

B.M.S COLLEGE OF ENGINEERING BENGALURU
Autonomous Institute, Affiliated to VTU



Submitted in partial fulfillment of the requirements for record of

OBJECT ORIENTED JAVA PROGRAMMING

(23CS3PCOOJ)

Submitted by:

SOHAN T SANJEEV

2023BMS02532

Faculty incharge:

Dr Seema Patil

Department of Computer Science and Engineering
B.M.S College of Engineering
Bull Temple Road, Basavanagudi, Bangalore 560 019

**B.M.S COLLEGE OF ENGINEERING DEPARTMENT OF COMPUTER
SCIENCE AND ENGINEERING**

Lab 1:

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

```
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 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:");
            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("Root1 = Root2 = " + r1);
        } else if (d > 0) {
            r1 = ((-b) + (Math.sqrt(d))) / (2 * a);
            r2 = ((-b) - (Math.sqrt(d))) / (2 * a);
            System.out.println("Roots are real and distinct");
            System.out.println("Root1 = " + 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("Root1 = " + r1);
        System.out.println("Root2 = " + r2 + "i");
    }
}

class QuadraticMain {
    public static void main(String[] args) {
        System.out.println("USN = 2023BMS02532 NAME = SOHAN T SANJEEV");
        Quadratic q = new Quadratic();
        q.getd();
        q.compute();
    }
}
```

Lab 2:

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

```
import java.util.*;  
  
class Subject{  
    int subjectMarks;  
    int credits;  
    int grade;  
}  
  
class Student{  
    Subject subject[];  
    String name;  
    String usn;  
    double SGPA;  
    Scanner sc=new Scanner(System.in);  
  
    Student(){  
        int i;  
        subject = new Subject[8];  
        for(i=0;i<8;i++)  
            subject[i] = new Subject();  
    }  
  
    public void getStudentDetails(){  
        System.out.println("enter your name");  
        name=sc.nextLine();  
        System.out.println("usn");  
        usn=sc.nextLine();  
    }  
}
```

```
public void getMarks(){
    for(int i=0;i<8;i++){
        System.out.println("enter the marks for "+(i+1)+" subject");
        subject[i].subjectMarks=sc.nextInt();
        System.out.println("enter the credits of "+(i+1)+" subject:");
        subject[i].credits=sc.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;
        }
    }
}
```

```
public void computeSGPA(){
    int effectiveScore=0;
    int total=0;
    float SGPA=0;
    for(int i=0;i<8;i++){
        effectiveScore+=(subject[i].credits*subject[i].grade);
        total+=subject[i].credits;
    }
    System.out.println("effective score:"+effectiveScore);
    System.out.println("total credits:"+total);
    SGPA=effectiveScore/total;
    System.out.println("SGPA;"+SGPA);
}
}
```

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

```
Student s=new Student();
s.getStudentDetails();
s.getMarks();
s.computeSGPA();
System.out.println("name:"+s.name);
System.out.println("usn:"+s.usn);
}
}
```

Lab3:

Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a `toString()` method that could display the complete details of the book.

Develop a Java program to create n book objects.

```
import java.util.Scanner;

class Books{

String name,author;
int price,numPages;

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

String tostring(){
    String author,name,price,numPages;
    name="This Book is "+this.name+"\n";
    author="Author is "+this.author+"\n";
    price="price ="+this.price+"\n";
    numPages="No of pages "+this.numPages+"\n";
    return name+author+price+numPages;
}

}

public class Main{
    public static void main(String[] args) {
        System.out.println("USN = 2023BMS02532 NAME = SOHAN T SANJEEEV");
        Scanner s=new Scanner(System.in);
        int n,numPages,price;
        String name,author;
        System.out.println("Enter No of Books ");
        n=s.nextInt();
        Books arr[];
        arr=new Books[n];
```

```
for(int i=0;i<n;i++){  
    System.out.println(" Enter the name of Book ");  
    name=s.next();  
    System.out.println(" Enter the author of Book ");  
    author=s.next();  
    System.out.println("Enter the price of Book ");  
    price=s.nextInt();  
    System.out.println("Enter the no of pages of the Book ");  
    numPages=s.nextInt();  
    arr[i]=new Books(name,author,price,numPages);  
}  
  
for(int i=0;i<n;i++){  
    System.out.println(arr[i].tostring());  
}  
}
```

Lab 4:

Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea() that prints the area of the given shape.

```
import java.util.Scanner;

class InputScanner{
    Scanner s;
    InputScanner(){
        s=new Scanner(System.in);
    }
}

abstract class Shape extends InputScanner{
    double a;
    double b;
    abstract void getDetails();
    abstract void printArea();
}

class Rectangle extends Shape{
    void getDetails(){
        System.out.println("Enter length and breadth");
        a=s.nextDouble();
        b=s.nextDouble();
    }
    void printArea(){
        System.out.println("Area of Rectangle = "+(a*b));
    }
}

class Triangle extends Shape{
    void getDetails(){

```

```
System.out.println("Enter a and b");
a=s.nextDouble();
b=s.nextDouble();
}

void printArea(){
    System.out.println("Area of Triangle = "+(a*b/2));
}

}

class Circle extends Shape{
void getDetails(){
    System.out.println("Enter radius");
    a=s.nextDouble();
}
void printArea(){
    System.out.println("Area of Circle = "+(3.14*a*a));
}
}

class AbstractMain{
public static void main(String[] args) {
    Rectangle r = new Rectangle();
    Triangle t = new Triangle();
    Circle c = new Circle();
    r.getDetails();
    r.printArea();
    t.getDetails();
    t.printArea();
    c.getDetails();
    c.printArea();
    System.out.println("NAME:SOHAN T SANJEEV, USN = 2023BMS02532"); }}}
```

Lab 5:

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

```
import java.util.*;
import java.util.*;

class Account{
String name;
int accno;
String type;
double balance;
Account(String name,int accno,String type,double balance){
this.name = name;
this.accno = accno;
this.type = type;
this.balance = balance;
}

void deposit(double amount){
balance +=amount;
}
void withdraw(double amount){
if((balance-amount)>=0){
balance -=amount;
}
else{
System.out.println("Insufficient balance");
}
}
void display(){
System.out.println("Name : "+name+"\n"+
"AccountNo : "+accno+"\n"+
"Type : "+type+"\n"+
"balance: "+balance+"\n");
}

class SavingAccount extends Account{
private static int rate = 5;
SavingAccount(String name,int accno,String type,double balance){
super(name,accno,type,balance);
}
void balanceWithInterest(){
balance +=balance*rate/100;
System.out.println("balance: "+balance);
}
}
```

```

class CurrentAccount extends Account{
    private static int rate = 5;
    CurrentAccount(String name,int accno,String type,double balance){
        super(name,accno,type,balance);
    }
    void balanceWithInterest(){
        balance +=balance*rate/100;
        System.out.println("balance: "+balance);
    }
}

public class Main{
    public static void main(String args[]){
        Scanner s = new Scanner(System.in);
        System.out.println("Enter your name: ");
        String name = s.nextLine();

        System.out.println("Enter the account type (current or savings)");
        String type = s.next();

        System.out.println("Enter the account number: ");
        int accno = s.nextInt();

        System.out.println("Enter the initial balance: ");
        double balance = s.nextDouble();

        Account acc = new Account(name,accno,type,balance);
        SavingAccount sa = new SavingAccount(name,accno,type,balance);
        CurrentAccount cc= new CurrentAccount(name,accno,type,balance);
        double amount;
        while(true){
            if(acc.type.equals("savings")){
                System.out.println("\n-----MENU-----\n");
                System.out.println("1. Deposit \n2.Withdraw \n3.compute interest for SavingsAccount \n4.Display Account Details\n 5.Exit\n");
                System.out.println("Enter your choice");
                int choice = s.nextInt();

                switch(choice){
                    case 1:System.out.println("Enter the deposit amount");
                    amount = s.nextDouble();
                    sa.deposit(amount);
                    break;
                    case 2: System.out.println("Enter the withdrawl amount ");
                    amount = s.nextDouble();
                    sa.withdraw(amount);
                    break;
                    case 3:sa.balanceWithInterest();
                    break;
                    case 4:System.out.println("Details: ");
                    sa.display();
                    break;
                }
            }
        }
    }
}

```

```
case 5: return;
default: System.out.println("Invalid choice ");
}

}

else if(acc.type.equals("current")){
    System.out.println("\n-----MENU-----\n");
    System.out.println("1. Deposit \n2.Withdraw \n 3.compute interest for SavingsAccount \n
4.Display Account Details\n 5.Exit\n");

    System.out.println("Enter your choice");
    int choice = s.nextInt();

    switch(choice){
        case 1:System.out.println("Enter the deposit amount");
        amount = s.nextDouble();
        cc.deposit(amount);
        break;
        case 2: System.out.println("Enter the withdrawl amount ");
        amount = s.nextDouble();
        cc.withdraw(amount);
        break;
        case 3:cc.balanceWithInterest();
        break;
        case 4:System.out.println("Details: ");
        cc.display();
        break;
        case 5: return;
        default: System.out.println("Invalid choice ");
    }

    System.out.println("NAME:SOHAN T SANJEEV,USN: 2023BMS02532");
}
}
}
}
```

Lab 6:

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

Cie/Student.java:

```
package cie;  
import java.util.Scanner;  
public class Student{  
    public int sem;  
    public String usn,name;  
    Scanner sc=new Scanner(System.in);  
    public void setStudentDetails(){  
        System.out.println("Enter your Name :");  
        name=sc.nextLine();  
        System.out.println("Enter your USN :");  
        usn=sc.nextLine();  
        System.out.println("Enter your Sem :");  
        sem=sc.nextInt();  
    }  
}
```

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

}

Cie/Internals.java:

```
package cie;  
import java.util.Scanner;  
  
public class Internals extends Student{
```

```

public int internalsMarks[] = new int[5];
Scanner sc = new Scanner(System.in);
public void setCIE(){
for(int i=0;i<5;i++){
    System.out.println("Enter cie marks of Subject "+(i+1));
    internalsMarks[i] = sc.nextInt();
}
}
}

```

See/External.java:

```

package see;
import cie.Internals;
import java.util.Scanner;
public class External extends Internals{
    public int seeMarks[] = new int[5];
    public int finalMarks[] = new int[5];
    Scanner sc = new Scanner(System.in);
    public void setSEE(){
        for(int i=0;i<5;i++){
            System.out.println("Enter see marks of Subject "+(i+1));
            seeMarks[i] = sc.nextInt();
        }
    }
    public void computeFinal(){
        for(int i=0;i<5;i++){
            finalMarks[i] = internalsMarks[i] + seeMarks[i]/2;
        }
    }
    public void displayMarks(){
        for(int i=0;i<5;i++){

```

```
        System.out.println("Subject "+(i+1)+" = "+finalMarks[i]);  
    }  
  
}  
}
```

DemoMain.java:

```
import see.External;  
public class DemoMain{  
    public static void main(String[] args) {  
        External obj=new External();  
        obj.setStudentDetails();  
        obj.getDetails();  
        obj.setCIE();  
        obj.setSEE();  
        obj.computeFinal();  
        obj.displayMarks();  
    }  
}
```

Lab 7:

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge() when the input age=father’s age.

```
import java.util.Scanner;

class WrongAge extends Exception{
    public WrongAge(String A){
        super(A);
    }
}

class Father{
    int fatherAge;
    Scanner sc=new Scanner(System.in);
    public void validAge() throws WrongAge{
        System.out.println("Enter Fathers age");
        fatherAge=sc.nextInt();
        if(fatherAge<=0){
            throw new WrongAge("Invalid fathers age");
        }
    }
}

class Son extends Father{
    int sonAge;
    Scanner sc=new Scanner(System.in);
    public void validAge() throws WrongAge{
        System.out.println("Enter sons age");
        sonAge=sc.nextInt();
        super.validAge();
        if (sonAge>=fatherAge) {
            throw new WrongAge("Sons age cant be greater than Fathers age");
        }
    }
}
```

```
else if(sonAge<0){  
    throw new WrongAge("Invalid son age");  
}  
}  
  
}  
  
public class MyMain{  
    public static void main(String[] args) {  
        Son obj = new Son();  
        try{  
            obj.validAge();  
        }  
        catch(WrongAge e){  
            System.out.println("Exception "+e.getMessage());  
        }  
    }  
}
```

Lab 8:

Write a program which creates two threads, one thread displaying “BMS College of Engineering” once every ten seconds and another displaying “CSE” once every two seconds.

```
class Bmsce extends Thread{  
    public void run(){  
        for(int i=0;i<10;i++){  
            System.out.println("Bmsce");  
            try{  
                this.sleep(10000);  
            }  
            catch(InterruptedException e){System.out.println(e);}  
            System.out.println(i);  
        }  
    }  
  
    class Cse extends Thread{  
        public void run(){  
            for(int i=0;i<10;i++){  
                System.out.println("CSE");  
                try{  
                    this.sleep(2000);  
                }  
                catch(InterruptedException e)  
                {System.out.println(e);}  
            }  
        }  
  
        public class MyMain{  
            public static void main(String[] args) {  
                Bmsce obj1 = new Bmsce();  
                Cse obj2 = new Cse();  
                obj1.start();  
                obj2.start();}}}
```

Lab 9:

Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an ArithmeticException. Display the exception in a message dialog box.

```
import 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 dividend:");

        // 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
jfrm.add(jlab);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);

ActionListener listener = new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        System.out.println("Action event from a text field");
    }
};

ajtf.addActionListener(listener);
bjtf.addActionListener(listener);

button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        try {
            int a = Integer.parseInt(ajtf.getText());
            int b = Integer.parseInt(bjtf.getText());
            if (b == 0) {
                throw new ArithmeticException();
            }
            int ans = a / b;
            alab.setText("\nA = " + a);
            blab.setText("\nB = " + b);
            anslab.setText("\nAns = " + ans);
            err.setText("");
        } catch (NumberFormatException e) {
    }
}

```

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

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

public static void main(String args[]) {
    // create frame on event dispatching thread
    SwingUtilities.invokeLater(new Runnable() {
        System.out.println("SOHAN T SANJEEV, 2023BMS02532");
        public void run() {
            new SwingDemo();
        }
    });
}
}
```

Lab 10:**Demonstrate Inter process Communication and deadlock.**

a) IPC

```
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("InterruptedException 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 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.");
    }
}

b) DeadLock

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");  
}  
}  
public 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);  
        System.out.println("Back in mainthread");  
    }  
    public void run() {  
        b.bar(a);  
        System.out.println("Back in other thread");  
    }  
    public static void main(String args[]) {  
        new Deadlock();  
    }  
}
```

INDEX

Name : Sohan T. Sangeen Subject : Java Observation.

Std. : 11th Sem. Div. : E Roll No. : 2023 BMS02532

School / College : BMS CE

Sl No.	Date	Title	Page No.	Teacher Sign/ Remarks
①	12/12/23	Lab - ①	1 - 3	
②	19/12/23	Lab - ② Student database	4 - 9	
③	26/12/23	Lab - ③	10 - 13	
④	2/1/24	Lab - ④ Abstraction	14 - 18	
⑤	9/1/24	Lab - ⑤ Bank	19 - 23	
⑥	16/1/24	Strings & Generics	24 - 30	
⑦	23/1/24	Lab - 6 package	31 - 35	
⑧	30/1/24	Lab - 7 Exceptions	36 - 38	
⑨	06/2/24	Lab - 8 Multi threading	39 - 41	
⑩	13/2/24	Lab - 10 SPC & deadlock	42 - 48	
⑪	20/2/24	Lab - 9 Applet	49 - 53	

LPA 12-12-23

```
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 values 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");  
            Scanner s = new Scanner(System.in);  
            a = s.nextInt();  
        }  
        d = b * b - 4 * a * c;  
    }  
}
```

if ($d == 0$)

$$r_1 = (-b) / (2 * a);$$

System.out.println ("Roots are real & Equal");

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

}

else if ($d > 0$)

{

$$r_1 = ((-b) + (\text{Math.sqrt}(d))) / (\text{double}(2 * a));$$

$$r_2 = ((-b) - (\text{Math.sqrt}(d))) / (\text{double}(2 * a));$$

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

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

}

else if ($d < 0$)

{

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

$$r_1 = (-b) / (2 * a);$$

$$r_2 = \text{Math.sqrt}(-d) / (2 * a);$$

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

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

}

}

}

②

class QuadraticMain {
 public static void main(String args[]) {
 Quadratic q = new Quadratic();
 q.getd();
 q.compute();
 }
}

System.out.println("USN: 2023BMS02532, Name:
" + "Sohan T Sanjeev");

q
q

Output:-

c:\java programs>javac QuadraticMain.java

c:\java programs>java QuadraticMain

Enter the coefficients of a, b, c

8

100

11

Roots are imaginary

Root1 = 0.0 + i0.9921567416492215

Root2 = 0.0 - i0.9921567416492215

USN: 2023BMS02532, Name: Sohan T Sanjeev (3)

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

formula for SGPA

$$\text{SGPA} = \frac{\sum [(\text{course credits}) (\text{grade points})]}{\sum [\text{course credits}]}$$

$$\text{GPA} = \frac{\sum [(\text{course credits}) (\text{grade points})]}{\sum [\text{course credits}]}$$

```
import java.util.*;
```

```
class Subject {
```

```
    int subjectMarks;
```

```
    int credits;
```

```
    int grade;
```

3

```
class Student {
```

```
    Student student[];
```

```
    String name;
```

```
    String usn;
```

```
    double SGPA;
```

```
    Scanner sc = new Scanner(System.in);
```

```
Student() {
```

```
    int i;
```

```
    subject = new Subject[8];
```

```
    for (i=0; i<8; i++) {
```

```
        subject[i] = new Subject();
```

4

```
public void getStudentDetails() {
```

```
    System.out.println("Enter your name");
```

```
    name = sc.nextLine();
```

```
    System.out.println("USN");
```

```
    USN = sc.nextLine();
```

(5)

5

```
public void getMarks() {
```

```
    for (int i=0; i<8; i++) {
```

```
        System.out.println("Enter the marks  
        (i+1);
```

```
        subject[i].subjectMarks = sc.nextInt();
```

```
        System.out.println("Enter the credits  
        + (i+1) "Subject");
```

```
        subject[i].marks credits = sc.nextInt();
```

```
        subject[i].grade = (subject[i].marks / 10) + 1;
```

```
        if (subject[i].grade == 10) {
```

```
            subject[i].grade = 10
```

y

```
        if (subject[i].grade <= 4) {
```

```
            subject[i].grade = 4
```

y

y

```
public void ComputeSGPA() {
```

```
    int effectiveScore = 0;
```

```
    int total = 0;
```

```
    float SGPA = 0;
```

```
for (int i=0; i<8; i++) {
```

```
    effectiveScore += (subject[i].credit * subject[i].  
                        grade);
```

```
total += subject[i].credit;
```

8g

```
System.out.println("effective score : " + effectiveScore);
```

```
System.out.println("total : " + total);
```

```
SGPA = effectiveScore / total;
```

```
System.out.println("SGPA : " + SGPA);
```

g

g

```
class javaMain {
```

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

```
    Student s = new Student();
```

```
    s.getStudentDetails();
```

```
    s.getMarks();
```

```
    s.computeSGPA();
```

```
    System.out.println("name : " + s.name);
```

```
    System.out.println("usn : " + s.usn);
```

```
    System.
```

g F

(7)

Output :-

enter your name

SOHAN

USN

2023BMS02532

enter the marks for 1 subject

80

enter the credits of 1 subject

4

enter the marks for 2 subject

70

enter the credits of 2 subject

4

enter the marks for 3 subject

67

enter the credits of 3 subject

3

enter the marks for 4 subject

58

enter the credits of 4 subject

3

enter the marks for 5 subject

69

enter the credits of 5 subject

4

enter the marks for 6 subject

87

enter the credits of 6 subject

2

enter the marks for 7 subject

159

enter the credits of 7 subject

1

enter the marks for 8 subject

50

enter the credits of 8 subject

1

effective score : 165

total credit : 22

SGPA; 7.0

name: SOLTAN

USN: 2023BMS02532

Create a class book which contains four members name, author, price, num-pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a `toString()` method that could display the complete details of the book. Develop a Java program to create n book objects.

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

```
class Book {
```

```
    String name;
```

```
    String author;
```

```
    int price;
```

```
    int num-pages;
```

```
    Book (String name, String author, int price,  
          int num-pages) {
```

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

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

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

```
        this.num-pages = num-pages;
```

```
public String toString() {  
    String name, author, price, numPages;  
    name = "name :" + this.name + "\n";  
    author = "author :" + this.author + "\n";  
    price = "price :" + this.price + "\n";  
    numPages = "no of pages :" + this.numPages + "\n";  
    return name + author + price + numPages;  
}
```

```
Public class Main {
```

```
public static void main (String args[]) {  
    Scanner s1 = new Scanner (System.in);  
    String name, author;  
    int price, numPages;  
    int n;  
    System.out.println ("Enter the number of books: \n");  
    n = s1.nextInt();  
    Book b[];  
    b = new Book[n];  
    for (int i=0; i<n; i++) {  
        System.out.println ("Enter the name of  
                           book : \n");  
    }  
}
```

```
name = s1.nextLine();
System.out.println("Enter the name of author");
author = s1.nextLine();
System.out.println("Enter the price of book:");
price = s1.nextInt();
System.out.println("Enter the no. of pages for book's \n");
numPages = s1.nextInt();
b[i] = new Book(name, author, price, numPages);
for (int i = 0; i < n; i++) {
    System.out.println(b[i].toString());
}
System.out.println("name : " + Sohan T Sanjeet
System.out.println("USN : " + 2023BMS02522 + "\n");
output:-
```

Enter the number of books:

2

Enter the name of book:

Java

Enter the author of book:

herbert

Enter the price of book :

234

enter the no of pages for book :

678

enter the name of author :

~~Author~~ jungle-book

author

enter the ~~price~~ of book :

Stalin

enter the price of book :

~~245~~ 245

enter the no of pages for book :

700

name: Java

author: herbert

price: 234

no of pages : 678

name: jungle-book

author & name : Stalin

price : 245

no of pages : 700

name: Sohan T Sanjeeve

USN: 2023 BM502572

LAB PROGRAM 4

LPT 2-1-2h.

2/1/21

Develop a Java program to create an abstract class shape that contains two integers, an empty method named printArea().
three classes named Rectangle, Triangle, Circle such that each one of the classes extends the class shape. Each one of the classes contain only the method printArea() that prints the area of the given shape.

```
import java.util.*;
```

```
class InputScanner {
```

```
    Scanner s1 = new Scanner(System.in);
```

3

```
abstract class shape extends InputScanner {
```

```
    double a;
```

```
    double b;
```

```
    abstract void getInput();
```

```
    abstract void displayArea();
```

9

(tu)

```
class Rectangle extends Shape {  
    void get Input () {  
        System.out.println ("enter value of a");  
        a = sc.nextInt ();  
        System.out.println ("enter value of b");  
        b = sc.nextInt ();  
    }  
}
```

```
void displayArea () {  
    System.out.println ("area of rectangle  
is " + (a * b));  
}
```

```
class Triangle extends Shape {
```

```
void get Input () {
```

```
System.out.println ("Enter the value of a");
```

```
a = sc.nextInt ();
```

```
System.out.println ("Enter the value of b");
```

```
b = sc.nextInt ();
```

Void displayArea () {

System.out.println ("area of Triangle is"
+ (a * b / 2)))

;

}

class Circle extends Shape {

void getInput () {

System.out.println ("Enter value of b :");

b = sc.nextDouble();

;

void displayArea () {

System.out.println ("area of circle is"
+ (b * b * 3.14));

;

class AbstractDemo {

public static void main (String args []) {

Rectangle r = new Rectangle ();

Circle c = new Circle ();

Triangle t = new Triangle ();

r. getInput();

r. displayArea();

c. getInput();

f. displayArea();

t. getInput();

t. displayArea();

System.out.println("name:
Sohan T Sanjeev");

8

↳ System.out.println("USN: 2023BMS02T32")

9

Output:-

enter the value of a:

1

enter the value of b:

4

area of rectangle is 4.0

enter the value of b:

3

area of circle is 28.26

enter value of a

5

enter value of b

area of triangle is 17.5

(17)

Name: Sohan T Sanjeev

USN: 2023BMS02632

QUESTION 1: Explain the following terms:

(a) Electrolytic Conductivity

(b) Electrolytic Conductance

(c) Electrolytic Conductance Cell

(d) Electrolytic Conductance Cell Constant

(e) Electrolytic Conductance Cell Constant

(f) Electrolytic Conductance Cell Constant

(g) Electrolytic Conductance Cell Constant

(h) Electrolytic Conductance Cell Constant

(i) Electrolytic Conductance Cell Constant

(j) Electrolytic Conductance Cell Constant

(k) Electrolytic Conductance Cell Constant

(l) Electrolytic Conductance Cell Constant

(m) Electrolytic Conductance Cell Constant

(n) Electrolytic Conductance Cell Constant

(o) Electrolytic Conductance Cell Constant

(p) Electrolytic Conductance Cell Constant

Develop a Java program to create a class Bank that contains two kinds of account for its customers, one called savings account and other current account provides cheque book facility. The current account provides interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

- (*) Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-Acc and Sav-Acc to make them more specific to their requirements. include necessary methods in order to achieve the following tasks
- (*) accept deposit from customer and update the balance
- (*) display the balance
- (*) compute and deposit interest
- (*) permit withdrawal and update balance
- (*) check for minimum balance, impose penalty if necessary and update balance

```
import java.util.*;  
class Account {  
    String name;  
    int accno;  
    String type;  
    double balance;  
    Account (String name, int accno, String type, double  
        balance) {  
    }  
}
```

this.name = name;

this.accno = accno;

this.type = type;

this.balance = balance;

}

value of class of abstract parent class

void deposit (int amount)

balance += amount;

if ((balance - amount) >= 0) {

balance -= amount;

else {

System.out.println ("insufficient balance");

```
void display () {  
    System.out.println ("Name : " + name + "\n")  
    + "Account No. : " + acno + "\n" + "Type : "  
    + type + "\n" + "balance : " + balance + "\n");  
  
class SavingAccount extends Account {  
    private static int rate = 5;  
  
    SavingAccount (String name, int acno,  
    String type, double balance) {  
        super (name, acno, type, balance);  
  
        void balanceWithInterest () {  
            balance += balance * rate / 100;  
            System.out.println ("balance : " + balance);  
  
public class Main {  
    public static void main (String args []) {  
        Scanner s = new Scanner (System.in);  
        System.out.print ("Enter the account type  
        (current or deposit)");  
        String type = s.next();  
    }  
}
```

```
System.out.println("Enter account number");
int accno = s.nextInt();
System.out.println("Enter initial balance");
double balance = s.nextDouble();
Account acc = new Account(name, accno, type, balance);
Saving Account sa = new SavingAccount(name, accno, type, balance);
double amount;
while (true) {
    if (acc.type.equals("savings")) {
        System.out.println("\n --- MENU --- \n");
        System.out.println("1. Deposit \n 2. Withdrawal \n 3. Compute Interest for Savings Account \n 4. Display Account details \n 5. Exit \n");
        System.out.println("Enter your choice");
        int choice = s.nextInt();
        switch (choice) {
            case 1: System.out.println("Enter deposit amount");

```

amount = s. nextDouble();

sa. deposit(amount);
break;

case 2: System.out.println ("Enter withdraw amount");

amount = s. nextDouble();
sa. ~~next~~ withdraw(amount);
break;

case 3: sa. balanceWithInterest();

break;

case 4: System.out.println ("Details")

sa. display();
break;

case 5: return;

default : System.out.println ("Invalid choice");

System.out.println (Sohan T Sanjeev USN: 2023
Date: 11/12/2023
BM1502532))

Op:-

Enter your Account number

02532

Menu

- | | | | | |
|-----------|------------|-----------|---------------------|--------|
| 1 Deposit | 2 Withdraw | 3 Display | 4 Compound Interest | 5 Exit |
|-----------|------------|-----------|---------------------|--------|

Sohan T Sanjeev, USN:

(23)

Lab-6

1) Demonstrate String constructors

2) Demonstrate string length, string literal, concat

```
class SubstringCons {
```

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

```
        String s1 = new String ("");
```

```
        String s2 = new String ("hello");
```

```
        System.out.println ("s1 = " + s2);
```

```
        char chars[] = {'a', 'b', 'c', 'd', 'e'};
```

```
        String s3 = new String (chars, 0, 3);
```

```
        System.out.println ("s3 : " + s3);
```

```
        byte ascii [] = {65, 66, 67, 68, 69, 70};
```

```
        String s4 = new String (ascii);
```

```
        System.out.println (s4);
```

```
        String s5 = new String (ascii, 2, 3);
```

```
        System.out.println (s5);
```

```
        String s6 = "hello";
```

```
        System.out.println ("length of string s :  
            " + s6.length());
```

```
        String s7 = "java";
```

```
String obj = new String (s6+s7);
System.out.println("obj = " + obj);
System.out.println("Sohan T Sangeev", 2023BMS02532);
```

9. Output:-

Output:-

s1 = hello

s3 = abc

ABCD E F

C D E

length of string s6 : 5

obj = hello java

Sohan T Sangeev, 2023BMS02532

~~program 2 :-~~

WAP to create an abstract class Bird with methods fly() & makeSound(). Create subclass Eagle and Hawk that Extend class & Implement the respective methods to describe how each bird flies & makes a sound.

abstract class Bird {

 abstract void fly();

 abstract void makeSound();

}

class Eagle Extends Bird {

 void fly() {

 System.out.println("eagle soars high
 in the sky");

}

 void makeSound() {

 System.out.println("eagle makes screaming
 sound");

}

 Eagle e = new Eagle();

 Bird b = e;

class Hawk extends Bird {

 void fly() {

 System.out.println("hawk glides
 gracefully in air");

}

 void makeSound() {

system.out.println("hawk makes distinct cry")

class AbstractBird {
 public static void main (String args[]){
 eagle e = new Eagle();
 Hawk h = new Hawk();
 e.fly();
 e.makeSound();
 h.fly();
 h.makeSound();
 }
 System.out.println("name : Sohan T Sanjeev, USN : 2023BMS02532");
 ~~16/11/24.~~

Output:-

eagle soars high in the sky
eagle makes screeching sound
hawk glides gracefully in air
hawk makes distinct cry
name : Sohan T Sanjeev, USN : 2023BMS02532

③ WAP to create a generic class ~~Stack~~
which hold 5 integers and 5 double

```
public class GenericStack<T> {  
    private Object[] stackArray;  
    private int top;  
    private static final int MAX_SIZE = 10;
```

```
public GenericStack() {  
    stackArray = new Object[MAX_SIZE];  
    top = -1;
```

```
public void push (T element) {
```

```
    if (top < MAX_SIZE - 1) {  
        stackArray[++top] = element;
```

```
    else {  
        System.out.println ("Stack is full can't  
        push more elements");
```

```
public T pop () {  
    if (!is Empty ()) {  
        @ Suppress Warnings ("unchecked").  
        T element = (T) StackArray [top--];  
        System.out.println (" popped : " + element);  
        return element;  
    }  
}
```

else {

```
    System.out.println (" stack is empty : "  
        cannot pop elements ");  
    return null;  
}
```

y

z

```
public boolean isEmpty () {  
    return top == -1;  
}
```

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

```
    GenericStack < Integer > integerStack = new  
    GenericStack < > ();
```

```
    integerStack.push (1);
```

```
    integerStack.push (2);
```

integer stack (3);

integer stack . pop ();

generic stack <double> doubleStack = new

stack <> (7)

doubleStack . push (1.5);

doubleStack . push (2.5);

doubleStack . push (3.5);

doubleStack . pop ();

System.out.println ("Sohan T Sanjeev, USN: 2023BM502532
32 NOJ 18/11/20")

Q

O/P:

pushed 1

pushed 2

pushed 3

popped 3

pushed 1.5

pushed 2.5

pushed 3.5

popped 3.5

Lab-6

Create a package CIE which has two classes - students and Internals. The class Student has members like USN, name, sem. The class Internals derived from student has an array that stores the internal marks scored in five courses of the current semester of the student.

Create another package SEE which has the class Internals which is a derived class of student. This class has an array that stores the SEG marks scored in five ~~marks~~ courses of student. Import the java package in a file that declare final marks of nth student in all five courses.

~~CIE\Student.java~~

```
package cie; import java.util.Scanner;  
public class Student { Scanner sc = new  
Scanner(System.in);  
public String name, USN;  
public int sem;  
public void setDetails() {  
name = sc.nextLine();
```

```
public void getDetails () {  
    System.out.println ("name:" + name);  
    System.out.println ("USN:" + USN);  
    System.out.println ("Sem:" + Sem);  
}
```

CIE/ Internals.java

```
package cie;  
import java.util.Scanner;  
public class Internals extends student {  
    public int Intermarks [] = new int [5];  
    Scanner sc = new Scanner (System.in);  
    public void setcie () {  
        for (int i=0; i<5; i++) {  
            System.out.print ("Enter marks  
                for " + (i+1)) + " : ";  
            Internalmarks [i] = sc.nextInt();  
    }  
}
```

3

(32)

see / External

```
package see;
import java.util.Scanner;
import cie.internals;

class External extends Internals {
    public int SeeMarks [] = new int [5];
    public int finalMarks [] = new int [5];
    Scanner sc = new Scanner (System.in);

    public void setSee () {
        for (int i = 0; i < 5; i++) {
            System.out.print ("Enter marks
of " + (i + 1));
            SeeMarks [i] = sc.nextInt ();
        }
    }

    public void computeFinal () {
        for (int i = 0; i < 5; i++) {
            finalMarks [i] = SeeMarks [i] / 2 +
                internalMarks [i];
        }
    }
}
```

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

DemoMain.java

```
import See.External;  
  
public class DemoMain {  
  
    public static void main (String [] args) {  
        External obj = new External ();  
        obj.setDetails ();  
        obj.getDetails ();  
        obj.setCIC ();  
        obj.setSee ();  
        obj.computeFinal ();  
        obj.displayMarks ();  
        System.out.println ("Sohan T Sangeev", URN: 2023BMS021  
                           324);  
    }  
}
```

~~offer~~
Sohan T Sanjeev, USN : 2023BMS02532
2023 BMS 02532

enter the 15 marks of 15 subjects

50
40
35
45
50

40
35
45
50

40
35
45
50

40
35
45
50

40
35
45
50

40
35
45
50

40
35
45
50

40
35
45
50

40
35
45
50

40
35
45
50

40
35
45
50

40
35
45
50

40
35
45
50

40
35
45
50

40
35
45
50

X
30/11/2024

(35)

LAB-7

QAP to demonstrate handling of exception in inheritance tree. Create a base class called "Father" & derived class "Son" which extends base class. In father class, implement a constructor that check age, throw exception wrongage(). In son class, implement a constructor that check age condition and throw exception wrongage().

Import java.util.Scanner;

class WrongAge extends Exception {

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

 3
 2

class Father {

 int fatherAge;

 Scanner sc = new Scanner (System.in);

 public void validAge () throws WrongAge {

 System.out.println ("Enter father's age");

 fatherAge = sc.nextInt();

 if (fatherAge <= 0) {

 throw new WrongAge ("Invalid father's age");

(36)

```
class son extends father {  
    int sonAge;  
    Scanner sc = new Scanner(System.in);  
    public void validAge() throws WrongAge {  
        System.out.println("Enter son's age");  
        sonAge = sc.nextInt();  
        super.validAge();  
        if (sonAge >= fatherAge) {  
            throw new WrongAge("Son's age can't  
            be greater than father's age");  
        }  
        else if (sonAge < 0) {  
            throw new WrongAge("Invalid son  
            age");  
        }  
    }  
  
public class MyMain {  
    public static void main(String[] args) {  
        Son obj = new Son();  
        try {  
            obj.validAge();  
        }  
    }  
}
```

catch (WrongAge e) & System.out.println ("exception message")
System.out.println ("exception message")

you will get

4
Exception message from catch block

4
Exception message from catch block

Exception message from catch block

Output

Java mymain.

Enter sons age

40

Enter fathers Age

20

Exception sons age can't be greater than
fathers age

Y
30/11/2024

32

Lab-8

Write a program which creates two threads displaying "BMS college of Engineering" once every 5 seconds and another displaying "CSE" once every two seconds

```
class Bms extends Thread {  
    public void run() {  
        for (int i=0; i<10; i++) {  
            this.start();  
            try {  
                Thread.sleep(10000);  
            } catch(InterruptedException e) {  
                System.out.println(e);  
            }  
            System.out.println(i);  
        }  
    }  
}
```

```
class Cse extends Thread {  
    public void run() {  
        for (int i=0; i<10; i++) {  
            this.start();  
            try {  
                Thread.sleep(2000);  
            } catch (InterruptedException e) {  
                System.out.println(e);  
            }  
            System.out.println(i);  
        }  
    }  
}
```

```
class Threads extends Thread {  
    public static void main (String args[]) {  
        Bms b = new Bms ();  
        Cse c = new Cse ();  
        b.start();  
        c.start();  
        for (int i=0; i<10; i++) {  
            try {  
                System.out.println(i);  
            } catch (Exception e) {  
                System.out.println(e);  
            }  
        }  
    }  
}
```

D.P.
BMS college of Engineering

CSE
CSE
CSE
CSE
CSC

BMS college of Engineering

CSE
CSE
CSC
CSC
CSC

BMS college of Engineering

Lab - 10

Demonstrate Inter process communication
deadlock

class Q {

int n;

Synchronized int get() {

System.out.println("got : " + n);

return n;

}

Synchronized void put(int n) {

this.n = n;

System.out.println("put : " + n);

}

}

class Producer implements Runnable {

Q q;

producer(Q q) {

this.q = q;

new Thread(this, "producer").start();

y

public void run() {

④

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

y

y

Consumer implements Runnable {

class

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();

i++;

y

y

y

```
class Interprocess {  
    public static void main (String args[]){  
        Q q = new Q();  
        new producer (q);  
        new consumer (q);  
        System.out.println ("press control - C to  
                           stop");  
    }  
}
```

QP:

put: 0

got: 0

put: 1

got: 1

put: 2

got: 2

put: 3

got: 3

put: 4

got: 4

put: 5

got: 5

13.03.2014

(64)

Lab - program 10

Demonstrate Deadlock.

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

```
class B {  
    synchronized void bar(A a) {  
        String name = Thread.currentThread().  
            getName();  
        System.out.println(name + " entered B's bar");  
        try {  
            Thread.sleep(1000);  
        } catch (InterruptedException 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("Main Thread");
 Thread t = new Thread(this, "Racing Thread");
 t.start();

 a.foo(b);

 System.out.println("Back in main Thread");

³

 public void run() {

 b.bar(a);

 System.out.println("Back in other Thread");

³

 public void main(String args[]) {

 new DeadLock();

9

(47)

Off

13.02.24 10:00 AM - Java Concurrency

Main Thread Entered A.job

Racing Thread entered B.bar

Main Thread trying to call B.last()

Inside A.last

Back in main Thread

Racing thread trying to call A.last()

Inside B.last

Back in other thread

~~13.02.24 10:00 AM - Java Concurrency~~

Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the result field when the divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormat Exception. If Num2 were zero, the program would throw an ArithmeticException displaying the exception in a message dialog box.

```

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 glab = new JLabel("Enter the
        Dividend and Divident");
    }
}

```

```
JTextField af_tf = new JTextField(8);  
JTextField bf_tf = new JTextField(8);  
JButton button = new JButton("Calculate");  
  
JLabel err = new JLabel();  
JLabel alab = new JLabel();  
JLabel blab = new JLabel();  
JLabel anslab = new JLabel();  
  
frm.add(err);  
frm.add(alab);  
frm.add(af_tf);  
frm.add(bf_tf);  
frm.add(button);  
frm.add(blab);  
frm.add(anslab);
```

```
Action Listener1 = new ActionListener()  
{  
    public void actionPerformed(ActionEvent evt)  
    {  
        System.out.println("Action Event from  
        Text field");  
    }  
};  
af_tf.addActionListener(L1);  
bf_tf.addActionListener(L1);
```

```
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        try {
            int a = Integer.parseInt(tf1.getText());
            int b = Integer.parseInt(tf2.getText());
            int ans = a/b;
            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!");
        }
        gfrm.setVisible(true);
    }
});
```

61

```

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

```

ops c:\005> javac SwingDemo.java
c:\005> java SwingDemo

Enter: Action event from a Text Field

enter the Divider and Dividend

 $A = 10 \quad B = 5 \quad Ans = 2$

Enter the Divider and Dividend

 B Should be non zero!

Enter the Divider and Dividend

 ~~Enter only integers!~~ (32)

~~Enter only integers!~~

functions :-

(1) JFrame :- JFrame class in Java is a part of Swing framework, which is used for creating graphical user interface (GUI) in Java Applications. It is a top level container that contains main window of an application.

(2) setSize(int width, int height) method is used to get the size of a graphical user component, such as JFrame in Swing. This method takes two parameters, width & height of component specified in pixels.

(3) setLayout :- SetLayout (Layout Manager) is a method of the Container which chooses how components inside a window are arranged.

(4) JLabel :- Displays the text or images on the window of graphical User Interface (GUI).

(5) JTextField :- provides an editable text box for user input.

(6) Addframe :- Used to add new frame

X
2021/24

(53)