B.M.S COLLEGE OF ENGINEERING BENGALURU

Autonomous Institute, Affiliated to VTU

LAB REPORT

23CS3PCOOJ

Submitted in partial fulfilment of the requirements for Lab

Bachelor of Engineering

in

Computer Science and Engineering

Submitted by:

Roshni P (1BM22CS223)

Department of Computer Science and Engineering, B.M.S

College of Engineering,

Bull Temple Road, Basavanagudi, Bangalore, 560 019 2023-2024.

INDEX

Sl.No.	Title	Date
1	Complete scanned Observation Book	12/12/2023 - 20/02/2024
2	Lab 1	12/12/2023
3	Lab 2	19/12/2023
4	Lab 3	26/12/2023
5	Lab 4	02/01/2024
6	Lab 5	09/01/2024
7	Lab 6	16/01/2024
8	Lab 7	23/01/2024
9	Lab 8	30/01/2024
10	Lab 9	06/02/2024
11	Lab 10	20/02/2024

LAB 1

PROGRAM - 1

```
import java.util.*;
class demo
{
public static void main (String args[])
{
 System.out.println ("Hello World");
}
}
```

Output :-

Hello World

PROGRAM - 2

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

Output:-

java RectangleArea 10 5
length of rectangle = 10
breadth of rectangle = 5
area of rectangle = 50

PROGRAM - 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("Root 1 = "+r1+" Root 2 = "+r2);
}
    else if(d<0)
{
    System.out.println("Roots are imaginary");
    r1 = (-b)/(2*a);
    r2 = Math.sqrt(-d)/(2*a);
    System.out.println("Root 1 = "+r1+" + i "+r2);
    System.out.println("Root 1 = "+r1+" - i "+r2);
}
}
}
```

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

Output :-

Enter coefficients of a, b, c

4 5 6

Roots are imaginary

Root 1 = 0.0 + i1.0583687216470449

Root 2 = 0.0 - i1.0583687216470449

Enter coefficients of a, b, c

1 -2 1

Roots are real and equal

Root 1 = Root 2 = 1.0

Enter coefficients of a, b, c

1 -3 2

Roots are real and distinct

Root 1 = 2.0 Root 2 = 1.0

PROGRAM 3

```
class factorial {

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

Output:

```
Enter a number
3
The factorial is 6
```

store
67 19/12/23

LAB 2

PROGRAM 8 :

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 subjectMarks;
 int credits;
 char grade;

  Subject()
  {
   this.subjectMarks = 0;
   this.credits = 0;
   this.grade = 'n';
  }
}
 class student
 {
   String name;
   String usn;
   double SGPA;
   Scanner s;
   Subject[] subject;

   student()
   {
```

```
int i;
subject = new Subject [9];
for (i=0; i<9; i++)
{
  subject [i] = new subject ();
}
s= new Scanner (System.in );
}
void get Student Details ()
{
System.out.print (" Roshni P  1BM22CS223 ");
System.out.print ("Enter Student Name : ");
  name = s.nextLine ();
  System.out.print (" Enter Student USN: ");
  usn = s.nextLine ();
}
void getMarks() {
for (int i= 0; i<8; i++)
{
System.out.println ("Enter marks and credits
                    for subject " + (i+1)+":");

System.out.print (" Marks: ");
int marks = s.nextInt ();
System.out.print (" Credits: " );
int credit = s.nextInt ();
 subject [i].subjectMarks = marks;
subject [i].credits = credit;

if (marks >= 90)
{
  subject [i].grade = 'S';
}
```

```
else if (marks >= 80)
{
 subject [i]. grade='A';
}
else if (marks >=70
 {
    subject [i]. grade = 'B';
 }
 else if (marks >= 60)
 {
  subject [i]. grade = 'c';
 }
 else if (marks >= 50)
 {
  Subject [i]. grade = 'D';
 }
 else if (marks > - 40)
   {
     subject [i]. grade = 'E';
   }
 else
  {
    subject [i]. grade = 'F';
  }
}
}
Void compute SGPA ()
 {
   double totalCredits = 0;
   double totalGradeGradePoints = 0;
   for (int i= 0; i < 8; i++)
   {
```

```
totalCredits += subject[i].credits;
switch (subject[i].grade)
{
  case 'S':
     totalGradePoints += 10 * subject[i].credits;
     break;
     case 'A':
        totalGradePoints += 9 * subject[i].credits;
        break;
  case 'B':
        totalGradePoints += 8 * subject[i].credits;
        break;
  case 'C':
        totalGradePoints += 7 * subject[i].credits;
        break;
  case 'D'
     totalGradePoints += 6 * subject[i].credits;
     break;
     case 'E':
     totalGradePoints += 5 * subject[i].credits;
     break;
     default:
        totalGradePoints += 0;
        break;
  }
}
SGPA = totalGradePoints / totalCredits;
}

void displayResult()
{
```

```
System.out.println("\n Student Name : "+name);
System.out.println("Student USN : "+usn);
System.out.println("SGPA : "+SGPA);
}
}
public class Sgpa
public static void main (String [] args)
{
    Student s1 = new Student();
    s1.getStudentDetails();
    s1.getMarks();
    s1.computeSGPA();
    s1.displayResult();
}
}
```

Output :-

```
Enter your name :
Roshni
Enter your usn :
1BM22CS223
Enter the marks and credits for course 0:
Marks : 90
Credits : 4
Enter the marks and credits for course 1:
Marks :
91
Credits :
4
Enter the marks and credits for course 2
Marks :
99
```

credits : 3

Enter the marks & credits for course 3:
Marks: 91
credits: 3
Enter the marks & credits for course 4:
Marks: 93
Credit : 2

The sgpa is : 10.000000

store
67

PROGRAM 1

```
Class Modulus {
public static void main (string args [])
{
  int x = 42;
  double y = 42.25;
  byte a = 64, b, b1;
  int i;
  i = a<<2;
  b1 = (a<<2);
  b = (byte) (a<<2);
  i = a<<2;
  System.out.println("Roshni P 1BM22CS223");
  System.out.println("Original value of a "+a);
  System.out.println("i and b: "+i+" "+b);
  int a1 = -1;
  a1 = a1 >>24;
  system.out.println("value of a1: "+a1);

  a1 = -1
  a1 = a1 >>>24;
  system.out.println("value of a1: "+ a1);
}
}
```

Output:

Roshni P, 1BM22CS223
Original value of a: 64
i and b: 256 0
value of a1: -1
value of a1: 255

19.12.22

LAB 3 26/12/2023

PROGRAM 3:

```
int import java.util.Scanner;
  class Book {
  String name;
  String author;
  int price;
  int numPages;
  public 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 bookDetails = "Book name: "+ this.name + "\n"
+ "Author name : " + this.author = "\n"
+ "price : "+ this.price + "\n"
+ "Number of pages : " + this.numPages + "\n";
return bookDetails;
}
}
class Main {
public static void main (String[] args){
Scanner scanner = new Scanner (System.in);
System.out.print ("enter the number of books: ");
int n = scanner.nextInt();
Book []books = new Book [n];
for (int i=0; i<n; i++) {
System.out.print ("enter name of the book: ");
String name = scanner.next();
```

```
System.out.print("enter author of the book: ");
String author = scanner.next();

system.out.print("enter price of the book: ");
int numPages : scanner.nextInt();

System.out.print("enter the number of pages: ");
int numPages = scanner.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("Book" + (i+1) + ":\n" + books[i]);
}
}
}
```

Output:

```
Enter the number of books: 2
Enter name of the book : HarryPotter
Enter author of the book: JKRowling
Enter the price of the book: 500
Enter the number of pages of the book: 600
Enter name of the book: JungleBook
Enter author of the book: RuskinBond
Enter the price of the book: 200
Enter the number of pages of the book: 300
```

Books Details:

Book 1:
Book name: HarryPotter
Author name: JK Rowling
price: 500
Number of pages: 600

Book 2
Book name: JungleBook
Author name: RuskinBond
price: 200
Number of pages: 300

PROGRAM 1:

```
class OverloadDemo {
void test() {
System.out.println("no parameters");
}
void test(int a) {
System.out.println("a:" +a);
}
void test(int a, int b)
{
System.out.println("a and b:"+a+" "+b);
}
double test.(double a) {
System.out.println("double a:" +a);
return a*a;
}
}
```

```
class Overload {
public static void main (String args[]){
OverloadDemo ob= new OverloadDemo ();
double result;
ob.test ();
ob. test (10);
ob.test (10, 20);
result = ob. test (123.2);
System.out.println (" ob.test (123.2):" + result);
}
}
```

Output:

```
no parameters
a: 10
a and b : 10 20
double a: 123.2
ob.test (123.2) : 15178.040000000002
```

26.12

LAB 4

02/01/2023

PROGRAM 1

```
import java.util.Scanner;
class Inputscanner {
int d1, d2;
Scanner sc = new Scanner (System.in);
Input Scanner () {
if (this.getClass() == circle.class) {
}
System.out.println ("Enter d1 : ");
d1 = sc.nextInt ();
}
else {
System.out.println ("Enter d1 and d2 : ");
d1 = sc.nextInt ();
d2 = sc.nextint ();
}
}
}
abstract class Shape extends InputScanner {
abstract void printArea ();
}

class Rectangle extends Shape {
void print Area () {
System.out.println (" Area of triangle is :" + (double)
                    (d1 * d2)/2 );
}
}

class Rectangle extends Shape {
void print Area ()
System.out.println (" Area of rectangle is " + (double)
                    (d1 * d2));
```

```
}
}
class Circle extends Shape {
void printArea(){
System.out.println("Area of circle:" +(double)(3.14*d1*d1));
}
}
}

class AreaMain{
public static void main (String args[]){
Rectangle r = new Rectangle();
Triangle tr = new Triangle();
Circle c = new Circle();
r.printArea();
tr.printArea();
c.printArea();
}
}
```

Output

```
Enter d1 and d2:
5
4
Enter d1 and d2:
5
3
Enter d1:
6
```

Area of rectangle is : 20.0
Area of triangle is : 7.5
Area of circle : 113.03999999999999

2/01/24

store
67 /

LAB 5

```
Import java.util.*;

class Bank
{
 String name;
 int accno;
 boolean current;
 Scanner SC = new Scanner (System.in);
 Bank ()
 {
   if (this.getClass () == CurrentAcc.class)
   {
     current = true;
   }
   else
   {
     current = false;
   }
   System.out.print ("Enter name :");
    acc_no = sc.nextInt();
    System.out.print ("Enter account no.: ");
    acc_no = sc.nextInt ();
 }

 Void deposit () {
    System.out.print ("Enter deposit amount :");
    balance += Sc.nextDouble ();
 }
 Void withdraw () {
 System.out.print ("Enter withdraw amount:");
 double withdraw = sc.nextDouble();
```

```
while (withdraw > balance) {
    system.out.print("withdraw amount
        greater than balance, enter new amount:");
}
balance -= withdraw;
if (current ff balance < min_balance){
system.out.println("Below min balance
100, removing remaining money in account");
    balance = 0;
}
}
void withdraw (double withdraw) {
if (withdraw > balance) {
    system.out.println("withdraw amount
        greater than balance");
}
if (current ff balance < min_balance)
{
System.out.println("Below min balace of
        100, removing remaining
        money in account");
    balance = 0;
}
}
void showBalance() {
    system.out.print("balance = "+balance);
}
}
class currentAcc extends Account {
void cheque() {
    System.out.print("Enter cheque amount");
    double cheque = sc.nextDouble();
    withdraw(cheque);
```

```
System.out.println("cheque created...");
}
}
class savingsAcc extends Account {
void compound(int t, int r)
{
balance = balance*(Math.pow(1+((double)r/100)),t));
System.out.print("Balance after given rate and time = "+balance);
}
}
```

Output :

```
Enter name : john
Enter account no : 1
Enter name : smith
Enter account no : 2

--- Menu ---

Deposit
withdraw
compute interest for savings Acc
Display account details
Create cheque
Exit
choice :
1
```

Enter account no: 1
Enter deposit amount: 100
Enter account no: 2

Menu - - -

1] Deposit
2] Withdraw
3] Compute interest for savings Acc
4] Display account details
5] Create cheque
6] Exit

29.01.24

store
67 16/1/24/

LAB - 6 String Function

Output :-

String length : 4
concatenated string : somebmscce
converted Integer to string : 55 -->
11 5 111 109 101 BMSCE equal BMSCE : true
BMSCE equals same : false
BMSCE equalsIC bmscc : true

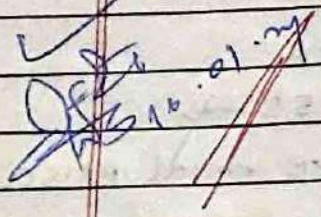
String created with char ~~array~~ array
String created with literal
string created with string

55 - same
String length = 4
Concatenated string : Somebmscc
Converting Integer to string : 55 --> 55
11 5 111 109 101
BMSCE equals BMSCE : true
BMSCE equals some : false
BMSCE equalsIC bmscc : true
startwith ('welcome') : true
end with ("college") : true
Reference equal b/w s1 and s5 (--) : false
Value equal b/w s1 and s5 (equals(1)) : true
get char() successful; Result - "BMSCE"

apple ball ..... yatchzee
1 2 3 4 5 6 7 8 9 10
Initial string : This is a test. This is too
Find string : Thwas was a test. Thwas was too
Concatenated string, helloworld

14] Find string : commagr

15] Find string : Hello Friends



store
67 23/1/2024

LAB 7

Student.java

```
package cie;
public class Students{
    public string name
    public int sem;
}
```

Internal.java

```
package cie;
import java.util.scanner;
public class Internals extend student;
    public int marks[] = new int [5];
    public void InputMarks()
{
    Scanner sc = new Scanner (system.in);
    for (int i=0; i<5; i++) {
        system.out.println("Enter subject" + (i+1) +
                           "Marks :");
        marks[i] = sc.nextInt();
}
}
public void displayMarks()
{
    for (int i=0; i<5; i++) {
    System.out.println("subject" + (i+1) + "marks:"
                       + Marks[i]);
}
}
```

External.java

```
package see;
import cie.student
import java.util.scanner;
public class External extends Student{
    public int marks[] = new int [5];
    public void inputMarks (){
        Scanner sc = new Scanner (System.in);
        for (int i = 0 ; i< 5; i++) {
        System.out.println ("Enter subject+
                                (i+1)+"marks:");
        marks[i] = sc.nextInt();
    }
}
public void displaymarks () {
for (int i=0; i<5; i++) {
    System.out.println("Subject" + "marks:"
                        +marks[i]);
}
}
}
```

Main.java

```
import cie.Student;
import cie.Internals;
import see.Externals;
import java.util.Scanner;
class Main {
    public static void main (String args[]) {
    int no = 2;
Externals finalmarks[] = new Externals[no];
```

```
Internals intmarks [] = new Internals [no];
for (int i=0 ; i<no ; i++) {
  finalmarks [i] = new External()
  int marks [i]. inputMarks ();
  Intmarks [i] . inputMarks ();
}
for (int i=0; i<no ; i++){
    system.out. println ("CIE:");
    Intmarks [i]. display marks ();
    System.out. println ("SEE");
    finalmarks [i]. displayMarks ();
}
}
}
```

Output -

Enter subject 1 marks : 30
Enter subject 2 marks : 50
Enter subject 3 marks : 40
Enter subject 4 marks : 20
Enter subject 5 marks : 10

Enter subject 1 marks : 30
Enter subject 2 marks : 90
Enter subject 3 marks : 60
Enter subject 4 marks : 80
Enter subject 5 marks : 90
Enter subject 1 marks : 70
Enter subject 2 marks : 60
Enter subject 3 marks : 20
Enter subject 4 marks : 80
Enter subject 5 marks : 10

CIE:

Subject 1 marks : 70
Subject 2 marks : 50
Subject 3 marks : 40
subject 4 marks : 20
subject 5 marks : 10

SEE -

Subject 1 marks : 30
subject 2 marks : 70
subject 3 marks : 60
Subject 4 marks : 80
Subject 5 marks : 90

CIE -

subject 1 marks : 70
subject 2 marks : 40
Subject 3 marks : 20
subject 4 marks : 20
subject 5 marks : 10

SEE -

subject 1 marks : 80
subject 2 marks : 60
Subject 3 marks : 30
Subject 4 marks : 20
Subject 5 marks : 30

23.01.24

store
67 30/01/24

LAB 8

```
import java.util.Scanner;
class WrongAge extends RuntimeException {
Public WrongAge() {
 super("Age cannot be negative");
}
publicWrongAge(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() {
System.out.println("Enter father's age:");
fatherAge = super.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() {
 super();
 System.out.println("Enter son's age: ");
 sonAge = super.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 InheritanceException {
  public static void main(String[] args) {
  try {
      Son son = new Son();
      son.display();
  } catch (WrongAge e) {
     System.out.println("Exception: " + e.getMessage());
  }
 }
}
```

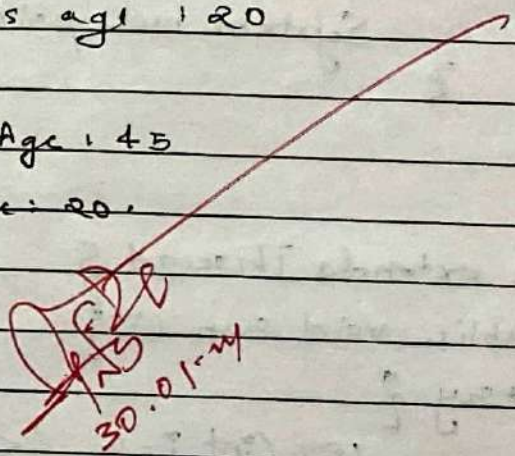
Output

Enter father's age : 20
Enter Son's age : 40
son's age cannot be greater than father's age.

Enter father's age : -10
Age cannot be negative

Enter father's age : 45
Enter son's age : 20

Father's Age : 45
son's age : 20.



LAB 8

```
import java.io.*;
class B extends Thread {
public void run () {
try {
    for (int i = 0 ; i < 3; i++) {
        System.out.println ("BMS");
        Thread.sleep (10000);
    } catch (InterruptedException e){
        System.out.println (e);
    }
}
}

class C extends Thread {
    public void run () {
    try {
        for (int i=0; i<3; i++) {
            system.out.println("CSE");
            Thread.sleep (2000);
        }
    } catch (Interrupted Exception e) {
        System.out.println (e);
    }
}
}

class ThreadMain {
    public static void main (String args []){
        B b = new B();
        C c = new C();
        b.start();
        c.start();
    }
}
```

store
67 /

Output :-

BMS
CSE
CSE
CSE
BMS
BMS

lab 9

Deadlock.java :

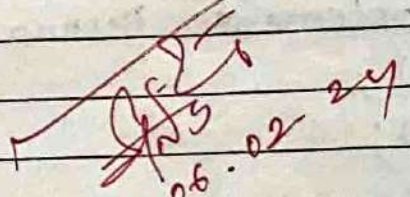
```
class A {
  synchronized void foo (B b) {
    String name = Thread.currentThread().
                                getName();
    System.out.println (name + " entered A.foo ");
    try {
    Tread.sleep(1000);
    }
      catch (Exception e) {
        System.out.println ("A Interrupted");
    }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

```
void last (){
    System.out.println ("Inside A.last");
}
}
class Deadlock implements Runnable {
A a = new A();
B b = new B();

Deadlock() {
Thread.currentThread().setName("Main Thread");
Thread t = new Thread(this, "Racing Thread");
t.start();
a.foo(b);
System.out.println("Back in main thread");
}
public void run() {
    b.bar(a);
    System.out.println("Back in other thread");
}
public static void main(String args[]) {
    new Deadlock();
}
}
```

Output :-

```
MainThread entered A.foo
RacingThread entered B.bar
MainThread trying to call B.last()
Inside A.last
```

Back in main thread
RacingThread trying to call A.last()
Inside A.last
Back in other thread.



Procon.java

```
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(e);
        }
    System.out.println("Got: " + n);
    valueSet = false;
    System.out.println("Tell producer");
    notify();
    }
    return n;
}
```

```
synchronized void put (int n) {
  while (value set) {
  try {
        System.out.println ("Producer waiting");
        wait ();
      }
      catch (InterruptedException e) {
        System.out.println (e);
      }
      this.n = n;
      value set = true;
      System.out.println ("Put : '+n');
      System.out.println ("Tell consumer");
      notify ();
    }
  }
class Producer implements Runnable {
  Qq;
  Producer (Qq) {
      this.q = q;
      new Thread (this, "Producer").
                                    start();
  }
  public void run () {
      int i = 0
      while (i<3) {
      q.put (i++);
    }
  }
class consumer implements Runnable {
  Qq;
  consumer (Q q) {
```

```
        this.q = q;
        new Thread(this, "consumer").start();
    }
    public void run() {
        int i = 0;
        while (i < 3) {
            int r = q.get();
            System.out.println("Consumed: " + r);
            i++;
        }
    }
}
class Procon {
    public static void main(String args[]) {
        Q q = new Q();
        new Producer(q); new Consumer(q);
```

Output

```
Put : 1
Got : 1
Put : 2
Got : 2
Put : 3
Got : 3
```

store
67

LAB 9

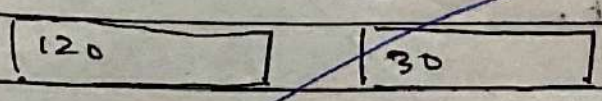
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

class SwingDemo {
SwingDemo () {
  JFrame jframe = 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 JTextFeld (8);
  JTextField bjtf = new JLabel TextField (8);

  JButton button = new JButton ("Calculate");
  JLabel err = new JLabel ();
  JLabel alab = new JLabel ();
  JLabel blab = new JLabel ();
  JLabel anslab = new JLabel ();

  jfrm.add (err);
  jfrm.add (jlab);
  jfrm.add (ajtf);
  jfrm.add (bjtf);
  jfrm.add (button);
```

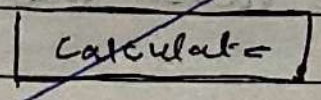
```
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);

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

ajtf.addActionListener(l);
bjtf.addActionListener(l);

button.addActionListener(new ActionListener
                                    () {
public void actionPerformed(ActionEvent evt) {

try {
int a = Integer.parseInt(ajtf.getText());
int b = Integer.parseInt(bjtf.getText());
int ans = a/b;

alab.setText("\nA = "+a);
blab.setText("\nB = "+b);
anslab.setText("\nAns = "+ans);
}
catch(NumberFormatException e) {
alab.setText("");
blab.setText("");
anslab.setText("");
```

```
        err.setText ("Enter Only Integers!");
    }
    catch (ArithmeticException e1) {
        alab.setText("");
        blab.setText("");
        anslab.setText("");
        err.setText ("B should be NoN zero!");
    }
    }
});
    jfrm.setVisible (true);
}
public static void main(String args[]){
SwingUtilities.invokeLater(new Runnable
                                    (){
    public void run (){
        new SwingDemo ();
    }
});
}
}
```

Output :

Enter the divider and dividend :





A = 120 B = 30 Ans = 4

* imports: *

- * javax.swing.*: Imports all classes from the Swing toolkit, used for creating graphical user interfaces.
- * java.awt.*: Imports all classes from the Abstract Window Toolkit (AWT), providing basic GUI components and event handling.
- * java.awt.event.*: Imports classes for handling events like button clicks and text field changes.

* classes: *

- * SwingDemo: * The main class defining the application's logic.
- * JFrame: * A top-level window container for Swing components.
- * JLabel: * A non-editable text label to display information.
- * JTextField: * A single-line text field for user input.
- * JButton: * A clickable button that triggers actions

* FlowLayout: * A layout manager that arranges components in a horizontal flow.

- ActionListener: * An interface for handling action events (like button clicks).

- * ActionEvent: * An event object representing an action.

- * SwingUtilities: * A utility class for working with Swing components on the event Dispatching Thread (EDT).

* Key Functions and Objects: *

- * JFrame: *

  - * new JFrame ("Divider App"): * Creates a new JFrame with the specified title.

  - * setSize (275, 150): * sets the initial size of the frame.

  - * setLayout (new FlowLayout ()): * Sets the layout manager to FlowLayout.

* setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE): * terminates the program when the frame is closed.

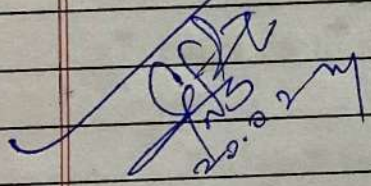
- *setVisible(true);* Makes the frame visible.

. * JLabel: *

- *new JLabel("Enter the divider and dividend:"); * creates a label with the given text.

- * JButton: *

- *new JButton("calculate"); *
creates a button with the given text.



LAB 1

Develop a Java program that prints all real solutions to the quadratic equation ax2 +bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminate b2 -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 sc = new Scanner(System.in);
		System.out.println("Enter coeffs a, b, c: ");
		a = sc.nextInt(); b = sc.nextInt(); c = sc.nextInt();


	}


	void compute(){
		while(a == 0){
			System.out.println("Invalid coeff, enter new one: ");
			Scanner sc = new Scanner(System.in);
			a = sc.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("r1 = r2 = " + r1);
		}
		else if (d > 0){
			r1 = (-b + Math.sqrt(d))/(2*a);
			r2 = (-b - Math.sqrt(d))/(2*a);
```

```
                System.out.println("Roots are real and distinct");
                System.out.println("r1 = " + r1 + "; " + "r2 = " + r2);
            }
            else if (d < 0){
                r1 = -b/(2*a);
                r2 = Math.sqrt(-d)/(2*a);
                System.out.println("Roots are imaginary");
                System.out.println("r1 = " + r1 + " + i" + r2 + "r2 = " + 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 SubjectMarks; int credits; int grade;
}


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

	Student(){
		subject = new Subject[10];
		sc = new Scanner(System.in);
		System.out.println("Enter no.  of subjects: ");
		n = sc.nextInt();
		for(int i = 0; i < 9; i++){
			subject[i] = new Subject();
		}
		sc.nextLine();

	}

	void getStudentDetails(){

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

```
            name = sc.nextLine();
            System.out.println("Enter usn: ");
            usn = sc.nextLine();
        }

        void getMarks(){
            System.out.println("\n");
            for(int i = 0; i < n; i++){
                System.out.println("Enter no. of credits: ");
                subject[i].credits = sc.nextInt();
                System.out.println("Enter marks obtained: ");
                subject[i].SubjectMarks = sc.nextInt();
                System.out.println("\n");
                if (subject[i].SubjectMarks > 100) subject[i].SubjectMarks = 100;
                else if (subject[i].SubjectMarks < 40) subject[i].SubjectMarks = 0;
                subject[i].grade = (subject[i].SubjectMarks / 10) + 1;
                if (subject[i].grade == 11) subject[i].grade = 10;
                if (subject[i].SubjectMarks >= 40 && subject[i].SubjectMarks < 50)
subject[i].grade = 4;
                else if (subject[i].SubjectMarks >= 50 && subject[i].SubjectMarks < 55)
subject[i].grade = 5;
                else if (subject[i].SubjectMarks >= 55 && subject[i].SubjectMarks < 60)
subject[i].grade = 6;

            }
        }

        double computeSGPA(){
            int effective = 0, credits = 0;
            for(int i = 0; i < n; i++){
                effective += (subject[i].grade * subject[i].credits);
                credits += subject[i].credits;
```

```
                }

                sgpa = effective/credits;
                return sgpa;

        }
}

class StudentMain{
        public static void main(String args[]){
                Student student = new Student();
                System.out.println("Pranav Y - 1BM22CS204");
                student.getStudentDetails();
                student.getMarks();
                System.out.println("Name of student is: " + student.name);
                System.out.println("USN of student is: " + student.usn);
                System.out.println("SGPA of student is: " + student.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 Book{
	String name, author;
	int price, no_pages;


	public Book(String name, String author, int price, int no_pages){
		this.name = name;
		this.author = author;
		this.price = price;
		this.no_pages = no_pages;
	}


	public String toString(){
		System.out.println("Name: " + this.name);
		System.out.println("Author: " + this.author);
		System.out.println("Price: " + this.price);
		System.out.println("Pages: " + this.no_pages);
		return this.name + this.author + this.price + this.no_pages;
	}
}


class BookMain{
	public static void main(String args[]){
		System.out.println("Roshni P - 1BM22CS223");
		Book books[] = new Book[10];
		Scanner sc = new Scanner(System.in);
```

```
                System.out.println("Enter no. of book objects: ");
                int n = sc.nextInt();
                sc.nextLine();
                for(int i = 0; i < n; i++){
                        String name, author;
                        int price, no_pages;
                        System.out.println("Enter name: ");
                        name = sc.next();
                        System.out.println("Enter author: ");
                        author = sc.next();
                        System.out.println("Enter price: ");
                        price = sc.nextInt();
                        System.out.println("Enter no. of pages: ");
                        no_pages = sc.nextInt();
                        books[i] = new Book(name, author, price, no_pages);
                }
                System.out.println("\n");
                for(int i = 0; i < n; i++){
                        System.out.println("Book " + (i+1) + " Details:\n");
                        books[i].toString();
                        System.out.println("\n");
                }

        }
}
```

LAB 4

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

```
import java.util.Scanner;


class InputScanner{
	int d1, d2;
	Scanner sc = new Scanner(System.in);
	InputScanner(){
		if(this.getClass() == Circle.class){
			System.out.println("Enter d1: ");
			d1 = sc.nextInt();
		}
		else{
			System.out.println("Enter d1 and d2: ");
			d1 = sc.nextInt();
			d2 = sc.nextInt();
		}
	}
}


abstract class Shape extends InputScanner{
	abstract void printArea();
}


class Triangle extends Shape{
	void printArea(){
		System.out.println("Area of triangle is: " + (double)(d1*d2)/2);
	}
}
```

```
class Rectangle extends Shape{
	void printArea(){
		System.out.println("Area of rectangle is: " + (double)(d1*d2));
	}
}



class Circle extends Shape{
	void printArea(){
		System.out.println("Area of circle: " + (double)(3.14*d1*d1));
	}
}

class AreaMain{
	public static void main(String args[]){
		System.out.println("Roshni P - 1BM22CS223");
		Rectangle r = new Rectangle();
		Triangle tr = new Triangle();
		Circle c = new Circle();
		r.printArea();
		tr.printArea();
		c.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;

import java.lang.*;


class Account {
	String name;
	int acc_no;
	boolean current;
	double balance = 0;
	int min_balance = 100;
	Scanner sc = new Scanner(System.in);


	Account(){
		if(this.getClass() == CurrentAcc.class){
			current = true;
		} else {
			current = false;
		}
		System.out.print("Enter name: ");
		name = sc.next();
		System.out.print("Enter account no.: ");
		acc_no = sc.nextInt();
	}
```

```
	void deposit() {
		System.out.print("Enter deposit amount: ");
		balance += sc.nextDouble();
	}

	void withdraw() {
		System.out.print("Enter withdraw amount: ");
		double withdraw = sc.nextDouble();
		while (withdraw > balance) {
			System.out.print("Withdraw amount greater than balance, enter new
amount: ");
			withdraw = sc.nextDouble();
		}
		balance -= withdraw;
		if (current && balance < min_balance) {
			System.out.println("Below min balance of 100, removing remaining money
in account");
			balance = 0;
		}
	}

	void withdraw(double withdraw) {
		if (withdraw > balance) {
			System.out.println("Withdraw amount greater than balance");
		}
		if (current && balance < min_balance) {
			System.out.println("Below min balance of 100, removing remaining money
in account");
			balance = 0;
		}
	}
```

```
	void showBalance() {
		System.out.print("balance = " + balance);
	}
}

class CurrentAcc extends Account {
	void cheque(){
		System.out.print("Enter cheque amount: ");
		double cheque = sc.nextDouble();
		withdraw(cheque);
		System.out.println("Cheque created...");

	}
}

class SavingsAcc extends Account {
	void compound(int t, int r) {
		balance = balance * (Math.pow((1 + ((double) r / 100)), t));
		System.out.print("Balance after given rate and time = " + balance);
	}
}

class Bank {
	public static void main(String args[]) {
		SavingsAcc john = new SavingsAcc();
		CurrentAcc smith = new CurrentAcc();
		Account ref = null;
		Scanner sc = new Scanner(System.in);
		int acc, choice;
		System.out.println("------MENU------\n");
		System.out.println(
```

```
                    "1.Deposit\n2.Withdraw\n3.Compute intrest for Savings
Acc\n4.Display account details\n5. Create cheque\n6.Exit\nChoice:");
            choice = sc.nextInt();
            System.out.println("Enter account no.: ");
            acc = sc.nextInt();
            if (acc == 1) {
                ref = john;
            } else {
                ref = smith;
            }
            while (choice != 6) {
                if (choice == 1) {
                    ref.deposit();
                } else if (choice == 2) {
                    ref.withdraw();
                } else if (choice == 3) {
                    if (acc == 1) {
                        john.compound(1, 5);
                    } else {
                        System.out.println("Not a savings account");
                    }
                } else if (choice == 4) {
                    ref.showBalance();
                } else if (choice == 5) {
                    if (acc == 2) {
                        smith.cheque();
                    } else {
                        System.out.println("Not a current account");
                    }
                }
                System.out.println("Enter account no.: ");
```

```
                    acc = sc.nextInt();
                    System.out.println("------MENU------\n");
                    System.out.println(
                                    "1.Deposit\n2.Withdraw\n3.Compute intrest for Savings
Acc\n4.Display account details\n5. Create cheque\n6.Exit\nChoice:");
                    choice = sc.nextInt();
                }

            }
}
```

Lab 6

Demonstrate the utilization of String and StringBuffer functions as well as the usage of abstract classes

```
abstract class Bird{
   abstract void fly();
   abstract void makeSound();
}
class Eagle extends Bird{
   void fly(){
      System.out.println("Eagle fly method");
   }
   void makeSound() {
      System.out.println("Eagle sound method");
   }
}
class Hawk extends Bird{
   void fly() {
      System.out.println("Hawk fly method");
   }
   void makeSound() {
      System.out.println("Hawk sound method");
   }
}
class BirdMain {
   public static void main(String[] args) {
      Eagle e = new Eagle(); Hawk h = new Hawk();
      e.fly();h.fly();e.makeSound();h.makeSound();
   }
}


import java.util.Scanner;
```

```
import java.lang.Math;


class InputScanner{
        int d1, d2, d3;
        Scanner sc = new Scanner(System.in);
        InputScanner(){
                if(this.getClass() == Circle.class){
                        System.out.println("Enter d1: ");
                        d1 = sc.nextInt();
                }
                else{
                        System.out.println("Enter a, b, c: ");
                        d1 = sc.nextInt();
                        d2 = sc.nextInt();
      d3 = sc.nextInt();
                }
        }
}


abstract class Shape extends InputScanner{
        abstract void calculateArea();
   abstract void calculatePerimeter();
}


class Triangle extends Shape{
        void calculateArea(){
     double s = (d1+d2+d3)/2;
                System.out.println("Area of triangle is: " + (double)Math.sqrt(s*(s-d1)*(s-d2)*(s-d3)));
        }
   void calculatePerimeter(){
     System.out.println("Perimeter of triangle is: " + (double)(d1+d2+d3));
```

```
    }
}

class Circle extends Shape{
        void calculateArea(){
                System.out.println("Area of circle: " + (double)(3.14*d1*d1));
        }
    void calculatePerimeter(){
       System.out.println("Perimeter of circle: " + (double)(3.14*2*d1));
    }
}

class ShapeMain{
        public static void main(String args[]){
                System.out.println("Pranav Y - 1BM22CS204");
                Triangle tr = new Triangle();
                Circle c = new Circle();
                tr.calculateArea(); tr.calculatePerimeter();
                c.calculateArea(); tr.calculatePerimeter();
        }
}
import java.util.*;

class StringMain {
        public static void main(String args[]) {
                /* 1 */ char arr[] = { 'B', 'M', 'S', 'C', 'E' };
                String s1 = new String(arr);
                String s2 = new String("bmsce");
                String s3 = new String(s2);
                /* 2 */ String s4 = "some";
```

```
		System.out.println("String length: " + s4.length() + "\n" + "Concatenated string: " +
s4.concat(s2));
		/* 3 */ int d = 55;
		String sd = Integer.toString(d);
		System.out.println("Converting Integer to string: " + d + " -> " + sd);
		/* 4 */ char res[] = new char[20];
		String str = new String("Welcome to BMSCE College");
		str.getChars(10, 16, res, 0);
		/* 5 */ byte byte_arr[] = s4.getBytes();
		for (int i = 0; i < 4; i++) {
			System.out.print(byte_arr[i] + " ");
		}
		/* 6 */ System.out.println("BMSCE equals BMSCE: " + s1.equals("BMSCE"));
		System.out.println("BMSCE equals some: " + s1.equals(s4));
		System.out.println("BMSCE equalsIC Bmsce: " + s1.equalsIgnoreCase(s2));
		/* 7 */ System.out.println(str.regionMatches(11, "BMSCE College", 0, 11));
		/* 8 */ System.out.println(str.startsWith("Welcome"));
		/* 9 */ System.out.println(str.endsWith("College"));
		/* 10 */ String s5 = new String("BMSCE");
		System.out.println("Reference equal b/w s1 and s5 (==): " + s1 == s5);
		System.out.println("Value equal b/w s1 and s5 (equals()): " + s1.equals(s5));
		/* 11 */ String str_arr[] = { "van", "watch", "ball", "cat", "xmas", "yatch", " zee",
"apple", "ice", "jug",
				"kite", "lift", "man", "net", "orange", "dog", "ent", "free", "gun",
"hen", "parrot", "queen", "ring",
				"star", "tree", "umbrella" };
		for (int i = 0; i < str_arr.length; i++) {
			for (int j = i + 1; j < str_arr.length; j++) {
				if (str_arr[i].compareTo(str_arr[j]) > 0) {
					String temp;
					temp = str_arr[i];
					str_arr[i] = str_arr[j];
```

```
                                        str_arr[j] = temp;
                                }
                        }
                }
                for (int i = 0; i < str_arr.length; i++) {
                        System.out.print(str_arr[i] + " ");
                }
                /*11*/ String num_arr[] = {"1", "4", "3", "2", "5"};
                for (int i = 0; i < num_arr.length-1; i++) {
                        for (int j = i + 1; j < num_arr.length; j++) {
                                if (num_arr[i].compareTo(num_arr[j]) > 0) {
                                        String temp;
                                        temp = num_arr[i];
                                        num_arr[i] = num_arr[j];
                                        num_arr[j] = temp;
                                }
                        }
                }
                System.out.println("\n");
                for (int i = 0; i < num_arr.length; i++) {
                        System.out.print(num_arr[i] + " ");
                }

        }
}
```

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 see;


import cie.Student;

import java.util.Scanner;


public class Externals extends Student{

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


	public void inputMarks() {

		Scanner sc = new Scanner(System.in);

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

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

			marks[i] = sc.nextInt();

		}

	}


	public void displayMarks() {

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

			System.out.println("Subject " + (i + 1) + " marks: " + marks[i]);

		}

	}

}
```

```
package cie;


import java.util.Scanner;
```

```
public class Internals extends Student {
	public int marks[] = new int[5];

	public void inputMarks() {
		Scanner sc = new Scanner(System.in);
		for (int i = 0; i < 5; i++) {
			System.out.println("Enter subject " + (i + 1) + " marks: ");
			marks[i] = sc.nextInt();
		}
	}

	public void displayMarks() {
		for (int i = 0; i < 5; i++) {
			System.out.println("Subject " + (i + 1) + " marks: " + marks[i]);
		}
	}
}
import cie.Student;
import cie.Internals;
import see.Externals;
import java.util.Scanner;

class Main{
	public static void main(String args[]){
		int no = 2;
		Externals finalmarks[] = new Externals[no];
		Internals intmarks[] = new Internals[no];
		for (int i = 0; i < no; i++){
			finalmarks[i] = new Externals();
			intmarks[i] = new Internals();
			finalmarks[i].inputMarks();
```

```
                intmarks[i].inputMarks();
            }

            for(int i = 0; i < no; i++){
                System.out.println("CIE: ");
                intmarks[i].displayMarks();
                System.out.println("SEE: ");
                finalmarks[i].displayMarks();
            }
        }
}
package cie;

public class Student {
    public String name, usn;
    public int sem;

}
```

Lab 7

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

```
import java.util.Scanner;


class WrongAge extends RuntimeException {
public WrongAge() {
super("Age cannot be negative");
}

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() {
System.out.println("Enter father's age:");
fatherAge = super.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() {
super();
System.out.println("Enter son's age:");
sonAge = super.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 InheritanceException {
public static void main(String[] args) {
try {
Son son = new Son();
son.display();
} catch (WrongAge e) {
System.out.println("Exception: " + e.getMessage());
}
}
}
```

Lab 8

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

```
import java.io.*;


class B extends Thread{
	public void run(){
		try{
			for(int i = 0; i < 3; i++){
				System.out.println("BMS");
				Thread.sleep(10000);
			}
		} catch (InterruptedException e){
			System.out.println(e);
		}

	}
}

class C extends Thread{
	public void run(){
		try{
			for(int i = 0; i < 3; i++){
				System.out.println("CSE");
				Thread.sleep(2000);
			}
		} catch (InterruptedException e){
			System.out.println(e);
		}
```

```
        }
}

class ThreadMain{
        public static void main(String args[]){
                B b = new B();
                C c = new C();
                b.start();
                c.start();
        }
}
```

Lab 9

Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an 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 dividend:");


      // add text field for both numbers

      JTextField ajtf = new JTextField(8);

      JTextField bjtf = new JTextField(8);


      // calc button

      JButton button = new JButton("Calculate");


      // labels

      JLabel err = new JLabel();

      JLabel alab = new JLabel();
```

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

// add in order :)
jfrm.add(err); // to display error 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

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

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

   synchronized void put(int n) {
      while (valueSet)
         try {
            wait();
         } catch (InterruptedException e) {
            System.out.println("InterruptedException caught");
         }
      this.n = n;
      valueSet = true;
      System.out.println("Put: " + n);
      notify();
   }
```

```
}

class Producer implements Runnable {
   Q q;

   Producer(Q q) {
      this.q = q;
      new Thread(this, "Producer").start();
   }

   public void run() {
      int i = 0;
      while (i < 5) {
         q.put(i++);
      }
   }
}

class Consumer implements Runnable {
   Q q;

   Consumer(Q q) {
      this.q = q;
      new Thread(this, "Consumer").start();
   }

   public void run() {
      int i = 0;
      while (i < 5) {
         int r = q.get();
         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.");
    }
}
```