        Case 5:
            System.out.println("Exit");
            break;
        default:
            System.out.println("Invalid choice.
            Please enter again.");
        }
    }
    while (choice != 5);
}
}
```

Output:

Menu :
1. Deposit
2. Display Balance
3. Deposit Interest
4. Withdraw
5. Exit.

Enter your choice: 1
Enter deposit amount : 2000
Deposit successful. Updated balance 3000.0

Enter your choice : 2
Current balance 3000.0

Enter your choice : 3.
Interest of 100.0 deposited. Updated
Balance : 3100.0

Enter your choice : 4
Enter withdrawal amount : 200
Withdrawal successful. Updated balance :
2800.0

Enter your choice : 5
Exit.

16/11/24

LAB - 06

1. Program to demonstate string constructo

Output:

Hello
abc
hello
hell

2. Demonstate string length, string literal, contatenation.

Output:

Length of string : 13
String literal: Hello World
Name: Navya Billalar.

3. Demonstrate tostring().

Output:

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

4. Using getchars(), extract Bmsce from "Welcome to Bmsce college.

Output:

Extracted substring : Bmsce.

5. Demonstrate getbytes(), tochar Array() with proper java program

Output:

Byte array: 72 101 108 108 111

Char Array: Java

7.6. Output:

Substring is matched

Substring is not matched.

8. Demonstrate startwith()

Output:

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

9. Demonstrate endswith()

Output:

Ends with 'World!': true
Ends with 'Java': false.

10. Demonstrate equals() versus ==

Output:

Hello equals Hello -> true
Hello == Hello -> false.

11. Write a Java program to perform sorting four alphabets using compareTo().

Output:

~~Sorted names~~: sorted alphabets:
~~Anya~~ apple, ball, cat, dog, ent,
~~Bhavya~~ free, gun, hen, ice, jug, kite, lift,
~~Chetri~~ man, net, orange, parrot,
queen, ring, star, tree, umbrella,
van, watch, xmas, yatch, zee.

12. Java program to perform sorting of numbers from 10 to 1 using compareTo()

Output:

Sorted numbers: 1 10 2 3 4 5 6 7 8 9

10. Checked.

13. Java program to demonstrate concat() for s1 = "hello" & s2 = "world"

Output:

~~hello world~~
helloworld

14. Output:
modified string: This is a test. This is, too.

15. Output:

~~Replaced~~ modified string: Welcome to Bmsce Commiege

16. Output:

Trimmed string: [Hello Friends]

17. ~~Output:~~

Registration Number: 2456
Full name: Navya Bhalaj
Semester

6 Output:
Bmsce equals Bmse → true
Bmsee equals College → false
Bmsce equals BHSCE → false
Bmsce equalsIgnoreCase BHSCE → true

18. SetLength(), charAt(), setCharAt(), getChars(), append(), insert(), reverse(), delete(), deleteCharAt(), replace(), substring().

Output:

After setLength(5): Hello
Character at index 1: e
After setCharAt(7, 'H'): Hello, HoWld!
Extracted characters: Hello.
After append(', WoWld!'): Hello, WoWld!
After insert(5, ', WoWld!'): Hello, WoWld!
After reverse: !dlroW ,olleH
After delete(7, 12): Hello!
After deleteCharAt(5): Hello World!
After replace(7, 12, 'universe'): Hello, Universe!
Extracted substring(7, 12): Universe

Output:

20. Circle - Area: 78.5398 ,
perimeter : 31.415
Triangle - Area : 6.0 , perimeter : 12.0

19. Eagle: Eagle flies high in the sky
Eagle screeches loudly

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

17. Enter the details four student 1:
Registration Number : 10
Full name : Navya Billalar
Semester : 3
CGPA : 8.9
Enter details four student 2:
Registration number : 102
Full name : Nikita B
Semester : 3
CGPA : ~~8.8~~ 8.9
Enter details four student 3:
Registration number : 103
Full name : [redacted] Nikhilesh
Semester : 3
CGPA : 9.3

Lab - 06

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

```
Package CIE;
public class ~~extends~~ Internals extends Student {
    public int [] internalMarks,
    public Internals(String usn, String name,
        int sem) , int [] internalMarks ) {
        super (usn, name, sem)
        this.internalMarks = internalMarks;
    }
}
```

```
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 CIE.Internals;
import SEE.Externals;
import java.util.Scanner;
public class FinalMarks {
    public static void main(String args[]) {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the
        no.of students");
        int n = s.nextInt();
        Internals [] cieStudents = new Internals [n];
        Externals [] seeStudents = new External [n];
        for int (i=0; i<n; i++) {
        System.out.println("Enter the CIE
        details" + (i+1));
        System.out.println("USN:");
        String usn = s.next();
        System.out.println("Name:");
        String name = s.next();
        System.out.println("Sem:");
        int sem = s.nextInt();
        int [] cieMarks = new int [5];
        System.out.println("Enter the
        marks for
        for(j=0; j<5; j++) {
        cieMarks [j] = s.nextInt();
        }
        cieStudents [i] = new External
        (usn, name, sem, ciemarks);
        }
        for(int i=0; i<n; i++) {
        System.out.println("Enter SEE details"
        + (i+1));
        System.out.println("USN:");
```

```
String usn = s.next();
System.out.println("Name: ");
String name = s.next();
System.out.println("Sem: ");
int sem = s.nextInt();
int [] seeMarks = new int [5];
System.out.println("Enter SEE marks:");
    for(int j=0; j<5; j++){
        seeMarks [j] = s.nextInt();
    }
    SetStudent[i] = newExtend(usn, name,
        sem, seeMarks)
}
System.out.println("Find marks of students");
    for(i=0; i<n; i++){
        System.out.println("Enter details
            of student: " + (i+1));
        System.out.println("USN" + cieStudent
                (i).usn);
        System.out.println("Name" + cieStudent
                (i).name);
        System.out.println("Sem" + cieStudent
        System.out.println          (i).sem);
        System.out.println("CIE Marks");
        for(j=0; for(int j=0; j<5; j++)
        {
            System.out.println(cieStudent
                [i].internal
                Marks[j]
        }
    }
}
```

Output:

Enter the number of students:
Enter details for CIE of student 1
USN: 1BH22CS175
Name: Navya
Sem: 3
Enter CIE marks for
34
39
32
28
40
Enter details for CIE of student 2
USN: 1BH22CS002
Name: Name2
Sem: 3
Enter CIE marks for 5 subjects
20
32
38
31
28
Enter the details for SEE of student1
USN: 1BH22CS175
Name: Navya
Sem: 3
Enter SEE marks:
37
43
23
43

Enter details four see of student 2
USN: 1BM22CS002
Name: name2
Sem: 3
Enter see marks
49
48
47
46
44
Final marks of students
Details of student 1
USN: 1BM22CS175
name: Navya
Sem: 3
CIE marks:
34 39 32 28 40
SEE marks:
40 37 45 23 43
Details of student 2
USN: 1BM22CS002
name: Name2
Sem: 3
CIE marks:
20 32 38 31 28
See marks:
49 48 47 46 44

22/4/24

30-01-24

Lab - 08

1. Exception Handling

```
import java.util.Scanner;

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

Class InputScanner {
    public static void main (String args[]) {
    public static int getInput(String prompt)
    {
        Scanner s = new Scanner(System.in);
        System.out.println(prompt);
        return scanner.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 = 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 ("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 : 45
Enter Son's age : 12

Father's age : 45
Son's age : 12

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

30/01/24

Enter Father's age : 12
Enter Son's age : 45
Error: Son's age cannot be greater than Father's age.

Enter Father's age : 30
Enter Son's age : 30
Error: Son's Aage cannot be equal to Father's age.

Lab-09

1.
```
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 TwoThread {
    public static void main (String args[]) {
        MessageThread Thread1 = new MessageThr-
        -ead ("BMS College of Engineering ",
            10000);
        MessageThread Thread2 = new MessageThread
            ("CSE", 2000);

        Thread1.setName ("Thread1");
        Thread2.setName ("Thread2");

        Thread1.start();
        Thread2.start();
```

```
try {
    Thread.sleep(30000);
  }
  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
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
```

ESE
CSE
Main thread exiting
Thread 2 interrupted.
Thread 1 interrupted.

```
Class Q {
    int n;
    boolean valueSet = false;
    Synchronized int get() {
        while (!valueSet)
        try {
        System.out.println ("Consumer waiting");
        wait();
        } catch (InterruptedException e) {
            System.out.println ("InterruptedExcepti-
            -on caught");
            }
        System.out.println ("Got :" +n);
        valueSet = false;
        System.out.println ("Intimate Producer");
        notify();
        return n;
    }
    Synchronized void put (int n) {
        while (valueSet)
        try {
        System.out.println ("Producer waiting");
        wait();
    } catch (InterruptedException e) {
        System.out.println ("InterruptedException
            caught");
    }
    this.n = n;
```

```
        valueset = true;
        System.out.println("Put: " + n);
        System.out.println("Intimate Consumer");
        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 Main {
    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
Got: 0
Intimate producer
put: 1
Intimate consumer
producer waiting
consumed: 0
Got: 1
Intimate Producer
consumed: 1
Put: 2
Intimate Consumer
```

R/Platform

-02-24

Lab-10

1.
```
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().setNam
            ("MainThread");
            Thread t=new Thread(this,
            "RacingThread");
            t.start();
            a.demo(b);
            System.out.println("Back in main
                thread");
        }
        public void run(){
                b.bar(a);
                System.out.println("Back in
                    other thread");
        }
        public static void main(String args[])
        {
            new Deadlock();
        }
}
```

Output:-

RacingThread entered b bar
MainThread entered A.demo
RacingThread trying to call A.last()
MainThread trying to call B.blast()

MainThread entered A.demo
RacingThread entered B.bar
Mainthread trying to call B.last()
Racing thread trying to call A.last()
inside B.last()
back in main thread
inside A.last()
Back in other thread

13/2/2024

20-2-24

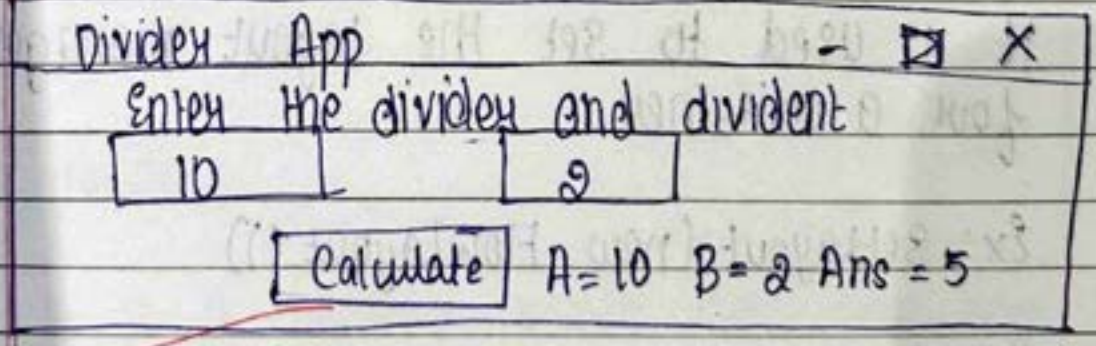
Lab-09

1)
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import
Class SwingDemo {
    SwingDemo() {
        JFrame jfrm = new JFrame("
            Divider App");
        jfrm.setSize(275, 500);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.
            EXIT_ON_CLOSE);
        JLabel jlab = new JLabel("Enter the
        divider and divident:");
        JTextField ajtf = new JTextField(8);
        JTextField bjtf = 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(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("\B = "+b);
        anslab.setText("\nAns = "+ans);
    }

    catch(NumberFormatException e) {
            alab.setText("");
            blab.setText("");
            anslab.setText("");
            err.setText("Enter only integers");
    }
```

```
        catch (ArithmeticException e) {
                alab.setText("");
                blab.setText("");
                ansLab.setText("");
                err.setText("B should not be
                zero");
            }
        }
    });
    jfrm.setVisible(true);
}

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

Output :-



Run
20/12/24

Report on all the functions used :

1. JFrame :

* It is part of the Java Swing library, which provides a set of components for building desktop applications.

Ex: JFrame frame = new JFrame ("My Swing Application");

2. SetSize :

* It is a method in Java Swing which is used to set the dimensions (width and height) of a component.

Ex: setSize (400, 300).

3. SetLayout

* It is used to set the layout manager for a container

Ex: setLayout (new FlowLayout ())

4. SetdefaultCloseOperation

* It is used to set the default operation that should happen when the user closes a window.

5. JLabel:

* It is used to display a single line of non-editable text or an image.

6. JTextField

* It is a component that allows the user to enter and edit a single line of text.

7. addFrame

* It is a method in the 'container' class (which ~~extends~~ 'JFrame' extends) that adds a component to the container.

8. addActionListener:

* It is an ~~method~~ interface that responds to events, such as button clicks.

* JButton extends it. It adds an 'ActionListener' to a button.

9. SetText:

* It is a method which sets the text displayed by the label.

20/12/24