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:

SHREE VARNA M

1BM22CS263

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	20/02/2024
11	Lab 10	06/02/2024

SHREE VARNA. M
1BM22CS263

12/12/23

PAGE:
DATE: / / 01

Develop a Java program that prints all the real Solutions to the quadratic equation $ax^2 - bx + c$ read in a, b, c and use quadratic formula if the discriminant is negative Display a message stating that there are no real Sol^n.

```
import java.util.Scanner;
class Quadratic
{
  int a, b, c;
  double r1, r2, d;
  void getId()
  {
    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("Root1 =" + r1 +
                        "Root2 =" + r2);
}

else if(d<0)
{
    System.out.println("Roots are imagin
        -ary");
    r1 = (-b)/(2*a)
    r2 = Math.sqrt(-d)/(2*a)
    System.out.println("Root1 =" + r1 + "i"
                        + r2);
```

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

Class Qudratic Main
{
  public Static void main (String args[])
  {
    Qudratic q = new Quadratic();
      q.getd();
      q.Compute();
  }
}
```

Output :-

Enter the Coefficient of a, b, C
2
3
4
Roots are imaginary.
Root 1 = 0.0 + i 1.1989578808
Root 2 = 0.0 - i 1.1989578

Enter the coefficient of a, b, c
2
4
2
Roots are real & equal
Root 1 = Root 2 = 1

Enter the coefficient of a, b, c
2
8
2
Roots are real and distinct
Root 1 = +0.2679
Root 2 = 3.732

10
Rev 12/12/23

SHREE VARNA.M
1BM22CS263

PAGE: 5
DATE: 19/12/23

SHREE VARNAM Lab-2.
1BM22CS263.

1) Develop a Java program to create 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.

Note: SGPA = $\frac{\Sigma[(\text{Course credits})(\text{Grade points})]}{\Sigma[(\text{Course credits})]}$
[including F Grade]

CGPA = $\frac{\Sigma[(\text{Course credits})(\text{Grade points})]}{\Sigma[(\text{Course credits})]}$

```
import java.util.*;
Class Subject{
    int subjectMarks;
    int credits;
    int grade;
}

class Student{
    Subject subjid[];
    String name;
    String USN;
    double SGPA;
```

```
Scanner s;

Student ()
{
    int i;
    Subject = new Subject[3];
    for (i=0; i<3; i++)
    {
        Subject[i] = new Subject();
    }
    s = new Scanner(System.in);
}

}

Void getStudentDetails()
{
    System.Out.println("Enter the name");
    Ford name = s.next();
    System.out.println("Enter the USN");
    USN = s.next();

}

Void get Marks()
{
  for(i=0; i<3; i++)
  {
     System.Out.println("Enter the marks
```

```
for Subject " + (i+1) + ":");
Subject[i].SubjectMarks = S.nextInt();
System.out.println("Enter Credits for
  Subject " + (i+1) + ":");
Subject[i].credits = S.nextInt();
Subject[i].grade = Subject[i].Subject
                    Marks/10) + 1;

if (Subject[i].grade == 11)
    Subject[i].grade = 10;

if (Subject[i].grade <= 4)
    Subject[i].grade = 0;

  }
}

void ComputeSGPA()
{
  int Score=0;
  int totalcredit = 0;
  for(int i=0; i<9; i++)
  {
    Score += Subject[i].credits * Subject[i].grade;
    totalCredit += Subject[i].credits;
  }
 SGPAtotal = (double) Score / (double) totalcredit
}
```

```
class Main
{
    public static void main(String args[])
    {
        Student S1 = new Student();
        S1.getStudentDetails();
        S1.getMarks();
        S1.computeSGPA();

        System.out.println("Name :" +
                                    S1.name);
    System.out.println("USN" + S1.USN);

    System.out.println("SGPA :" + S1.SGPA);

    }
}
```

Output:

```
Enter your Name : Shra Verma
Enter the USN: 1BMSSCS263
Enter the marks of Subject 1 is 90
Enter the credits of Subject 1 is 4
Enter the marks of Subject 2 is 95
Enter the credits of Subject 2 is 3
Enter the marks of Subject 3 is 80
Enter the credits of Subject 3 is 3
```

PAGE: 9
DATE: / /

Name : Shru
USN : 1BM22CS263
SGPA : 9.875.

(10) 14/12/23

SHREE VARNA.M
1BM22CS263

Lab - 3

1) Creat a class book which contain 4 member name, author, price, numpages. Include a constructor to set the values for the members. Include methods to set and get the details of the object, Include a toString method that could display the complete details of the book. Develop a Java program to creat n book objects

Soln

```
import Java.util.*;
class Books{
    String name;
    String author;
    int price;
    int numpages;

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

public String toString()
```

```
{ String name, author, price, numpages;
  name = "Book name:" + this.name + "\n";
author = "Author name:" + this.author + "\n";
price = "Price:" + this.price + "\n";
numpages = "No of pages:" + this.numpages
                          + "\n";
 return name + author + price + numpages;
}

Class Books1
{
  public static void main(String args[])
  {
     Scanner s = new Scanner()
     int n,
     String name, author;
     int price, numpages;
    System.out.println("Enter No of Books")
     n = s.nextInt()
     Books b[];
     b[] b = new books[n]
     int i;

for(i=0; i<n; i++)
{
     Scanner sc = new Scanner();
     name = sc.next();
     author = sc.next();
     price = sc.nextInt();
     numpages = sc.nextInt();
```

```
b[i] = new books(name, author,
                 price, numpages);

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

→ Enter no of bks : 2
Hi
Hello
23
45
Sufi
Kala
95
82

SHREE VARNA.M
1BM22CS263

Books name : Hi
author name : Hello
price : 23
numpages : 45

Books name : Sufi
author name : Kala
price : 95
numpages : 45.

SHREE VARNA. M
1BM22CS263

PAGE: 13
DATE: 21/1/24

LAB-04

1) 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.

Area of Rectangle = l × b;
Area of triangle = $\frac{1}{2}$ × b × h;
Area of circle = 3.14 × r × r.

Soln

```
import java.util.Scanner;

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

abstract class Shape extends InputScanner()
{
    double a;
    double b;
    abstract void getInput();
    abstract void displayarea();
}
```

```
Class Rectangle extends Shape(){
    public class void getInput(){
        System.out.println("Enter the value
of a & b in rectangle:");
}
        a=S.nextInt();
        b=S.nextInt();}
Public void display area(){
    System.out.println("The area of
rectangle:"+a*b);

Class Triangle extends Shape(){
    public void getInput(){
        System.out.println("Enter the
value of a & b in triangle:");
        a=S.nextInt();
        b=S.nextInt();
}

Public void display area(){
    System.out.println("The area of
triangle"+(0.5*a*b));}
}
Class

Class circle extends Shape(){
    Public void getInput(){
        System.out.println("Enter the
```

```
        value of a of circle (radius): ");
            a = s.nextInt();
        }
        public void displayarea(){
            System.out.println("The area of
rectangle: " 3.14*a*a);
        }

}

Public class MainShape{

public Static void main(String[] args)
{
    rectangle i = new rectangle();
    triangle t = new triangle();
    circle c = new circle();
    i.getInput();
    i.displayarea();
    t.getInput();
    t.displayarea();
    c.getInput();
    c.displayarea();
}
```

⇒ Output

Enter the value of a, b of rectangle:
2 3
The area of rectangle: 60

Enter the value of a, b of triangle:
2 3
The area of triangle: 3.0

Enter the value of a of circle:
4
The area of circle = 50.24

02/01/24 SHREE VARNA.M
1BM22CS263

PAGE: 17
DATE: 09/01/24

LAB-5

1) 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 Saving 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 task:

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.

Solve:

```
import java.util.*;

class account{
    String customer_name;
    int acc_no;
    String type;
    double balance;

    account (String name, int no, String
        type, double balance)
    {
        customer_name = name;
        acc_no = no;
        this.type = type;
        this.balance = balance;
    }

void deposit (double amount)
{
    balance += amount;
}

void withdraw (double amount)
{
    if ((balance - amount) >0)
        balance -=amount
    else
        System.out.println("Insufficient
                balance");
```

```
void display(){
    System.out.println("Name:" +
    customer_name + "\n" + "account
    number:" + acc_no + "\n" + type
    + type + "\n" + "Balance:" + balance);
  }
}

class Sav_acc extends account{

    private static double rate = 2;
    Sav_acc(String name, int acc_no,
            double balance)
    {
      super(name, acc_no, "Savings", balance)
    }

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

class curr_acc extends account{
    curr_acc(String name, int acc_no,
            double balance)
    {
      super(name, acc_no, "current", balance
    }
```

```
private double minbal = 250;
private double servicecharge = 100;

void checkmin(){
    if (balance < minbal)
    {
      System.out.println("Insufficient
    balance so service charges are
    Subtracted from your balance
      amount");
    balance -= servicecharge;
    System.out.println("Balance is:"
                        +balance);
    }
}
}

public class bank{

    public static void main(String
    args[])
{
    Scanner S = new Scanner(System.in);
    System.out.println("Enter the
                        name:");
    String name = S.next();
    System.out.println("Enter the type
                (current/Saving)");
    String type = S.next();
    System.out.println("Enter the
```

```
account Number");
int acc_no = S.nextInt();
System.out.println("Enter the initial
    balance:");
double balance = S.nextDouble();
int ch;
double amount1, amount2;
account acc = new account(name, acc_no,
Sav_acc sa = type, balance);
Curr_acc ca =
Sav_acc sa = new Sav_acc(name, acc_no
                balance);
Curr_acc ca = new Curr_acc(name,
                acc_no, balance);

while(true)
{
  if(acc.type.equals("Savings"))
  {
    System.out.println("\n Menu \n
    1. Deposit \n 2. withdraw 3. Intrest
    4. Display");
    ch = S.nextInt();

    Switch(ch)
    {
      case 1: System.out.println("Enter
        the amount:");
        amount1 = S.nextInt()
        Sa.deposit(amount1);
        break;
```

```
Case 2 : System.out.println(" Enter
          the amount");
         amount2 = S.nextInt()
         Sa.withdraw(amount2);
         break;

Case 3 : Sa.interest()
         break;

Case 4 : Sa.display()
         break;
default : System.out.println
          (" Invalid input");
          System.exit(0);

    }
}

else
{
    System.out.println("\n menu\n
    1.Deposit 2.withdraw 3.Display");
    System.out.println(" Enter the
                        choice");
    ch = S.nextInt();
    Switch(ch)
    {
        Case 1 : System.out.println("Enter
                 the amount");
                 amount1 = S.nextInt();
```

```
        Ca.deposit(amount1);
        break;
    Case 2: System.out.println("Enter
                the amount:")
        amount2 = S.nextInt();
        Ca.withdraw(amount2);
        Ca.checkmin();
        break;

    Case 3: Ca.display();
        break;

    default: System.out.println("Invalid
                input");
            System.exit(0);

    }
}
}
```

⇒ OUTPUT:

```
Enter the name: Shree
Enter the type (current / savings):
Enter the account number: 5556
Enter the initial balance: 2000
```

Menu
1. Deposit 2. withdraw 3. display
Enter the choice
1
Enter the amount: 4000

Menu
1. Deposit 2. withdraw 3. display
Enter the choice
2
Enter the amount: 200

Menu
1. Deposit 2. withdraw 3. display
Enter the choice
3

Name: Shree
account No: 8556
Type: current
balance: 5800.0

SHREE VARNA.M
1BM22CS263

KTU-16

Generics

1) Write a java program with using generic shows the class Stack for 5 integer & 5 double

```
import java.util.Scanner;

Class Stack<E>
{
    E stck[];
    int top;
Final int Size = 10;

    Stack()
    {
        stck[] = (E[]) new Object [Size];
        top = -1;
    }

Void push (E n)
{
  if top == Size - 1)
  {
    System.out.println("overflow");
  }
  else
  {
    top++
    stck[ta] = n;
  }
}
```

```
E pop()
{
  if (top <0)
  {
    System.out.println("Underflow");
    Return null;
  }
  else
  {
    top--;
    Return Stck[top];
  }
}
}

Class IntStack
{
  public static void main(String[]
                           args)
  {
    Stack<Integer> mystack1 = new
        Stack<Integer>();
    Stack<double> mystack2 = new
        Stack<double>();
    Scanner s= new Scanner(System.in);
    System.out.println("Enter the element
        into Stack");
```

```
for(int i=0; i<5; i++)
{
    int n=s.nextInt();
    mystack1.push(n);
}

System.out.println("Enter Elements
    into the Double Stack");

for(int i=0; i<5; i++)
{
    double m = s.nextDouble();
    mystack2 = push(m);
}

System.out.println("Elment of Stack1");
for(int i=0; i<5; i++)
{
System.out.println(mystack1.pop());
}

System.out.println("Elment of Stack2")
for(int i=0; i<5; i++)
{
System.out.println(mystack2.pop());
}
s.close();
}
```

Output:

Enter the element into Stack
1
2
3
4
5
Enter the Element into doubl Stack
6
7
8
9
1

Elements of stack 1: 5
4
3
2
1

Element of Stack 2:
7.0
9.0
8.0
7.0
6.0

SHREE VARNA. M
1BM22CS263

1) Write a Java program to create an abstract class Shape with abstract methods calculateArea() & calculatePerimeter(). Create Subclass circle & triangle.

```
import java.util.*;

abstract class Shape
{
    double a;
    double b;
    double c;
    abstract void Area();
    abstract void perimeter();
}

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

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

```
Void calculate perimeter(){
System.out.println("perimeter"+
    (a+b+c));
}
class Circle calculate Shape{
    Circle(double r)
    {
        a=r;
    }

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

    Void calculate Perimeter()
    {
      System.out.println("perimeter="
          (2*Math.PI*a));
    }
}

Class Shapum{

    public static void main(String[]
                                args)
    {
      triangle t = new triangle(2.0,
                    3.0, 5.0)
```

```
Circle C = new Circle(50);
t.calculateArea();
t.calculatePerimeter();
C.calculateArea();
C.calculatePerimeter();
}

}
```

OUTPUT:

```
Area = 4.1578...
Perimeter = 110
Area = 78.539...
perimeter = 31.415...
```

SHREE VARNA.M
1BM22CS263

String

Demonstrate String length, String literal, String concat

```
public class String1
{
    public static void main(String args[]),
    {
      System.out.println("Demonstrating String length:")
      String a = "Hello";
      System.out.println(a.length());
      System.out.println("Str concat:")
      String age = "9";
      String msg = "He's " + age + " years old";
      System.out.println(msg);
      System.out.println("Demonstrat literal");
      System.out.println("abc".length());
    }
}
```

OUTPUT:

Demonstrating String length

5

Str concat:

He is 9 years old
Demonstrate literals
3

Use getChars() to extract Bmsce from "Welcome to BmsCe college")

```
Public class String2{

    public Static Void main (String args[])
{
    String S: "Welcome to BMSCE College";
    int Start = 10;
    int end = 16;
    char buf[] : new char [end-Start];
    S. get chars (Start, end, buf, 0);
    System out println (buf);
}
}
```

Output

BMSCE

SHREE VARNA.M
16/01/24 1BM22CS263

PAGE:
DATE: 23/01/2024

1) LAB-06

a) Create a package called Maths having a class called number (add and Subtract) method. Implement a simple class maths demo to use maths (outside package maths) that makes use of class which is provided by maths.

b) 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 5 Courses of the current Semester of the Student

Create another package SEE which has the class external which is derived class of Student. This class has an array that stores the SEE marks scored in 5 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.

```
// 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 sc = new Scanner(System.in);
    System.out.println("Enter USN:");
    usn = sc.next();
    System.out.println("Enter Name:");
    name = sc.next();
    System.out.println("Enter Semester")
    sem = sc.nextInt();
}
    public void displayStudentDetails(){
        System.out.println("USN:"+usn);
        System.out.println("Name:"+name);
        System.out.println("Semester:"+sem);
    }
}
```

```
// internals.java
package CIE;
import java.util.Scanner;

public class internals extends Student
{
  protected int marks[] = new int[5];

  public void inputCIEmarks(){

    Scanner sc = new Scanner(System.in);
    System.out.println("Enter Internal
                        marks for:" +name);

for(int i=0; i<5; i++)
{
  System.out.println("Subject "+(i+1)+
                     "marks:");
  marks[i] = sc.nextInt();
}
}
}
```

// external.java

```
Package SEE;

import CIE.internals;
import java.util.Scanner;

public class external extends internals
{
    protected int marks[];
    protected int final marks[];

    public external(){
        marks = new int[5];
        finalmarks = new int[5];
    }

    public void input SEE marks(){
        Scanner sc = new Scanner(System.in);
    System.out.println("Enter SEE marks"
            + ram);
    for(int i=0; i<5; i++)
    {
        System.out.println("Subject" + (i+1) +
                "marks:");
        marks[i] = sc.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]);
    }
}
}
```

// finalmarks.java

```
import SEE.external;

public class finalmarks{

    public static void main(String args[])
    {
        int numofStudents = 2;
        external finalMarks[] = new
                external[numof
                Students];
```

```
        for(int i=0; i<numofStudents; i++)
        {
          finalmarks[i] = new external();
          finalmarks[i].inputStudentDetails();
        System.out.println("Enter CIE marks");
        finalmarks[i].inputCIEmarks();
        System.out.println("Enter SEE marks");
          finalmarks[i].inputSEEmarks();
        }

        System.out.println("Displaying data:\n");

        for(int i=0; i<numofStudents; i++)
        {
          finalmarks[i].CalculateFinalmarks();
          finalmarks[i].displayfinalmarks();
        }
    }
}
```

Output:

Enter USN: 12
Enter name: Shree
Enter Sum: 3
Enter CIE marks
Enter Internal marks for Shree
Subject 1 marks: 11
Subject 2 marks: 22
Subject 3 marks: 33
Subject 4 marks: 12
Subject 5 marks: 23
Enter SEE marks for Shree
Subject 1 marks: 90
Subject 2 marks: 91
Subject 3 marks: 92
Subject 4 marks: 93
Subject 5 marks: 94

Enter USN: 63
Enter name: Varna
Enter Sum: 3
Enter CIE marks
Enter Internal marks for Varna
Subject 1 marks: 25
Subject 2 marks: 24
Subject 3 marks: 40
Subject 4 marks: 38
Subject 5 marks: 23

Enter SEE marks for varna:
Subject 1 marks: 86
Subject 2 marks: 81
Subject 3 marks: 89
Subject 4 marks: 8390
Subject 5 marks: 8499

Displaying data:

USN: 19
Name: Arul
Semester: 3
Subject 1: 67
Subject 2: 78
Subject 3: 71
Subject 4: 77
Subject 5: 55

USN: 63
Name: Varna
Semester: 3
Subject 1: 73
Subject 2: 26
Subject 3: 78
Subject 4: 90
Subject 5: 71

SHREE VARNA.N
1BM22CS263

24.01.24

LAB-07

1) WAP 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 takes both father and son's age and throws an exception if son's age is >= father's age.

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

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

class Father extends InputScanner
{
    int fatherAge;
```

```
public Father() throws WrongAge
{
    System.out.println("Enter
        Father's age:");
    fatherAge = s.nextInt();
    if(fatheAge < 0)
    {
        throw new WrongAge("Age cannot
            be negative");
    }
}

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

class Son extends Father
{
    int SonAge;
    public Son() throws WrongAge{
        super();
        System.out.println("Enter Son's
            age:")
        SonAge = s.nextInt();

        if(SonAge >= FatherAge)
        {
            throw new WrongAge("Son's
```

```
        age cannot be greater than father's
            age");

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

public void display()
{
    Super.display();
    System.out.println("Son's age:"+
    SonAge = S.nextInt() + SonAge);

}
}

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

⟹ OUTPUT :

Enter Father's Age 94
Enter Son's Age : 34

Father's Age : 94
Son's Age : 34.

SHREE VARNA M
1BM22CS263

30.01.24

LAB-08

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

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

    DisplayMessageThread (String message,
                          long interval)
    {
        this.message = message;
        this.interval = message;
    }

    public void run()
    {
        try {
            while (true){
                System.out.println(message);
                Thread.Sleep(interval);
            }
        } catch (InterruptedException e)
    { System.out.println(Thread.currentThread()
            .getName() + " interrupted.");
    }
}
```

```
public class TwoThreadDemo
{
    public static void main (String[] a
    {
        DisplayMessageThread thread1 =
        new DisplayMessageThread ("BMS
          College of Engineering", 10000);

        DisplayMessageThread thread2 =
        new DisplayMessageThread ("CSE"
          , 2000);

        Thread1.setName ("Thread 1");
        Thread2.setName ("Thread 2");

        Thread1.start();
        thread2.start();

        try{
            Thread.sleep (30000);
        }
        catch (InterruptedException e)
        {
            System.out.println ("Main
              thread interrupted");
        }
        Th
        Thread1.interrupt();
        Thread2.interrupt();
      System.out.println("Main Thread
                          exiting");
    }
}
```

Output

BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
Main Thread Exiting
Thread 1 interrupted
Thread 2 interrupted
SHREE VARNA. M
1BM22CS263

LAB - 10 (a)

▷ Demonstrate Interprocess communication and deadlock.

a) class Q{
int n;
boolean valueSet = false;
Synchronized int get()
{
while (! valueSet)
try{
System.out.println("\n Consumer waiting \n");
wait();
}
catch (InterruptedException e)
{
System.out.println("Interrupted Exception caught");
}
System.out.println("Got " + n);
valueSet = false;
System.out.println("\n Intimate 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(" InterruptedExcep
        -tion caught");
  }
  This.n=n;
  valueSet = true;
  System.out.println("Put:" + n);
  System.out.println("\n Intimate
                      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<15)
  {
    q.put(i++);
  }
}
}

class Consumer implements Runnable
{
  Q q;
  Consumer(Q q)
  {
    this.q=q;
    new Thread(this,"Consumer").start();
  }
  public void run()
  {
    int i=0;
    while(i<15)
    {
      int r=q.get();
      System.out.println("Consumed:"+r);
      i++;
    }
  }
}
```

```
class PCFixed
{
    public static void main (String args[])
    {
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-C to Stop.");
    }
}
```

OUTPUT: Press Control-C to stop.
Put: 0
Intimate Consumer
Producer waiting
Got: 0
Intimate Producer
Put: 1
Intimate Consumer
Producer waiting
Consumed: 0
Got: 1
Intimate Producer
Consumed: 1
Put: 2
Intimate Consumer
Producer waiting
Got: 2
Intimate Producer

Consumed : 2
Put : 3
Intimate Consumer
Producer waiting
Get : 3
Intimate Producer
Consumed : 3.
Put : 4
Intimate

SHREE VARNA . M
1BM22CS263

Lab-10(b)
DEADLOCK

```
class A
{
  synchronized void foo(B b)
  {
    String name = Thread.currentThread()
                  .getName();
    System.out.println("A Interrupted");
    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();
  }
}
```

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

   void last()
   {
      System.out.println("Inside A.last");
   }

}
```

```
class Deadlock implements Runnable
{
    A a = new A();
    B b = new B();
    Deadlock()
    {
        Thread.currentThread().setName
            ("Main Thread");
        Thread t = new Thread(this,
                "RacingThread");
        t.start();
        a.foo(b);
        thread
        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

SHREE VARNA
1BM22CS263

13.02.24

30/07/2024

LAB-09

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

⇒

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

class SwingDemo
{
    SwingDemo()
    {
        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 divident:");
JTextField ajtf = new JTextField(8);
JTextField bjtf = new JTextField(8);

JButton button = new JButton("Calculate");

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

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

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

```
ajtf.addActionListener(I);
bjtf.addActionListener(I);

button.addActionListener(new
                ActionListener()
{
    public void actionPerformed(Action
                -Event evt)
    {
        try{
            int a = Integer.parseInt(ajtf
                                .getText());
            int b = Integer.parseInt(bjtf
                                .getText());
            int ans = a/b;

            alab.setText("\n A = " + a);
            blab.setText("\n B = " + b);
            anslab.setText("\n Ans = " + ans);
        }

        catch(NumberFormatException e)
        {
            alab.setText("");
            blab.setText("");
            anslab.setText("");
            Err.setText("Enter only Integers!");
        }
```

```
        catch (ArithmaticException e)
        {
            alab.setText("");
            blab.setText("");
            anslab.setText("");
            err.setText("B should be Non Zero!");
        }
    }
});
    jfrm.setVisible(true);
}

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

O/P Enter the dividend and divident:
[10] [2]
[calculate] A=10 B=0 Ans=5

SHREE VARNA. M - 1BM22CS263

O/P :-

B Should be Non Zero!
Enter the divider and dividend
[10] [0]
Calculate

FUNCTIONS

- JFrame - The javax.swing.JFrame class is a type of container which inherits the java.awt.Frame class. JFrame works like the main window where components like labels, textfields are added to create a GUI.

- setSize(int width, int height) - used to resize a frame using ~~to~~ width and height parameters.

- setLayout( ) - method allows you to set the layout of the container. The layout manager helps layout the components held by this container

- setDefaultCloseOperation() - method is used to specify one of several options for the close button.
  JFrame.EXIT_ON_CLOSE - Exit the application

- JLabel - The object of JLabel class is a component for placing text in a container. It is used to display a single line of read only text.

- JTextField - The object of a JTextField class is a text component that allows the editing of a single line text. It inherits JTextComponent class.

- add(Frame) - adds new frame in the existing frame.

- ActionListener - The Java ActionListener is notified whenever you click on the button or menu item. It is notified against ActionEvent. This interface is found in java.awt.event package.

- SetText() - This method substitutes new text for all or part of the text in the text field. This works only with the first line of multi-line text fields.

- setVisible() - is a method that has return type boolen.

20.02.24

LAB 1

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.**

Program:

```
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("Root1 = " + r1 + " - i"+r2);
}

}
}

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

LAB 2

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.

Program:

```
import java.util.*;
 class Subject
 {
    int subjectmarks;
    int credits;
    int grade;
 }
  class Student
 {
   String name;
   String usn;
   double SGPA;
   Scanner s;
   Subject[] subjects;

 Student()
 {
   int i;
   subjects=new Subject[9];
   for (i=0; i<9; i++)
 {
```

```
subjects[i]=new Subject();
}
s=new Scanner(System.in);
}
void getStudentdetails()
{
System.out.println("Enter student name:");
name=s.nextLine();
System.out.println("Enter Student usn:");
usn=s.nextLine();
}

void getMarks(){
   for(int i=0; i<9; i++)
{
   System.out.println("Enter marks for subject"+(i+1)+":");
   subjects[i].subjectmarks=s.nextInt();
   subjects[i].credits=4;
 if(subjects[i].subjectmarks>=90){
    subjects[i].grade=10;}
else if(subjects[i].subjectmarks>=75){
    subjects[i].grade=9;}
else if(subjects[i].subjectmarks>=60){
    subjects[i].grade=8;}
else if(subjects[i].subjectmarks>=50){
    subjects[i].grade=7;}
else if(subjects[i].subjectmarks>=40){
    subjects[i].grade=6;}
else{
```

```
        subjects[i].grade=0;
    }
    }
    }
    void computeSGPA()
    {
     double totalcredits=0;
     double totalgradepoints=0;
     for(int i=0; i<9; i++){
    totalgradepoints += subjects[i].grade*subjects[i].credits;
    totalcredits += subjects[i].credits;
    }
    SGPA= totalgradepoints/totalcredits;
    }
    }
    class Main
    {
       public static void main(String args[])
    {
     Student s2=new Student();
     s1.getStudentdetails();
     s1.getMarks();
     s1.computeSGPA();
    System.out.println("\n Student details");
    System.out.println("Name:"+s1.name);
    System.out.println("usn:"+s1.usn);
    System.out.println("SGPA:"+s1.SGPA);
    }
}
```

LAB 3

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.

Program:

```
import java.util.*;
class books{
	String name;
	String author;
	int price;
	int numpages;
	books(String name,String author,int price,int numpages)
	{
		this.name=name;
		this.author=author;
		this.price=price;
		this.numpages=numpages;
	}
public String toString()
{
	String name,author,price,numpages;
	name="Book name:"+this.name+"\n";
	author="Author name:"+this.author+"\n";
	price="price:"+this.price+"\n";
	numpages="number of pages:"+this.numpages+"\n";
	return name+author+price+numpages;
}
}
public class books1
{
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 the number of books");
        n=s.nextInt();
        books b[];
        b=new books[n];
        int i;

for (i=0;i<n;i++)
{
        Scanner sc=new Scanner(System.in);
        System.out.println("enter the name author price numpages:");
        name=sc.next();
        author=sc.next();
        price=sc.nextInt();
        numpages=sc.nextInt();
        b[i]=new books(name,author,price,numpages);
}
for(i=0;i<n;i++)
{
        System.out.println(b[i].toString());
}
}
}
```

LAB 4

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.

Program:

```
import java.util.Scanner;
 abstract class Shape1
{
  int a,b;
  Shape1(int a,int b)
{
  this.a=a;
  this.b=b;
}
abstract void printarea();
}
class Rectangle extends Shape1
{
   Rectangle(int a, int b)
{
   super(a,b);
}
   void printarea()
{
```

```
  System.out.println("Area of rectangle:"+(a*b));
  }
}
class Triangle extends Shape1
{
  Triangle(int a, int b)
{
  super(a,b);
}
  void printarea()
{
  System.out.println("Area  of  trangle:"+(0.5*a*b));
  }
}
class Circle extends Shape1
{
  Circle(int a, int b)
{
  super(a,b);
}
  void printarea()
{
  System.out.println("Area of circle:"+(3.14*a*a));
  }
}

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

```
{
  Scanner s=new Scanner(System.in);
  System.out.println("Enter length and breadth of Rectangle: ");
            int length = s.nextInt();
            int breadth = s.nextInt();
            Rectangle rectangle  = new Rectangle(length, breadth);
 System.out.println("Enter base and height of Triangle: ");
            int base = s.nextInt();
            int height = s.nextInt();
            Triangle triangle = new Triangle(base, height);
 System.out.println("Enter radius of Circle: ");
            int radius = s.nextInt();
            Circle circle = new Circle(radius, radius);
 rectangle.printarea();
 triangle.printarea();
 circle.printarea();
}
}
```

LAB 5

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.

Check for the minimum balance, impose penalty if necessary and update the balance.

Program:

```
import java.util.*;
class account{

   String customer_name;
   int acc_no;
   String type;
   double balance;


   account(String name,int no,String type,double balance){
      customer_name=name;
```

```
        acc_no=no;
        this.type=type;
        this.balance=balance;

    }
    void deposit(double amount){
        balance+=amount;
    }
    void withdraw(double amount){
        if((balance-amount)>0){
            balance-=amount;
        }
        else{
            System.out.println("Insufficient balance");
        }
    }
    void display(){
        System.out.println("Name:"+customer_name+"\n"+"account
number:"+acc_no+"\n"+"Type:"+type+"\n"+"Balance:"+balance);
    }
}
class sav_acc extends account{
    private static double rate=2;
    sav_acc(String name,int acc_no,double balance)
    {
    super(name,acc_no,"savings",balance);
    }
    void interest()
    {
        balance+=balance*(rate)/100;
        System.out.println("balance:"+balance);
    }
}
class curr_acc extends account{
    curr_acc(String name,int acc_no,double balance)
    {
    super(name,acc_no,"current",balance);
    }
```

```
    private double minbal=250;
    private double servicecharge=100;
    void checkmin(){
        if(balance<minbal){
            System.out.println("Insufficient balance so serivce charges are subtracted from
your balance amount");
            balance-=servicecharge;
            System.out.println("Balance is:"+balance);
        }
    }
}

public class bank {
    public static void main(String args[]){
        Scanner s=new Scanner(System.in);
        System.out.println("SHREE VARNA M:");
        System.out.println("1BM22CS263");
        System.out.println("enter the name :");
        String name=s.next();
        System.out.println("enter the type(current/savings):");
        String type=s.next();
        System.out.println("enter the account number:");
        int acc_no=s.nextInt();
        System.out.println("enter the intial balance:");
        double balance=s.nextDouble();
        int ch;
        double amount1,amount2;
        account acc=new account(name,acc_no,type,balance);
        sav_acc sa=new sav_acc(name,acc_no,balance);
        curr_acc ca=new curr_acc(name,acc_no,balance);
        while(true)
        {
            if(acc.type.equals("savings"))
            {
                System.out.println("\nMenu\n1.Deposit 2.Withdraw 3.Interest 4.Display");
                System.out.println("enter the choice:");
                ch=s.nextInt();
                switch(ch)
```

```
    {
        case 1:System.out.println("enter the amount:");
            amount1=s.nextInt();
            sa.deposit(amount1);
            break;
        case 2:System.out.println("enter the amount:");
            amount2=s.nextInt();
            sa.withdraw(amount2);
            break;
        case 3:sa.interest();
            break;
        case 4:sa.display();
            break;
        default:System.out.println("invalid input");
                System.exit(0);
    }
}
else
{
    System.out.println("\nMenu\n1.Deposit 2.Withdraw  3.Display");
    System.out.println("enter the choice:");
    ch=s.nextInt();
    switch(ch)
    {
        case 1:System.out.println("enter the amount:");
                amount1=s.nextInt();
                ca.deposit(amount1);
                break;
        case 2:System.out.println("enter the amount:");
                amount2=s.nextInt();
                ca.withdraw(amount2);
                ca.checkmin();
                break;

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

```
            }
        }
    }
}
}
```

LAB 6

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.

Program:

STUDENT FILE

```
package CIE;
import java.util.Scanner;
 public class Student1
{
    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 the Student usn:");
     usn=s.next();
```

```
    System.out.println("Enter the Student name: \n");
    name=s.nextLine();
    System.out.println("Enter the Student sem: \n");
    sem=s.nextInt();
  }
  public void displayStudentDetails()
  {
    System.out.println("USN: " + usn);
    System.out.println("Name: " + name);
    System.out.println("Semester: " + sem);
}
}
```

INTERNAL FILE

```
package CIE;
import java.util.Scanner;
 public class internal extends Student1
{
  protected int marks[] = new int[5];
  public void inputCIEmarks()
  {
    Scanner s1=new Scanner(System.in);
    for(int i=0; i<5; i++)
    {
      System.out.println("Enter internal marks of
```

```
CIE:"+(i+1));
      marks[i]=s1.nextInt();
   }

}
}
```

EXTERNAL FILE

```
package SEE;
import  CIE.internal;
import java.util.Scanner;
 public class external extends internal
 {
protected int marks[];
protected int finalMarks[];
public external()
{
marks = new int[5]; finalMarks = new int[5];
}
public void inputSEEmarks()
{
Scanner s = new Scanner(System.in);
for(int i=0;i<5;i++)
 {
System.out.print("Subject "+(i+1)+" marks: ");
marks[i] = s.nextInt();
 }
}
```

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

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

MAIN FILE

```
import SEE.external;
class Main
{
 public static void main(String args[])
{
int numOfStudents = 2;
external finalMarks[] = new
external[numOfStudents];
for(int i=0;i<numOfStudents; i++)
{
finalMarks[i] = new external();
finalMarks[i].inputStudentDetails();
```

```
System.out.println("Enter CIE marks"+(i+1));
finalMarks[i].inputCIEmarks();
System.out.println("Enter SEE marks"+(i+1));
finalMarks[i].inputSEEmarks();
}
System.out.println("Displaying data:\n");
for(int i=0;i<numOfStudents; i++)
{
finalMarks[i].calculateFinalMarks();
finalMarks[i].displayFinalMarks();
}
}
}
```

LAB 7

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.

Program:

```
import java.util.Scanner;

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

class Father {
    int fatherAge;

    Father() throws WrongAge {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter Father's age: ");
        fatherAge = s.nextInt();
        if (fatherAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
```

```
    }

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

class Son extends Father {
    int sonAge;

    Son() throws WrongAge {
        super();
        Scanner s = new Scanner(System.in);
        System.out.println("Enter Son's age: ");
        sonAge = s.nextInt();
        if (sonAge >= fatherAge) {
            throw new WrongAge("Son's age cannot be greater than father's age");
        } else if (sonAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
    }

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

 class Main {
```

```
    public static void main(String[] args) {
        try {
            Son s = new Son();
            s.display();
        } catch (WrongAge e) {
            System.out.println(e.getMessage());
        }
    }
}
```

LAB 8

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.

Program:

```
class College extends Thread {
    public void run() {
        for (int i = 0; i <5; i++) {
            System.out.println("BMS COLLEGE OF ENGINEERING");
           try {
               Thread.sleep(10000);
           } catch (InterruptedException e) {
               e.printStackTrace();
           }
       }
    }
}

class Dept extends Thread {
    public void run() {
        for (int i = 1; i <= 25; i++) {
      System.out.println("CSE");
           try {
               Thread.sleep(2000);
           } catch (InterruptedException e) {
               e.printStackTrace();
```

```
            }
        }

    }
}

class Main {
    public static void main(String args[]) {
        College c1 = new College();
        c1.start();
        Dept d1 = new Dept();
        d1.start();
    }
}
```

LAB 9

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.

Program:

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

class SwingDemo{
   SwingDemo(){

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

      JLabel jlab = new JLabel("Enter the divider and divident:");

      JTextField ajtf = new JTextField(8);
      JTextField bjtf = new JTextField(8);

      JButton button = new JButton("Calculate");

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

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

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

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

button.addActionListener(new ActionListener() {
   public void actionPerformed(ActionEvent evt) {
      try{
         int a = Integer.parseInt(ajtf.getText());
         int b = Integer.parseInt(bjtf.getText());
         int ans = a/b;

         alab.setText("\nA = " + a);
         blab.setText("\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!");
      }
   }
});
```

```
        jfrm.setVisible(true);
    }

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

LAB 10

10. Demonstrate Inter process Communication and deadlock

Program:

A. Deadlock

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

```
System.out.println(name + " entered B.bar");
try {
Thread.sleep(1000);
} catch(Exception e) {
System.out.println("B Interrupted");
}

System.out.println(name + " trying to call A.last()");
a.last();


}

void last() {
System.out.println("Inside  A.last");
}
}
class Deadlock implements Runnable

{
A a = new A();
B b = new B();
Deadlock() {

Thread.currentThread().setName("MainThread");
Thread t = new Thread(this,"RacingThread");
t.start();
a.foo(b); // get lock on a in this thread.
System.out.println("Back in main thread");

}

public void run() {
b.bar(a); // get lock on b in other thread.


System.out.println("Back  in  otherthread");

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

new Deadlock();

}}
```

B. InterprocessCommunication:

```
class Q {
int n;
boolean valueSet = false;
synchronized int get() {
while(!valueSet)
try {


System.out.println("\nConsumer waiting\n");
wait();
} catch(InterruptedException e) {


System.out.println("InterruptedException    caught");
}
System.out.println("Got: " + n);
valueSet = false;
System.out.println("\nIntimate Producer\n");
notify();
return n;
}
synchronized void put(int n) {
while(valueSet)
try {
```

```
System.out.println("\nProducer waiting\n");
wait();
} catch(InterruptedException e) {
System.out.println("InterruptedException caught");
}
this.n = n;
valueSet = true;
System.out.println("Put: " + n);
System.out.println("\nIntimate Consumer\n");
notify();
}
}

class Producer implements Runnable {
Q q;
Producer(Q q) {
this.q = q;
new Thread(this, "Producer").start();
}
public void run() {
int i = 0;
while(i<15) {
q.put(i++);
}
}
}

class Consumer implements Runnable {
Q q;
Consumer(Q q) {
```

```
this.q = q;
new Thread(this, "Consumer").start();
}
public void run() {
int i=0;
while(i<15) {
int r=q.get();
System.out.println("consumed:"+r);
i++;
}
}
}

class PCFixed {
public static void main(String args[]) {
Q q = new Q();
new Producer(q);
new  Consumer(q);
System.out.println("Press Control-C to stop.");
}
}
```