B. M. S. COLLEGE OF ENGINEERING

(AUTONOMOUS COLLEGE UNDER VTU, BELGAUM)
BANGALORE – 560019

2022-23

LAB RECORD

OBJECT ORIENTED JAVA PROGRAMMING (23CS3PCOOJ)

Submitted by :

NAME : SHREYASH SINHA

USN : 1BM22CS273
SEMESTER: III
SECTION : E

Submitted to
Dr. Seema Patil Assistant Professor Dept. of CSE, BMSCE

INDEX

OOJ LAB

1BM22CS243

NAME: Shreyash Sinha STD.: _____ SEC.: _____ ROLL NO.: _____

S. No.	Date	Title	Page No.	Teacher's Sign / Remarks
1	5/12/23	Lab Prog 1		
2	12/12/23	Lab Prog 2		
3	19/12/23	Lab Prog 3		
4	26/12/23	Lab Prog 4		
5	2/1/24	Lab Prog 5		
6	9/1/24	Lab Prog 6		
7	16/1/24	Lab Prog 7		
8	23/1/24	Lab Prog 8		
9	30/1/24	Lab Prog 9		
10	06/2/24	Lab Prog 10		
11	13/2/24	Lab Prog 11		
12	20/1/24	Lab Prog 12		

realme Shot by Shreyash' Realme
2024 02 24 20:26

```
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 coefficients of a, b
            and c");
        a = s.nextInt();
        b = s.nextInt();
        c = s.nextInt();
    }
    void compute()
    {
        while(a==0)
        {
            System.out.println("Not a quadratic eq^n");
            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)))/(double)(2*a);
    r2 = ((-b) - (Math.sqrt(d)))/(double)(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+ "i" +r2);
    System.out.println("Root2 =" +r1+ "i" +r2);
}
}
class QuadraticMain
{
    public static void main(String args[])
    {
        Quadratic q = new Quadratic();
        q.getd();
        q.compute();
        System.out.println("Shreyash Sinha
            1BM22CS273");
    }
}
```

Output :

1) Output 1 : Enter the coefficients of a, b and c

1

-2

1

Roots are real and equal

Root 1 = Root 2 = 1.0

Shreyash Sinha 1BM22CS273

2) Output 2 : Enter the coefficients of a, b and c

0

1

2

Not a quadratic equation

Enter a non zero value for a

-1

Root are real and distinct

Root 1 = -1.0, Root 2 = 2.0

Shreyash Sinha 1BM22CS273

3) Output 3 : Enter the coefficients of a, b and c

1

2

4

Roots are imaginary

Root 1 = -1.0 + i1.73205

Root 2 = -1.0 - i1.73205

Shreyash Sinha 1BM22CS273

10

Sw 12/12/23

2) develop a JAVA prog. to create class Student with members usn, name, array credits, array marks. Include methods to accept and display details and method to display SGPA of student.

$$SGPA = \frac{\sum[\text{course credits}][\text{gradept}]}{\sum[\text{course credit}]}$$

```
import java.util.Scanner;
public class Subject {
    int subjectMarks;
    int credits;
    int grade;
}
public class Student {
    Subject subject[];
    String name;
    String usn;
    double sgpa;
    Student() {
        subject = new Student[8];
        for(int i = 0; i < 10; i++) {
            subject[i] = new Student();
            Scanner s = new Scanner(System.in);
        }
    }
void getStudentDetails() {
System.out.print("Enter your name");
name = s.next();
System.out.print("Enter your usn");
usn = s.next();
}
```

realme Shot on realme 9 Pro+

```
void getMarks() {
    for(int i=0; i<8; i++)
    {
      System.out.print("Enter marks for
          subject" + (i+1) + ":");
      Subject[i].subjectMarks = s.nextInt();
      System.out.print("Enter credit for
          subject " + (i+1) + ":");
      subject[i].credit = 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 = 4;
    }

}
void computeSGPA() {
    int effectiveScore = 0;
    int totalCredits = 0;
    for(int i=0; i<8; i++)
    {
      effectiveScore += (subject[i].grade *
          subject[i].credits);
      totalCredits += subject[i].credits;
    } (double)
  SGPA = effectiveScore / (double) totalCredits;
}
class Main {
    public static void main(String args[])
    {
      Student s1 = new Student();
```

realme Shot on realme 9 Pro+

```
s1.getStudentDetails();
s1.getMarks();
s1.computeSGPA();
System.out.println("Name : " + s1.name+
"\n" "USN : " + s1.usn + "\n" SGPA : "
s1.SGPA);
}
```

Enter your Name Shreyash
Enter your USN 1BM22CS273
Enter marks for subject 1: 88
Enter credits for subject 1 4
Enter marks for subject 2 89
Enter credits for subject 2 4
Enter marks for subject 3 92
Enter credits for subject 3 3
Enter marks for subject 4 90
Enter credits for subject 4 3
Enter marks for subject 5 87
Enter credits for subject 5 3
Enter marks for subject 6 86
Enter credits for subject 6 2
Enter marks for subject 7 89
Enter credits for subject 7 2
Enter marks for subject 8 95
Enter credits for subject 8 1
Name : Shreyash
USN: 1BM22CS273
SGPA : 8.318

10

realme Shot on realme 9 Pro+

Price : 300
Number of pages : 400
Shreyash Sinha 1BM22CS273

Output 2: 2
One 8
Virat
200
300
Biograph
Annee
200
400
Book name : One 8
Author name : Virat
Price : 200
Number of Pages : 300
Book Name : Biography
Author name : Anne
Price : 300
Number of pages : 400
Shreyash Sinha 1BM22CS273

(10)

CS273

26/12/23

```
class Main
{
  public static void main(String args[])
  {
    Scanner s = new Scanner(System.in);
    int n; String name; String author;
    int price; int numPages;
    n = s.nextInt();
    Books b[];
    b = new Books[n];
    for(int i=0; i<n; i++){
        name = s.next();
        author = s.next();
        price = s.nextInt();
        numPages = s.nextInt();
        b[i] = new Books(name, author,
          price, numPages);
    }
    for(int i=0; i<n; i++)
    {
      System.out.print(b[i].toString());
    }
  }System.out.println("Shreyash Sinha 184210073");
}
```

Output:
1
Dogaan
Ashneer
300
400
Book name: Dogaan
Author: Ashneer

Output

10

26/12/

realme Shot on realme 9 Pro+

LAB-3

Q) create a book class which contains 4 members : name, author, price, num pages. Include constructor to set values for members. Include methods to set and get details of objects. Include a toString() method that could display complete details of the book. develop JAVA program to create n book objects.

```
import java.util.Scanner;
class Books
{
    String name; String author;
    int price; int numPages;
    Books(String name, String author, int price,
          int numPages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }

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

realme Shot on realme 9 Pro+

```
s.displayArea();
t.displayArea();
c.displayArea();
System.out.println("Sheeyash Sinha
        1BM22CS273");
    }
}
```

Output:

```
Enter dimensions of the rectangle (length
    and breadth):
2 3
Enter dimensions of the triangle (base and
    height):
2 4
Enter dimensions of the circle (radius):
3
Area of Rectangle: 6.0
Area of Triangle: 4.0
Area of Circle: 28.2599
Sheeyash Sinha 1BM22CS273
```

02/01/24

realme Shot on realme 9 Pro+

```
class Triangle extends Shape{
{   void getInput(){
        System.out.println("Enter dimensions
          of triangle(base and height): ");
        a = sc.nextDouble();
        b = sc.nextDouble();
    }
    void displayArea(){
        System.out.println("Area of Triangle: "
            +(0.5*a*b));
    }
}
class Circle extends Shape
{
    void getInput(){
    System.out.println("Enter dimensions of
      the circle(radius): ");
    a = sc.nextDouble();
    }
    void displayArea(){
    System.out.println("Area of circle: "+(3.14*a*a
                                        ));
    }
}
class Main{
    public static void main(String args[])
    {
    Rectangle r = new Rectangle();
    Circle c = new Circle();
    Triangle t = new Triangle();
    r.getInput();
    t.getInput();
    c.getInput();
```

LAB-4

8) develop a JAVA program to create an abstract class named Shape that contain 2 integers and an empty method named displayArea(). Provide three classes name Rectangle, Triangle and Circle such that each one of the classes contains only method getInput and displayArea for taking dimensions as input and prints area.

→
```
import java.util.Scanner;
class InputScanner{
    Scanner sc;
    InputScanner(){
        sc = new Scanner(System.in);
    }
}
abstract class Shape extends InputScanner{
    double a;
    double b;
    abstract void getInput();
    abstract void displayArea();
}
class Rectangle extends Shape{
    void getInput(){
        System.out.println("Enter dimensions of the rectangle (length and breadth): ");
        a = sc.nextDouble();
        b = sc.nextDouble();
    }
    void displayArea(){
        System.out.println("Area of Rectangle: "
            +(a*b));
    }
}
```

Output :

```
Enter customer name : Shreyash
Enter account number : 2
Enter the type of account : savings
--- MENU ---
1. Deposit
2. Withdraw
3. Display Account Details
4. Exit
Enter your choice : 1
Enter deposit amount : 1000
Deposit Successful
Interest computed and deposited : 1050.0
Enter your choice : 2
Enter withdrawal amount : 500
Withdraw Successful. Balance = 550.0
Enter your choice : 3
Customer name : Shreyash
Account number : 2
Type of Account : Savings
Balance : 550.0
```

```
switch (ch) {
case 1:
    System.out.println("Enter deposit
        account");
    double depoAmt = sc.nextDouble();
    userAccount.deposit(depositAmount);
    break;
case 2:
    System.out.println("Enter withdrawal
        amount");
    double withdrawalAmt = sc.nextDouble();
    userAccount.withdraw(withdrawAmt);
    break;
case 3:
    System.out.println("Customer name :"
        userAccount.customername);
    System.out.println("Account number:
        userAccount.accountNumber);
    System.out.println("Type of Account:"+
        userAccount.accountType);
    System.out.println("Balance:"+
        userAccount.balance);
case 4:
    System.out.println("Exiting prog");
    sc.close();
    System.exit(0);
}
}
}
```

```
public class Main{
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        int ch;
        Account userAccount = null;
        System.out.println("Enter customer
            name = ");
        String customername = sc.next();
        System.out.println("Enter account number
            ");
        int accountNumber = sc.nextInt();
        System.out.print("Enter the type of
            Account : ");
        String accounttype = sc.next();
        if(accountType.equals("savings")){
            userAccount = new SavAcct(
                customerName, accountNumber);}
        else if(accounttype.equals("current")){
            userAccount = new CurAcct(
                customerName, accountNumber);}
        else{
            System.out.println("Invalid account
                type");
            sc.close();
            System.exit(0);
        }
        while(true){
            System.out.println("---MENU---\n
                1. Deposit \n2. Withdraw \n3.
                Display Account Details \n4. Exit");
            System.out.print("Enter your choice : ");
            ch = sc.nextInt();
```

```
class SavAcct extends Account {
    double interestRate;
    SavAcct(String customerName, int
            accountNumber) {
        super(customerName, accountNumber,
            "savings")
        this.interestRate = 0.05;
    }
    @Override
    void deposit(double amount){
        super.deposit(amount);
        computeInterest();
    }
    void computeInterest(){
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest computed
            and deposited. Balance: " + balance);
    }
    void withdraw(double amount) {
        if(amount <= balance) {
            balance -= amount;
            System.out.println("Withdraw
        Successful. Balance=" + balance);
    }
    else {
        System.out.println("Insufficient
            funds for withdrawal");
    }
}
}
```

```
void displayBalance() {
    System.out.println("Account Balance: " +
        balance);
}
void withdraw(double amount) {}
}
class CurrAcct extends Account {
    double minbal;
    double serviceCharge;
    CurrAcct(String customername, int accountNumber)
    {
        super(customer name, account number,
            "Current");
        this.minBalance = 1000;
        this.serviceCharge = 50;
    }
    @Override
    void deposit(double amount){
        super.deposit(amount);
        checkMinimumBalance();
    }
    @Override
    void displayBalance(){
        super.displayBalance();
        checkMinimumBalance();
    }
    private void checkMinimumBalance {
        if(balance < minBalance) {
            balance -= serviceCharge;
            System.out.println("Service
                charge Imposed: Balance: " +
                balance);
        }
    }
}
```

→ Develop a JAVA prog to create a class Bank that maintains a kind of Account for its customers, savings and current account. saving acc provides compound interest and withdraw facilities. The current account provides check book facility. current acc holders should maintain minimum balance otherwise service charge imposed. class account contains customer name, account number, type of account and classes derived are Cur-acct and Sav-acct.

```
import java.util.Scanner;

class Account {
    String customerName;
    int accountNumber;
    String accountType;
    double balance;
{   Account(String customerName, int account
    Number, String accountType){
    this.customerName = customerName;
    this accountNumber = customerNumber;
    this.accountType = accountType;
    this.balance = 0;
    }
    void deposit(double amount){
    balance += amount;
    System.out.println("Deposit successful");
}
```

Date: Page:

Enter USN: 1BM22CS274
Enter Name: Shruti
Enter Semester: 2
Enter CIE Marks
Enter Internal Marks for Shruti
Subject 1: 36
Subject 2: 48
Subject 3: 50
Subject 4: 42
Subject 5: 55
Enter SEE Marks for Shruti
Subject 1 Marks: 80
Subject 2 Marks: 72
Subject 3 Marks: 85
Subject 4 Marks: 78
Subject 5 Marks: 90

// Display Data:
USN: 1BM22CS273
Name: Shreyash
Semester: 2
Subject 1 87
Subject 2 75
Subject 3 79
Subject 4 73
Subject 5 87
USN: 1BM22CS274
Name: Shruti
Semester: 2
Subject 1 98
Subject 2 89
Subject 3 92
Subject 4 81
Subject 5 100

23/01/24

```
for(int i=0; i< numofstudents; i++){
    finalMarks[i] = new External();
    finalMarks[i].inputStudentDetails();
    System.out.println("Enter CIE marks:");
    finalMarks[i].inputCIEMarks();
    System.out.println("Enter SEE marks:");
    finalMarks[i].inputSEEMarks();
}

System.out.println("Displaying data\n");
for(int i=0; i< numofstudents; i++){
    finalMarks[i].calculateFinalMarks();
    finalMarks[i].displayFinalMarks();
}
}
```

Output: Enter USN: 1BM22CS273
Enter Name: Sheyash
Enter Semester: 2
Enter Internal Marks for Sheyash
Subject 1 Marks: 40
Subject 2 Marks: 35
Subject 3 Marks: 42
Subject 4 Marks: 38
Subject 5 Marks: 45

Enter SEE marks for Sheyash
Subject 1 Marks: 75
Subject 2 Marks: 80
Subject 3 Marks: 65
Subject 4 Marks: 70
Subject 5 Marks: 85

```
public void inputSEEMarks() {
    Scanner s = new Scanner(System.in);
    System.out.println("enter see marks
        for " + name);
    for(int i = 0; i < 5; i++) {
        System.out.println("Subject " +
            +(i+1) + " marks: ");
        marks[i] = s.nextInt();
    }
}
public void calculateFinalMarks() {
    for(int i = 0; i < 5; i++){
        finalMarks[i] = marks[i]/2 + superMarks
                                    (i);
    }
public void displayFinalMarks() {
    displayStudentDetails();
    for(int i = 0; i < 5; i++){
        System.out.println("Subject "+(i+1)+"
            + finalMarks[i]);
    }
}
```

Main.java

```
import see.externals;
public class Main {
    public static void main(String args[])
    {
        int numOfStudents = 2;
    External finalMarks[] = new Externals
                    [numOfStudents];
```

realme Shot on realme 9 Pro+

Internals.java

```
package CIE;
import java.util.Scanner;
public class Internals extends Student {
    protected int marks[] = new int[5];
    public Internals() {
        // internals constructor
    }
    public void inputCIEmarks() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter internal marks
            for " + name);
        for(int i=0; i<5; i++) {
            System.out.println("Subject" + (i+1) +
                marks:");
            marks[i] = s.nextInt();
        }
    }
}
```

```
// Externals.java
package SEE
import CIE.Internals;
import JAVA.util.Scanner;
public class Externals extends Internals {
    protected int marks[];
    protected int finalMarks[];
    public Externals() {
        marks = new int[5];
        finalMarks = new int[5];
    }
```

LAB 6

Bafna Gold
Date: Page:

→ Create a package CIE which has 2 classes - Student and Internals the class Student has members usn, name, sem. class Internals derived from Student has array that stores internal marks scored in five courses of current sem of student. Create another package SEE having class external derived from class of Student class has an array that stores SEE marks scored in five courses of current semester of Student.

→ A student.java

```
package CIE;
import JAVA util. Scanner,
public class Student {
    protected String usn = new String();
    protected String name = new String();
    protected int sem;
    public void inputStudentDetails() {
    Scanner s = new Scanner(System.in);
    System.out.println("Enter USN:");
    usn = s.next();
    System.out.println("Enter name:");
    name = s.next();
    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);
    }
}
```

realme Shot on realme 9 Pro+

Internals.java

```
package CIE;
import java.util.Scanner;
public class Internals extends Student {
    protected int marks[] = new int[5];
    public Internals() {
        // internals constructor
    }
    public void inputCIEmarks() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter internal marks
            for " + name);
        for(int i=0; i<5; i++) {
            System.out.println("Subject" + (i+1) +
                marks: ");
            marks[i] = s.nextInt();
        }
    }
}
```

// External.java

```
package SEE;
import CIE.Internals;
import JAVA.util.Scanner;
public class External extends Internals {
    protected int marks[];
    protected int finalMarks[];
    public External() {
        marks = new int[5];
        finalMarks = new int[5];
    }
```

realme Shot on realme 9 Pro+

write a program that demonstrates handling of exceptions in inheritance tree. create base class father and derived class son who extends father. In father class, implement constructor that takes both father and son's age and throws an exception if son's age >= father's age

```
import java.util.Scanner
class WrongAge extends Exception {
    public WrongAge(String e) {
        super(e);
    }
}
class InputScanner {
    Scanner s = new Scanner(System.in);
}
class Father extends InputScanner {
    int fatherage;
    public Father() throws WrongAge {
        System.out.println("Enter father age:")
        fatherage = s.nextInt();
        if(fatherAge < 0) {
            throw new WrongAge(
            "Age cannot be negative");
        }
    }
    public void display() {
        System.out.println("Father age: "+
            fatherAge);
    }
}
```

Bafna Gold
Date: Page:

```
class Son extends Father {
    int sonage;
    public Son throws WrongAge{
      super();
      System.out.println("Enter son's age");
      sonage = s.nextInt();
      if(sonage >= fatherAge){
          throw new WrongAge("Son's age
          cannot be greater than father's");
      }
      else if(sonage < 0){
          throw new WrongAge("Age cannot
          be negative");
      }
    }
    public void display(){
        super.display();
        System.out.println("Sons Age: "+sonage
    }
}
public class Main{
    public static void Main(String args[]){
      try{
          Son s = new Son();
          s.display();
      }
      catch(wrongMessage e){
          System.out.println("Error: "+e.messag
      }
      System.out.println("Shreyash Sinha
                  1BM22CS273");
    }
}
```

Output:

Enter father's age: 23

Enter son's age: 24

Error: Son's age cannot be greater than father's age

Shreyash Sinha 1BM22CS273

Ans

30.01.24

Date: Page:

their age.

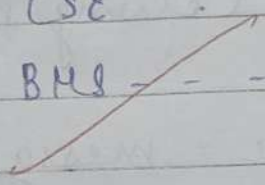
8) Write a program which creates two threads. one thread displaying "BMS College of Engineering" once every 10 sec and another displaying "CSE" once every 2 sec.

```
class Message implements Runnable {
    String message;
    int interval;
    public Message(String message, int interval)
    {
        this.message = message;
        this.interval = interval;
    }
@Override
public void run() {
    try {
        while(true) {
            System.out.println(message);
            Thread.sleep(interval);
        }
    }
    catch (InterruptException e) {
        System.out.println("Thread Interrupted");
    }
}
}
public class Lab8 {
    public static void main(String args[])
    {
        Thread thread1 = new Thread
            (new Message("BMS College of Eng",));
        thread1.start();
```

realme Shot by Shreyash' Realme 2024 02 17 19:46

```
Thread thread2 = new Thread( new Message
                    ( "CSE", 2000));
    }
}
```

Output : BMS College of Engineering
CSE
CSE
CSE
CSE
BMS - - - - - -



realme Shot by Shreyash' Realme 2024 02 17 19:47

LAB-9

Bafna Gold
Date: Page:

09) Write a JAVA prog that creates user interface to perform integer divisions. User enter two numbers in text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in result field. when division button clicked. If Num1 & 2 not were an integer, the program would throw NumberFormatException. If Num2 were 0, the program throw ArithmeticException display the exception in message dialog box.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

class SwingDemo {
    SwingDemo() {
        JFrame jfrm = new JFrame("Divide app");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation
                (JFrame.EXIT_ON_CLOSE);
        JLabel jlab = new JLabel("Enter
                dividee 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();
```

realme Shot by Shreyash' Realme 2024 02 24 10:26

```
jfm.add(ese);
jfm.add(jlab);
jfm.add(ajtf);
jfm.add(bjtf);
jfm.add(button);
jfm.add(alab);
jfm.add(blab);
jfm.add(anslab);
ActionListener l = new ActionListener(){
        public void actionPerformed(ActionEvent
                                    evt)
        {
          System.out.println("Action event
          from 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("In A= "+a);
            blab.setText("In B= "+b);
            anslab.setText("In Ans= "+ans);
        }
    catch(NumberFormatException e){
```

realme Shot by Shreyash' Realme 2024 02 24 10:26

```
        alab.setText("");
        blab.setText("");
        anslab.setText("");
        ui.setText("Enter only Integer");
    }
    catch(ArithmeticException e){
        alab.setText("");
        blab.setText("");
        anslab.setText("");
        ui.setText("B should be non zero");
    }
}
});
    jfm.setVisible(true);
}
public static void main(String args[]){
    SwingUtilities.invokeLater(new Runnable(){
        public void run(){
            new SwingDemo();
        }
    });
}
}
```

Output:

Enter the divider and divident

10 2

Calculate A = 10 B = 2 Ans = 5

realme Shot by Shreyash' Realme 2024 02 24 10:26

LAB-10

Bafna Gold
Date: Page:

a) demonstrate inter communication and deadlock

```
class Q {
    int n;
    boolean valueset = false;
    synchronised int get() {
        while (!valueset) {
        try {
            System.out.println("\n
                Consumer Waiting \n");
            wait();
        }
    catch(InterruptException e)
    {
        System.out.println("Interrupt
            Exception caught");
    }
    System.out.println("Got :" + n);
    valueSet = false;
    System.out.println("IntimateProducer\n");
    notify();
    return n;
}
synchronised void put(int n) {
    while (valueset)
        try {
            System.out.println(
                "Product Waiting");
        wait();
    }
    catch(InterruptException e)
```

realme Shot by Shreyash' Realme 2024 02 17 19:47

```
            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();
```

realme Shot by Shreyash' Realme 2024 02 17 19:47

```
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);
    }
}
```

Output:

Put: 0
Intimate Consumer
Producer Waiting
Got: 0
Intimate Producer
Consumed: 0
Consumer Waiting
Put: 1
Intimate Consumer
Producer Waiting
Got: 1
Intimate Producer
Consumed: 1
Put: 2
Intimate Consumer
Producer Waiting
Got: 2
Intimate Consumer
Consumed: 2
~~Producer Waiting~~
Put: 3
Intimate Consumer
Producer Waiting
Got: 3
Intimate Consumer
Consumed: 3
Put: 4

realme Shot by Shreyash' Realme
2024 02 17 19:47

Deadlocking

```
class A {
    synchronised 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 {
    synchronised void bar(A a) {
    String name = Thread.currentThread().
            getName();
    System.out.println(name +"
            entered B.bar");
    try {
        Thread.sleep(1000);
    }
```

2024 02 17 19:47
realme Shot by Shreyash' Realme

```
catch B(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);
  System.out.println("Back in
    main Thread");
}
```

Bafna Gold
Date: Page:
2024 02 17 19:47
realme Shot by Shreyash' Realme

1. Develop a Java program that prints all real solutions to the quadratic equation ax 2 +bx+c = 0. Read in a, b,c and use the quadratic formula. If the discriminate b 2 -4ac is negative, display a message 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))) / (double) (2 * a);
```

```
            r2 = ((-b) - (Math.sqrt(d))) / (double) (2 * a);
            System.out.println("Roots are real and distinct");
            System.out.println("Roo1 = " + 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 + " + i" + r2);
            System.out.println("Root1 = " + r1 + " - i" + r2);
        }

    }
}

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

        q.compute();
        System.out.println("Shreyash  S i n h a  1 B M 2 2 C S 2 7 3 ");
    }
}
```

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

```
import java.util.*;


class Subject {
   int subjectMarks;
   int credits;
   int grade;
}


class Student {
   Subject subject[];
   String name;
   String usn;
   double sgpa;
   Scanner sc;


   Student() {
      int i;
      subject = new Subject[8];
      sc = new Scanner(System.in);
      for (i = 0; i < 8; i++)
         subject[i] = new Subject();
   }


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

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

    void computeSGPA() {
        int effectiveScore = 0;
        int totalCredits = 0;
        for (int i = 0; i < 8; i++) {
            effectiveScore += (subject[i].grade * subject[i].credits);
            totalCredits += subject[i].credits;
        }
        sgpa = (double) effectiveScore / (double) totalCredits;
    }

}

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

```
        s1.getMarks();
        s1.computeSGPA();
        System.out.println("Name=" + s1.name);
        System.out.println("USN:" + s1.usn);
        System.out.println("SGPA=" + s1.sgpa);
    }
}
```

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

```
import java.util.Scanner;


class Books
{
	String name;
	String author;
	int price;
	int numPages;

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

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

```
public class Mainbook
{
	public static void main(String args[])
	{
		Scanner s=new Scanner(System.in);
		int n;
		int i;
		String name;
		String author;
		int price;
		int numPages;

		System.out.println("Enter the number of books:");
		n=s.nextInt();

		Books b[];
		b=new Books[n];

		for(i=0;i<n;i++)
		{
			System.out.println("Enter the details of book" + (i+1) + ":");
			System.out.println("Enter the name of the book:");
			name=s.next();
			System.out.println("Enter the author name:");
			author=s.next();
			System.out.println("Enter the price:");
			price=s.nextInt();
			System.out.println("Enter the number of pages:");
			numPages=s.nextInt();
```

```
                b[i]=new  Books(name,author,price,numPages);
            }

            System.out.println("Book Details:");
            for(i=0;i<n;i++)
            {
                System.out.println(b[i]);
            }
        }
}
```

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.*;
import  java.math.*;

class  InputScanner  {

   Scanner sc;

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

}

abstract class Shape extends InputScanner {

   double a;

   double b;

   abstract void getInput();

   abstract  void  displayArea();

}

class Rectangle extends Shape {
   void getInput() {
```

```
        System.out.println("Enter the length and breadth:");
        a = sc.nextDouble();
        b = sc.nextDouble();
    }

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

class Triangle extends Shape {
    void getInput() {
        System.out.println("Enter the length and height:");
        a = sc.nextDouble();
        b = sc.nextDouble();
    }

    void displayArea() {
        System.out.println("Area of triangle is :" + (a * b * 0.5));
    }
}

class Circle extends Shape {
    void getInput() {
        System.out.println("Enter the radius:");
        a = sc.nextDouble();
    }

    void displayArea() {
        System.out.println("Area of circle is :" + (a * a * Math.PI));
    }
```

```
}


class ShapeMain {
    public static void main(String[] args) {
        Rectangle r = new Rectangle();
        Triangle t = new Triangle();
        Circle c = new Circle();
        r.getInput();
        r.displayArea();
        t.getInput();
        t.displayArea();
        c.getInput();
        c.displayArea();
    }
}
```

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

- **Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:**
- **a)Accept deposit from customer and update the balance.**
- **b)Display the balance.**
- **c)Compute and deposit interest**
- **d)Permit withdrawal and update the balance**
- **Check for the minimum balance, impose penalty if necessary and update the balance.**

```
import java.util.Scanner;

class account {
   String 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,cant  withdraw");
        }
    }

    void display() {
        System.out.println("Name:" + name + "\nAccno:" + accno + "\nType:" + type + "\nBalance:" +
balance);
    }
}

class savAcct extends account {

    private static double rate = 5;

    savAcct(String name, int accno, double balance) {
        super(name, accno, "Savings", balance);

    }

    void interest() {
        balance += balance * (rate) / 100;
        System.out.println("Balance:"  +  balance);
    }
```

```
}

class curAcct extends account {

   private double minBal = 500;
   private double serviceCharges = 50;

   curAcct(String name, int accno, double balance) {
      super(name, accno, "Current", balance);

   }

   void checkmin() {

      if (balance < minBal) {
         System.out.println("Balance is less than min balance,service charges imposed:" +
serviceCharges);
         balance -= serviceCharges;
         System.out.println("Balance is:" + balance);
      }

   }

}

class Bank {
   public static void main(String a[]) {
      Scanner s = new Scanner(System.in);
      System.out.println("Enter the name,type(current/savings),account number,initial balance:");
      String name = s.next();
      String type = s.next();
```

```
int accno = s.nextInt();
double balance = s.nextDouble();
int ch;
double amount1, amount2;
account acc = new account(name, accno, type, balance);
savAcct sa = new savAcct(name, accno, balance);
curAcct ca = new curAcct(name, accno, balance);
while (true) {
   if (acc.type.equals("savings")) {
      System.out.println("\nMenu\n1.deposit 2.withdraw 3.compute interest 4.display");
      System.out.println("Enter the choice:");
      ch = s.nextInt();
      switch (ch) {
         case 1:
            System.out.println("Enter the amount:");
            amount1 = s.nextInt();
            sa.deposit(amount1);
            break;
         case 2:
            System.out.println("Enter the amount:");
            amount2 = s.nextInt();
            sa.withdraw(amount2);
            break;
         case 3:
            sa.interest();
            break;
         case 4:
            sa.display();
            break;
         case 5:
            System.exit(0);
```

```
        default:
            System.out.println("invalid input");
            break;
    }
} else {
    System.out.println("\nMenu\n1.deposit 2.withdraw 3.display");
    System.out.println("Enter the choice:");
    ch = s.nextInt();
    switch (ch) {
        case 1:
            System.out.println("Enter the amount:");
            amount1 = s.nextInt();
            ca.deposit(amount1);
            break;
        case 2:
            System.out.println("Enter the amount:");
            amount2 = s.nextInt();
            ca.withdraw(amount2);
            ca.checkmin();
            break;

        case 3:
            ca.display();
            break;
        case 4:
            System.exit(0);
        default:
            System.out.println("Invalid input");
            break;
    }
}
```

```
    }


  }
}
```

GENERICS

```
class GenericStack<T> {
    private Object[] stackArray;
    private int top = -1;
    private static final int MAX_SIZE = 5;

    public GenericStack() {
        stackArray = new Object[MAX_SIZE];
    }

    public void push(T value) {
        if (top < MAX_SIZE - 1) stackArray[++top] = value;
        else System.out.println("Stack is full. Cannot push more elements.");
    }

    @SuppressWarnings("unchecked")
    public T pop() {
        if (top >= 0)
            return (T) stackArray[top--];
        else {
            System.out.println("Stack is empty. Cannot pop more elements.");
            return null;
        }
    }

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

    public boolean isFull() {
        return top == MAX_SIZE - 1;
```

```
    }
}
class Main{
public static void main(String[] args) {
        GenericStack<Integer> integerStack = new GenericStack<>();
        GenericStack<Double> doubleStack = new GenericStack<>();

        // Push integers to the integer stack
        for (int i = 1; i <= 5; i++) {
            integerStack.push(i);
        }

        // Push doubles to the double stack
        for (double i = 1.0; i <= 5.0; i++) {
            doubleStack.push(i);
        }

        // Pop and print integers from the integer stack
        System.out.println("Popped integers from the stack:");
        while (!integerStack.isEmpty()) {
            System.out.println(integerStack.pop());
        }

        // Pop and print doubles from the double stack
        System.out.println("Popped doubles from the stack:");
        while (!doubleStack.isEmpty()) {
            System.out.println(doubleStack.pop());
        }
    }
}
```

Abstract class prog

```
import java.lang.Math;

abstract class Shape {
   double a;
   double b;
   double c;

   abstract void calculateArea();

   abstract void calculatePerimeter();
}

class Triangle extends Shape {
   Triangle(double x, double y, double z) {
      a = x;
      b = y;
      c = z;
   }

   void calculateArea() {
      double s = (a + b + c) / 2;
      System.out.println("Area=" + (Math.sqrt(s * (s - a) * (s - b) * (s - c))));
   }

   void calculatePerimeter() {
      System.out.println("Perimeter=" + (a + b + c));
   }
}

class Circle extends Shape {
```

```
    Circle(double r) {
        a = r;
    }

    void calculateArea() {
        System.out.println("Area=" + (Math.PI * a * a));
    }

    void calculatePerimeter() {
        System.out.println("Perimeter=" + (2 * Math.PI * a));
    }
}

class ShapeM {
    public static void main(String[] args) {
        Triangle t = new Triangle(2.0, 3.0, 5.0);
        Circle c = new Circle(5.0);
        t.calculateArea();
        t.calculatePerimeter();
        c.calculateArea();
        c.calculatePerimeter();
    }
}
```

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.

```
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 Internal Marks for " + name);
      for (int i = 0; i < 5; i++) {
         System.out.print("Subject " + (i + 1) + " marks: ");
         marks[i] = s.nextInt();
      }
   }
}
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() {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter USN: ");
        usn = s.next();
        System.out.print("Enter Name: ");
        name = s.next();
        System.out.print("Enter  Semester:  ");
        sem = s.nextInt();
    }

    public void displayStudentDetails() {
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Semester: " + sem);
    }
}
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);
        System.out.println("Enter SEE Marks for " + name);
        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]);
    }
}
import  SEE.Externals;

public class Pkgmain {
    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();
        }
    }
}
```

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<0. In Son class,

implement a constructor that cases both father and son's age and throws an exception if son's age is

>=father's age.

```
import java.util.Scanner;


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

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



class Father extends InputScanner
{
 protected int fatherAge;
 public Father() throws WrongAge
```

```
{
 System.out.println("Enter  Father's  Age:");
 fatherAge=s.nextInt();

 if(fatherAge<0)
 {
  throw new WrongAge("Age cannot be negetive:");
 }
}

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

}


class Son extends Father
{
 private int sonAge;

 public Son() throws WrongAge
 {
  super();
  System.out.println("Enter  Son's  age:");
  sonAge=s.nextInt();


  if(sonAge>fatherAge)
  {
```

```
   throw new WrongAge("Son's age cannot be greater than father's age");
  }
  else if (sonAge<0)
  {
   throw new WrongAge("Age cannot be negative");
  }
 }

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

}



public class FatherSonAge
{
 public static void main(String args[])
 {
  try
  {
   Son son=new Son();
   son.display();
  }

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

```
    }

}
```

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 DisplayMessageThread extends Thread {
   private final String message;
   private final long interval;

   DisplayMessageThread(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(30000);
        } catch (InterruptedException e) {
            System.out.println("Main  thread  interrupted.");
        }

        thread1.interrupt();
        thread2.interrupt();

        System.out.println("Main thread exiting.");
    }
}
```

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

```
import 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();
         }
      });
   }
}
```

10a) Interprocess communication using consumer and producer

```
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 < 6) {
            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 < 6) {
            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.");
    }
}
```

10b) 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 B.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();
    }
}
```