B.M.S COLLEGE OF ENGINEERING BENGALURU

Autonomous Institute, Affiliated to VTU

LAB REPORT

23CS3PCOOJ

Submitted in partial fulfilment of the requirements for LabBachelor of

Engineering

in

Computer Science and Engineering

Submitted by:

Prerana MK

(1BM22CS209)

Department of Computer Science and Engineering,

B.M.S College of Engineering,

Bull Temple Road, Basavanagudi, Bangalore, 560 0192023-2024.

INDEX

14/12/23

(8) → Lab programme 1 → Quadratic Equations

```
import java.util.scanner;
class Quadratic
{
    System.out.println ("Prerana, 1GH22CS209");
    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
        Scanner s = nr                    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 qual a
        System.out.println        equal");
        System.out.println ("Root1 = Root2=" + r1
    }
    else if (d>0)
    {
```

```
{
    System.out.println("roots are imaginary");
    r1 = ((-b) + (Math.sqrt(d))) / (double)(2*a);
    r2 = ((-b) - (Math.sqrt(d))) / (double)(2*a);
    System.out.println("roots are real and
                        distinct");
    System.out.println("root 1 = " + r1 + "root 2
                        = " + r2);
}
else if (d<0)
{
    System.out.println("roots are imaginary");
    r1 = (-b) / (2*a);
    r2 = Math.sqrt(-d) / (2*a);
    System.out.println("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;
    }
}
```

=> Output 1

Prerana
1BM22CS209
Enter the coefficients of a, b, c
1
-5
2
Roots are real and distinct
root1 = 4.561552812 8 root2 = 4.561552812 1

=> Output 2

Prerana 1BM22CS209
Enter the coefficients of a, b, c
1
2
1
roots are real and equal
root 1 = root 2 = -1.0

=> Output 3

Prerana 1BM22CS209
Enter the coefficients of a, b, c:
0
4
5
not a quadratic equation
enter a non zero value for a:
1
roots are imaginary
root1 = -2.0 + iNaN
root2 = -2.0 - iNaN

14/12/23

Calculating the SGPA of a student

```
import java.util.Scanner;

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

class Student {
    String name;
    String usn;
    double SGPA;
    Scanner s;
    Subject[] subjects;

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

    void getMarks() {
        for(int i=0 ; i<8 ; i++) {
            System.out.println ("Enter details for
            subject " + (i+1) + ":");
            System.out.println ("Enter Marks:");
            subjects[i].credits = s.nextInt();

            if (subject[i].subjectMarks >=90 ) {
                subject[i].grade =10; }
            else if (subject[i].subjectMarks >= 80 {
                subject[i].grade = 9; }
```

```
else if (subject[i].subjectMarks >= 70) {
        subject[i].grade = 8; }
else if (subject[i].subjectMarks >= 60) {
        subject[i].grade = 7; }
else if (subject[i].subjectMarks >= 50) {
        subject[i].grade = 5; }
else {
        subject[i].grade = 0; }
}
}

void computeSGPA() {
    double totalCredits = 0;
    double weightedSum = 0;

    for (int i = 0; i < 8; i++)
    {
        totalCredits += subject[i].credits;
        weightedSum += subject[i].grade *
                       subject[i].credits;
    }

void displayDetails()
{
    System.out.println("\n Student Details:");
    System.out.println("Name: " + name);
    System.out.println("USN: " + usn);
    System.out.println("SGPA: " + SGPA);
}
}
```

```
public class StudentMain
{
    public static void main (String [] args){
        Student s1 = new Student();
        s1.getStudentDetails();
        s1.getMarks();
        s1.computerSGPA();
        s1.displayDetails();
    }
}
```

→ output

```
Enter student Name : Pravana
Enter USN : 1KM22CS209
Enter details for Subject 1
Enter Marks : 98
Enter Credits : 4
Enter details for Subject 2
Enter Marks : 96
Enter credits : 4
Enter details for Subject 3
Enter Marks : 96
Enter Credits : 3
Enter details for Subject 4
Enter Marks : 87
Enter Credits : 3
Enter details for Subject 5
Enter Marks : 87
Enter Credits : 3
```

Date
Page

=> Enter details for subject 6:
Enter Marks : 90
Enter Credits : 1

=> Enter details for subject 7:
Enter Marks : 95
Enter Credits : 1

=> Enter details for subject 8:
Enter Marks: 94
Enter Credits : 1

Student details :
Name: Prerana
USN : 1BM22CI209
SGPA : 9.7

19/12/23

14/12/23 Lab Program 3

7) 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 objects. Include toString() method that could display the complete details of the books.

```
import java.util.Scanner;
class Books {
    String name;
    String author;
    int price;
    int numpages;

    Book (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 = "no. of pages" + this.numpages +
                   "\n";
        return name + author + price + numPages;
    }
```

```
}

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

        System.out.println ("Enter no. of books");
        n = s.nextInt();

        Book[] books = new Book[n];
        for (int i=0; i<n; i++) {
            System.out.println ("Enter details
            for book" + (i+1) + ":");
        System.out.println ("Name:");
        name = s.next();
        System.out.println ("Author:");
        author = s.next();
        System.out.println ("Price");
        price = s.nextInt();
        System.out.println ("Number of pages")
        numPages = s.nextInt();
        books[i] = new Book (name, author, price,
                             numPages);
        }

    System.out.println ("\n Book details:");
        for (int i=0; i<n; i++) {
            System.out.println (books[i].toString

    }
}
```

→ output

19

Enter the number of books 2

Enter details for book 1:

Name:

Dracula

Author

Stoker

Price

340

No. of pages

560

Enter details for book 2

Name:

Frankenstein

Author:

Shelly

Price:

523

No. of pages

560

Book details:

-) Book name : Dracula

Author name : Stoker

price : 340

no. of pages : 560

20/12/23

-) Book name: Frankenstein

Author name : Shelly

price : 523

no. of pages 456

4/01/2024 Lab 4

Date
Page

Q) Develope a Java program to create abstract class named Shape that contains 2 integers and an empty method named printArea(). Provide three classes named Rectangle, Circle and Triangle such that each one of the classes extends the class Shape. Each one of the class contain only the method printArea() that print the area of the given shape.

1. Add class InputScanner
2. Add class Shape extend InputScanner
3. Add class Rectangle extends Shape
4. Add class ~~Rectangle~~ Triangle extends Shape
5. Add class Circle extends Shape
6. Add main class invoking main method with objects and calling input method and findarea method.

```
import java.util.Scanner;
abstract class InputScanner
{
    abstract void get_dim();
}

abstract class Shape extends InputScanner
{
    int a, b;
    Shape(int a, int b)
    {
        this.a = a;
        this.b = b;
    }
    Shape(int a)
```

```
    {
        this.a = a;
        this.b = 0;
    };
    abstract void printArea();
}

class Rectangle extends Shape
{
    Rectangle(int a, int b)
    {
        super(a,b);
    }
    void get_dim()
    {
        System.out.println("Enter dimensions of
        the rectangle");
        Scanner s = new Scanner(System.in);
        a = s.nextInt();
        b = s.nextInt();
    }
    void printArea()
    {
        System.out.println("Area of rectangle = "
        + (a*b));
    }
}
class Triangle extends Shape
{
    Triangle(int a, int b)
    {
        super(a,b);
    }
```

```
void get_dim()
{
    System.out.println("Enter the dimensions of
    the triangle (base and height)");
    Scanner s = new Scanner(System.in);
    a = s.nextInt();
    b = s.nextInt();
}
void printArea()
{
        System.out.println("Area of triangle="
                + (0.5 * a * b));
}
}
class circle extends shape
{
    circle(int a)
    {
        super(a);
    }
    void get_dim()
    {
        System.out.println("Enter the dimensions
        of the circle (radius)");
        Scanner s = new Scanner(System.in);
        a = s.nextInt();
    }
}
class shapeMain
{
    public static void main(String args[])
    {
```

```
Rectangle r = new Rectangle (0,0);
Triangle t = new Triangle (0,0);
circle c = new circle (0);
r.get_dim();
t.get_dim();
c.get_dim();
r.printArea();
t.printArea();
c.printArea();
}
}
```

→ o/p

Enter dim of rectangle (length and breadth);
2 3
Enter dim of Triangle (base and height):
2 4
Enter dim of circle (radius)
3

Area of ~~triangle~~ rectangle = 6
Area of triangle = 4
Area of circle = 28.25499

9/07/24

Java Lab Program 5 { Current Account and Savings Account }

```
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 = amont;
  }
  else
  {
   System.out.println ("insufficient balance, cant withdraw");
  }
}
```

```
void display()
{
    System.out.println("name: " + name + "accno:"
    "type: " + type + "balance: " + balance);
}
}

class savAcc 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 crAcct extends account
{
    private double minBal = 500;
    private double serviceCharge = 50;

crAcct(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: " + serviceCharge);
        System.out.println("balance is: " + balance);
    }
}
class accountMain
{
    public static void main (String a[])
    {
    Scanner s = new Scanner (System.in);
    System.out.println ("enter the name: ");
    String name = s.next();    type (current/savings):");
    System.out.println ("enter the, account number")
    String type = s.next();
    System.out.println ("enter the account number");
    int acno = s.nextInt();
    System.out.println ("enter initial balance");
    double balance = s.nextDouble();
    int ch;
    double amount1, amount2;
    account acc = new account (name, acno, type, balance);
    savAcct sa = new savAcct (name, acno, balance);
    crAcct ca = new crAcct (name, acno, balance);
    while (true)
    {
    if (acc type.equals ("savings"))
    {
```

```
System.out.println ("\n Menu \n 1.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 5: System.exit (0)
default: System.out.println ("invalid input");
break;

}
}

else
{
System.out.println ("\n Menu \n 1. deposit 2 withdraw 3. disp
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;
}
}
}
}
```

→ output

Enter the type of account (Savings or Current)

Savings

Enter the customer name:

Kumara

Enter account number

12345

MENU

1 Deposit 2 Withdraw 3. Compute interest 4 Display

5. Exit

Enter choice:

1

Enter the deposit amount

2000

Deposit of 2000 successful

MENU

1. Deposit 2. Withdraw 3. Compute Int cost 4. Display Acc details

5. Exit

Enter choice:

2

Enter withdrawal amount:

200

Withdrawal of 200 successful

Menu

1. Deposit 2. Withdraw 3 Compute interest 4. Display
5. Exit

Enter choice 3

Interest of 90 computed and added to account

Enter choice : 4

Customer Name Pravana

Account number : 1234

Account type : Savings

Balance : 1590.0

Menu

1 Deposit 2 Withdraw 3 Compute interest 4. Display Account details

5. Exit

Enter choice

5

Exiting...

16/01/24

Strings

(2) String 1 : Hello
String 2 : Java!

Concatenated string : Hello Java!
length : 11

(4) extracted substring : SMSCE

(3) 101 Raja Lucknow
102 Vijay ghaziabad

(5) GetBytes ("Hello world!")
Byte representation :
72 101 108 108 111 32 87 111 114 108 100

Character array representation
J a v a P r o g r a m m i n g

(6) Bmsce equals Bmsce - get true
Bmsce equals College - get false
Bmsce equals SMSCE - get false
Bmsce equalsIgnoreCase SMSCE - get true

(7) Substring is matched

(3) and (9)
Praveena starts with Pr : True
Praveena starts with Na : False
Praveena ends with ana : True
Praveena ends with rome : false

(10) check if two strings are equal

Using == operator : false

Using equals () : true

(1) str : Welcome

str1 : Java

str 2 : Hi Dinesh

st 3 : INE

str 4 : ABCDEFG

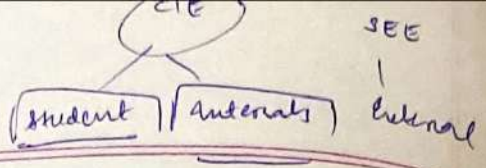
str 5 : CDE

str 6 :

str 7 : welcome

str : Welcome to java

16/01/24



→ 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 student. Import the 2 packages in a file that declares the final marks of n students in all five courses

```
//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.nextLine();

        System.out.println("Enter name:");
        name = s.nextLine();

        System.out.println("Enter semester:");
        sem = s.nextInt();
    }
```

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

//internals.java

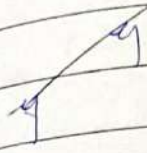
```
package CIE;
import java.util.Scanner;
public class Internals extends Student
{
    protected int marks[] = new int[5];
    public void inputCIEmarks()
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter CIE marks for subjects");
        for(int i=0; i<5; i++)
   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

- Student
- Internals

// External.java

```
package SEE
import CIE.Internals;
import java.util.Scanner;
public class Externals extends Internals
{
    protected int marks[];
    protected int finalMarks[];
    public Externals()
    {
        marks = new int[5];
        finalMarks = new int[5];
    }
    public void inputSEEMarks()
    {
        Scanner s = new Scanner(System.in);
        for(int i=0; i<5; i++)
        {
            System.out.println("Subject " + (i+1)
                + " marks");
            marks[i] = s.nextInt();
        }
    }
    public void calculateFinalMarks()
    {
        for(int i=0; i<5; i++)
            finalMarks[i] = marks[i]/2 +
                super.marks[i];
    }
```

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



```
// main.java

import SEE.Externals;

class Main
{
    public static void main(String args[])
    {
        int numofStudents = 2;
        Externals finalMarks[] = new
        Externals[numofStudents];
        for(int i=0; i<numofStudents; i++)
        {
            finalMarks[i] = new Externals()
            finalMarks[i].inputStudentDetails();
            System.out.println("Enter CIE marks");
            finalMarks[i].inputCIEmarks();
            System.out.println("Enter SEE marks");
            finalMarks[i].inputSEEmarks();
        }
        System.out.println("Display data:\n");
        for(int i=0; i<numofStudent; i++)
        {
            finalMarks[i].calculateFinalMarks();
            finalMarks[i].displayFinalMarks();
        }
    }
}
```

y

y

→ output

Enter the name
Kavya Prerana
Enter the sem
3
Enter CIE marks

Enter internal marks of course 1
45
Enter internal marks of course 2
46
Enter internal marks of course 3
47
Enter internal marks of course 4
48
Enter internal marks of course 5
49

Enter SEE Marks

Enter external marks of course 1
99
Enter external marks of course 2
85
Enter external marks of course 3
98
Enter external marks of course 4
87
Enter external marks of course 5
93

Enter the name
Ramesh
Enter the sem
5
→ Enter CIE Marks
Enter internal marks of course 1
45
Enter internal marks of course 2
43
Enter internal marks of course 3
46
Enter internal marks of course 4
50
Enter internal marks of course 5
44
→ Enter SEE Marks
Enter internal marks of course 1
48
Enter internal marks of course 2
49
Enter internal marks of course 3
47
Enter internal marks of course 4
49
Enter internal marks of course 5
48

→ Lab 7

```
import java.util.Scanner;
class WrongAge extends Exception {
    public WrongAge()
    {
        super("Age Error");
    }
    public WrongAge(String message)
    {
        super(message);
    }
}
class InputScanner {
    protected Scanner scanner;

    public InputScanner() {
        scanner = new Scanner(System.in);
    }
    public int nextInt() {
        return scanner.nextInt();
    }
}

class Father extends InputScanner{
    protected int fatherAge;

    public Father() throws WrongAge {
        System.out.print("Enter Father's Age:");
        fatherAge = scanner.nextInt();
```

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

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

class Son extends Father {
    private int sonAge;

    public Son() throws WrongAge {
     super ();

    System.out.print ("Enter son's age");
    sonAge = scanner.nextInt();
    if (sonAge >= fatherAge) {
        throw new WrongAge ("Son age cannot be
        greater than father's age");
    }

    if (sonAge < 0)
    {
        throw new WrongAge ("Age cannot be negative");
    }
}

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

```
public class ExceptionHandlingDemo {
    public static void main (String[] args)
    {
        try {
            Son son = new Son();
            son.display();
        } catch (WrongAgeE e) {
            System.out.println("Error: " +
                e.getMessage());
        }
    }
}
```

→ output

```
Enter Father's Age : 23
Enter Son's Age : 50
Error: Son's age cannot be greater than father's age

Enter Father's Age : 42
Enter Son's Age : -6
Error : Age cannot be negative

Enter Father's Age: -9
Error : Age cannot be negative
```

30.01.24

Lab 8

Date
Page

- Write a program which creates 2 threads, one thread displaying "SMS College of Engineering" once every 10 seconds and another displaying "CSE" once every 2 seconds

```
public class DisplayMessages {
    public static volatile boolean isRunning = true;

    public static void main (String [] args) {

        Thread bmsThread = new Thread (() -> (displayMessage
                                    (BMSCE, 2000));
        Thread cseThread = new Thread (() ->
                                    displayMessage
                                    ('CSE', 2000));

        bmsThread.start();
        cseThread.start();

        try {
            Thread.sleep (30,000)
        }
        catch (InterruptedException e) {
            e.printStackTrace();
        }
        isRunning = false;
    }

    private static void displayMessage (String message, long
                                        interval) {
        while (isRunning) {
            System.out.println (message);
            try {
                Thread.sleep (interval);
            }
```

```
catch ( InterruptedException e)
{
    e.printStackTrace();
}
}
}
}
```

→ Output

```
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
```

6/2/24

Lab 10

Date
Page

13/03/2023

9 Demonstrate Inter Process Communication and Deadlock

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

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

    synchronized void put(int n) {
        while (valueSet) {
            try {
                System.out.println("\n Producer Waiting\n");
                wait();
            } catch (InterruptedException e) {
                System.out.println("InterruptedExcption
                caught");
            }
        }
    }
```

```
this.n = n;
valueSet = true;
System.out.println("Put: " + n);
System.out.println("\n Notifying 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;
    Producer(Q q) {
        this.q = q;
        new Thread(this, "Producer").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("Consumed: " + r);
        i++;
    }
}
class PCFixed {
    public static void main (String args[]) {
        Q q = new Q()
        new Producer(q)
        new Consumer(q)
        System.out.println("Press Control-C to
                            stop.");
    }
}
```

→ Output

```
Press Control-C to stop.
Put: 0
Notifying Consumer
Producer waiting
Got: 0
Notifying Producer
Put: 1
Notifying Consumer
Producer waiting
```

consumed : 0

Got : 1

Notifying Consumer

~~Producer waiting~~

~~Got : 2~~

consumed : 1

Put : 2

Notifying Consumer

Producer waiting

Got : 2

Notifying Producer

consumer : 2

~~Produ~~ Put : 3

Notifying Consumer

Producer Waiting

Got : 3

Notifying Producer

consumed : 3

Put : 4

Notifying Consumer

Producer Waiting

Got : 4

Notifying Producer

consumed : 4

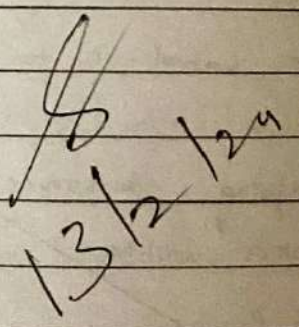
Put : 5

Notifying Consumer

Got : 5

Notified Producer

consumed : 5



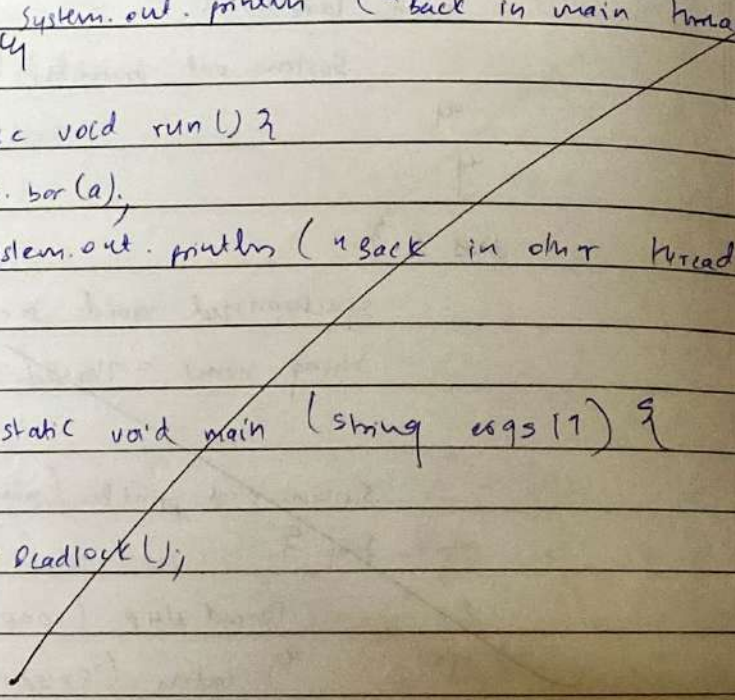
Date
Page

Cas 10

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(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");
    }

    public void run() {
        b.bar(a);
        System.out.println("back in other thread");
    }

    public static void main(String args[]) {

        new Deadlock();
    }
}
```



Main Thread enters A.foo
RacingThread unlend B.bar
MainThread trying to call B.last()
inside A.last
RacingThread trying to call A.last()
Inside A.last
Back in other thread
Back in main thread

13/2/24

LAB 9

Write a program that creates user interface to perform integer division. The user enters 2 number in the text field and the result is displayed by clicking on [/] division button

The exceptions are generated when the second number is 0 and when the text field is empty.

Make a report on following

(i) jframe, setsize, setlayout, setdefault, jtextfield, jlabel, add bar, add action listener, set text

→

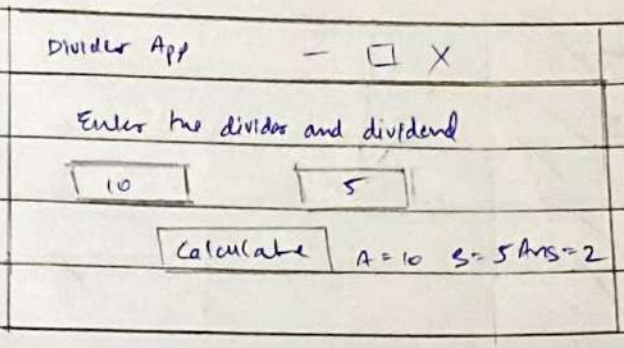
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

class UserInterface {
    UserInterface() {
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JLabel jlab = new JLabel("Enter the divider and
                                  dividend:");
        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);

        JButton button = new JButton("Calculate");
```

```
JLabel err = new JLabel();
JLabel alab = new JLabel();
JLabel blab = new JLabel();
JLabel anslab = new JLabel();

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

ActionListener calculateListener = 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("\n A = " + a);
        blab.setText("\n B = " + b);
        anslab.setText("\n Ans = " + ans);
        err.setText("");
    }
    catch (NumberFormatException e) {
        displayErrorMessage("Enter only Integer!");
    }
```

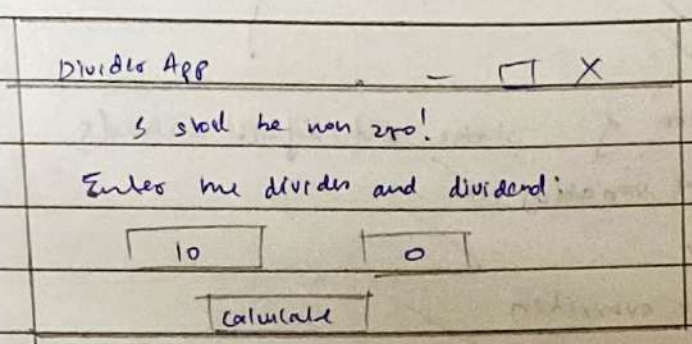
```
catch (ArithmeticException e) {
        displayErrorMessage ("s should be non zero");
}
}
private void displayErrorMessage (String Message)
{
    slab.setText("");
    blab.setText("");
    anslab.setText("");
    err.setText(message);
}
};

button.addActionListener (calculateListener);
    jfrm.setVisible(true);
}

public static void main (String[] args) {

SwingUtilities.invokeLater (new Runnable() {
        public void run() {
            new UserInterface();
        }
});
}
```

→ output

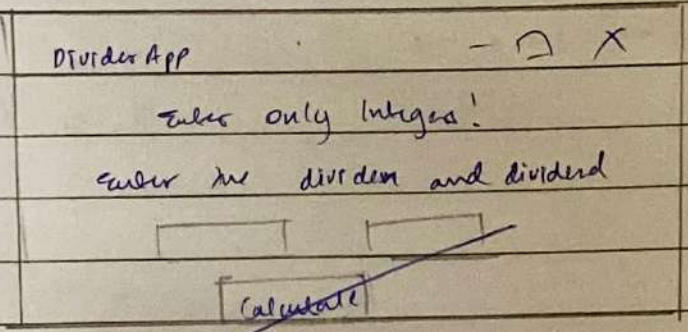
(i)


Divider App
Enter the divisor and dividend
10
5
Calculate
A=10 B=5 Ans=2

(ii)


Divider App
B shall be non zero!
Enter the divisor and dividend:
10
0
Calculate

(iii)


Divider App
Enter only Integers!
Enter the divisor and dividend
Calculate

Definition of the following functions

1. JFrame : It is a class that represents the window containing the graphical user interface. In this program 'JFrame' is created with the name "Divider App" and assigned to the variable 'jfrm'.

2. setSize : 'setSize' is a method of 'JFrame' class that sets the size of the frame. In this program, it is used to set the size of the frame to 275 pixels in width and 150 pixels in height.

3. setLayout : 'setLayout' is a method of the 'Container' class used to set the layout manager for the container. In this program 'FlowLayout' layout manager is set.

4. setDefaultCloseOperation : It is a method of the 'JFrame' class that sets the default operation when the frame is closed. In this program, 'JFrame.EXIT_ON_CLOSE' is set, meaning the application will terminate when frame is closed.

5. JLabel : 'JLabel' is a class used to display non editable text or image.

6. JTextField : It is a class used to create a textfield component that allows the user to enter a text. In this program two 'JTextField' objects ('ajtf' and 'bjtf') are created to input dividend and divider

7 add: 'add' is a method of the 'Container' class used to add components to the container

8 ActionListener: It is an interface used to handle Action Events

9 setText. It is a method of the JLabel class used to set the text of the label dynamically. In this program 'setText' is used to update the labels 'alab', 'blab' and 'anslab' with calculated values or error messages.

LAB 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("Roo1 = 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("Root2 = "+ r1 + " - i "+r2);
        }
    }
}
class QuadraticMain
{
    public static void main(String args[])
    {
        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 marks. Include methods to accept and display details and a method to calculate SGPA of a student.

```
import java.util.Scanner;
class Subject
{
	int marks;
	int credits;
	int grade;
}
class Student
{
	Subject sub[];
	String name;
	String usn;
	double SGPA;
	double n_sum=0;
	double d_sum=0;
	Scanner s;

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

```
void getDetails()
{
        System.out.println("Enter student name:");
        name=s.nextLine();
        System.out.println("Enter student usn:");
        usn=s.nextLine();
}

void getMarks()
{
        int i;
        int numerator;
        for(i=0;i<8;i++)
        {
                System.out.println("Enter marks:");
                sub[i].marks = s.nextInt();
                System.out.println("Enter credits:");
                sub[i].credits = s.nextInt();
                sub[i].grade=sub[i].marks//10+1;
                if(sub[i].grade<4 || sub[i].grade>10)
                        sub[i].grade=0;
                numerator= sub[i].credits * sub[i].grade;
                n_sum = n_sum + numerator;
                d_sum = d_sum + sub[i].credits;
        }
}

void computeSGPA()
{
        SGPA= n_sum / d_sum;
        System.out.println("SGPA= "+SGPA);
```

```
        }
}

class mainClass
{
public static void main(String args[])
{
        Student s1 = new Student();
        s1.getDetails();
        s1.getMarks();
        System.out.println("SGPA= "+s1.computeSGPA());
}}
```

LAB 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 title;
	String author;
	int price;
	int num_pages;

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

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

```
}

class Main
{
	public static void main(String args[])
	{
		Scanner s= new Scanner(System.in);
		int n,i;
		String title, author;
		int price, num_pages;

		System.out.println("Enter the no. of books: ");
		n= s.nextInt();

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

		for( i=0 ; i<n ; i++)
		{
			System.out.println("Enter the title, author name, price and number of pages of book:");
			title= s.nextLine();
			author= s.nextLine();
			price= s.nextInt();
			num_pages= s.nextInt();
			b[i] = new Books(title,author,price,num_pages);
		}

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

```
        }
}
```

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;
abstract class Shape
{
	int dim1;
	int dim2;
	Scanner s= new Scanner(System.in);
	abstract void printArea();
	abstract void input();
}

class Rectangle extends Shape
{
	void input()
	{
		System.out.println("Enter length and breadth:");
		dim1=s.nextInt();
		dim2=s.nextInt();
	}
	void printArea()
	{
		System.out.println("Area of rectangle= "+(dim1*dim2)+ " sq units");
	}
}
```

```
class Triangle extends Shape
{
	void input()
	{
		System.out.println("Enter base and height:");
		dim1=s.nextInt();
		dim2=s.nextInt();
	}
	void printArea()
	{
		System.out.println("Area of triangle= "+(dim1*dim2/2)+ " sq units");
	}
}

class Circle extends Shape
{
	void input()
	{
		System.out.println("Enter radius:");
		dim1=s.nextInt();
	}
	void printArea()
	{
		System.out.println("Area of circle= "+(3.14*dim1*dim1)+ " sq units");
	}
}

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

```
                Rectangle r= new Rectangle();
                Triangle t= new Triangle();
                Circle c= new Circle();
                Shape ref;
                ref=r;
                ref.input();
                ref.printArea();
                ref=t;
                ref.input();
                ref.printArea();
                ref=c;
                ref.input();
                ref.printArea();
        }
}
```

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.

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 Bank
{
	String customer;
	String accno;
	Scanner s=new Scanner(System.in);
	void get()
	 {
		System.out.println("Enter the customer name:");
		customer=s.next();
		System.out.println("Enter the account number:");
		accno=s.next();
	 }
```

```
}
class Cur_acct extends Bank
{
        double bal=0;
        double dep;
        void deposit1()
        {
                System.out.println("Enter the amount to be deposited");
                dep=s.nextInt();
                bal += dep;
                System.out.println("Amount"+dep+" is succesfully deposited");
        }
        void issue_cheque()
        {
                System.out.println("The cheque book is issued succesfully");
        }
        void check()
        {
              if(bal<1000)
             {
                System.out.println("The minimum amount must be 1000");
                bal=bal-5;
                System.out.println("Service charges are imposed");
              }
         }
        void display()
        {
                System.out.println("Balance = "+bal);
        }
}
```

```
class Sav_acct extends Bank
{
	double bal=0;
	double dep , draw;
	int rate=6;

	void deposit2()
	{
		System.out.println("Enter the amount to be deposited");
		dep=s.nextInt();
		bal  += dep;
		System.out.println("Amount"+dep+" is succesfully deposited");
	}
	void withdrawal()
	{
		System.out.println("Enter the amount to withdraw");
		draw=s.nextInt();
		bal -= draw;
		System.out.println("The balance amount is"+bal);
	}
	void comp_interest()
	{
		bal=bal+bal*0.06;
		System.out.println("The balance amount is"+bal);
	}
}

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

```
Scanner sc=new Scanner(System.in);
int ch, type;
Bank b=new Bank();
b.get();
Sav_acct a=new Sav_acct();
Cur_acct c=new Cur_acct();
System.out.println("Enter the account type (1.savings/2.current):");
type=sc.nextInt();
if(type==1)
{
       do
      {
          System.out.println("-----Menu-----");
          System.out.println("1:Deposit  2:Withdrawal  3.Compound interest  4.Exit");
          System.out.println("Enter your choice:");
          ch=sc.nextInt();
          switch(ch)
          {
                    case 1:
                              a.deposit2();
                              break;
                    case 2:
                              a.withdrawal();
                              break;
                    case 3:
                              a.comp_interest();
                              break;
          }
       } while(ch!=4);
          }
else
```

```
        {
            do
            {
              System.out.println("-----Menu-----");
              System.out.println("1:Deposit  2:Cheque issue  3.Display  4.Exit");
              System.out.println("Enter your choice:");
              ch=sc.nextInt();
              switch(ch)
              {
                    case 1:
                          c.deposit1();
                          break;
                    case 2:
                          c.issue_cheque();
                          break;
                    case 3:
                          c.check();
                          c.display();
                          break;
              }
             }while(ch!=4);
        }
    }
}
```

LAB 6

Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. The class internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of 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
package CIE;
import java.util.Scanner;
public class student
{
	protected String usn = new String();
	protected String name = new String();
	protected int sem;
	public void inputDetails()
	{
		Scanner s=new Scanner(System.in);
		System.out.println("Enter student usn:");
		usn = s.next();
		System.out.println("Enter student name:");
		name = s.next();
		System.out.println("Enter student sem:");
		sem = s.nextInt();
	}

	public void displayDetails()
	{
		System.out.println("Student Details:\n");
		System.out.println("USN: "+usn);
```

```
                System.out.println("Name: "+name);
                System.out.println("Sem: "+sem);
        }
}

public class internals extends student
{
        protected int marks[] = new int[5];
        public void CIEmarks()
        {
                Scanner s= new Scanner(System.in);
                for(int i=0;i<5;i++)
                {
                        System.out.println("Subject "+(i+1)+" marks");
                        marks[i] = s.nextInt();
                }
        }
}

// Package SEE
package SEE;
import CIE.internals;
import java.util.Scanner;
public class externals extends internals
{
        protected int marks[];
        protected int finalMarks[];
        public externals()
        {
                marks = new int[5];
                finalMarks = new int[5];
```

```
	}
	public void SEEMarks()
	{
		Scanner s= new Scanner(System.in);
		for(int i=0;i<5;i++)
		{
			System.out.println("Subject "+(i+1)+" marks:");
			marks[i] = s.nextInt();
		}
	}

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

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

// Class Main
import SEE.externals;
class Main
{
	public static void main(String args[])
	{
		int num = 2;
```

```
                externals finalMarks[] = new externals[num];
                for(int i=0;i<num;i++)
                {
                        finalMarks[i] = new externals();
                        finalMarks[i].inputDetails();
                        System.out.println("Enter CIE marks:");
                        finalMarks[i].CIEmarks();
                        System.out.println("Enter SEE marks:");
                        finalMarks[i].SEEMarks();
                }
                System.out.println("Displaying data:\n");
                for (int i=0;i<num;i++)
                {
                        finalMarks[i].calMarks();
                        finalMarks[i].displayDetails();
                        finalMarks[i].displayFinalMarks();
                }
        }
}
```

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<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.*;
class wrongAge extends Exception
{
        wrongAge(String s)
        {
                super(s);
        }

}

class input
{
        int f_age;
        int s_age;
        Scanner sc= new Scanner(System.in);



}
class father extends input
{
        father() throws wrongAge
```

```
	{
		System.out.println("Enter father's age:");
		f_age = sc.nextInt();
		if(f_age < 0)
			throw new wrongAge("Age cannot be negative");
	}
	void display1()
	{
		System.out.println("Father's age is : "+f_age);
	}
}
class son extends father
{
	son() throws wrongAge
	{
		System.out.println("Enter son's age:");
		s_age = sc.nextInt();
		if (s_age > f_age)
		{
			throw new wrongAge("Son's age cannot be greater than father's age");
		}
		else if (s_age <0)
			throw new wrongAge("Age cannot be negative");
	}
	void display2()
	{
		System.out.println("Son's age is : "+s_age);
	}
}

class Main
```

```
{
public static void main (String args[]){
        try
        {
                son s =new son();
                s.display1();
                s.display2();
        }
        catch(wrongAge e)
        {
                System.out.println("Error : "+e);
        }
}
}
```

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.

```
import java.util.*;
class NewThread implements Runnable
{
String name;
Thread t;
NewThread(String threadname)
{
        name = threadname;
        t = new Thread(this , name);
        t.start();
}

public void run()
{
        try {
        while(true)
        {       System.out.println(name);
                Thread.sleep(2000);
        } }
        catch (InterruptedException e) {
        System.out.println(name + " Interrupted");
        }
}
}

class MainThread
{
```

```
public static void main(String args[])
{
        new NewThread("Computer Science Engineering");
        try
        {
                while(true){
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000);
                Syste m.out.print("\n");
        }
        }
        catch (InterruptedException e) {
        System.out.println("BMS Interrupted");
        }

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

}

}
```

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

LAB 10

Demonstrate Inter process Communication and deadlock

```
// InterProcess Communication
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<5) {
        q.put(i++);
        }
        }
}

class Consumer implements Runnable {
        Q q;
        Consumer(Q q) {
        this.q = q;
        new Thread(this, "Consumer").start();
        }
        public void run() {
        int i=0;
        while(i<5) {
```

```
        int r=q.get();
        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.");
        }
}
// Dedlock
```