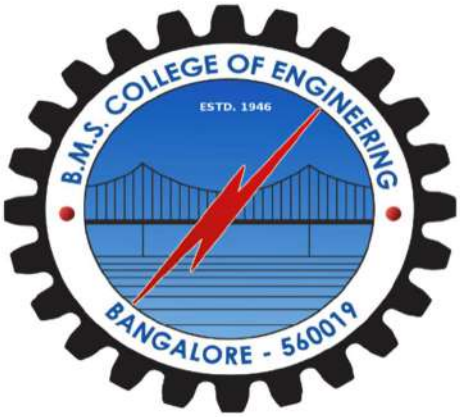
B.M.S. College of Engineering

(Autonomous College Affiliated to Visvesvaraya Technological University, Belgaum)
Bull Temple Road, Basavanagudi, Bengaluru – 560019



Department of
Computer Science & Engineering (CSE)

Object Oriented Java Programming (LAB)

(23CS3PCOOJ)

Name: Shamanth K Murthy

USN: 1BM22CS251

INDEX

Name Shamanth. K. Murthy Sub.

Std.: Div. Roll No.

Telephone No. E-mail ID.

Blood Group. Birth Day.

Sr.No.	Title	Page No.	Sign./Remarks
LAB 1	Quadratic Equation	10	
LAB 2	SGPA Calculator	10	26/12/23
LAB 3	Book Details	10	
LAB 4	Area of Shapes using Abstract Class	10	2/1/2024
LAB 5	Bank Account - Savings & Current Account	10	16/1/2024
LAB 6	CIE & SEE Package	10	30/1/2024
LAB 7	Exception Handling	10	1/2/2024
LAB 8	Multithreaded Program	10	6/2/2024
LAB 10 a) b)	Inter Process Communication Dead Lock		13/2/2024
LAB 9	Division		23/2/2024

LAB - 1

Quadratic Equation

```
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");
        a = s.nextInt();
        b = s.nextInt();
        c = s.nextInt();
    }
    void compute()
    {
        while (a==0)
        {
            System.out.println("Not a quadratic eq");
            System.out.println("Enter a nonzero value:");
            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 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 & distinct");
    System.out.println("Root1 = "+r1+" Root2 "
                       +r2);
}
else if (d<0)
{
    System.out.println("Roots are imaginary");
    r1 = (-b)/(2*a);
    r2 = Math.sqrt(-d)/(2*a);
    System.out.println("Root1 = "+r1+"+i"+r2);
    System.out.println("Root2 = "+r2+"+i"+r2);
}
}

class Quadratic Main
{
    public static void main(String args[])
    {
        Quadratic q = new Quadratic();
        q.getd();
        q.compute();
        System.out.println("Shamanth
                           -1BM22CS251");
    }
}
```

Output:

Entu the coefficients of a, b, c
1
2
1

Roots are real and equal
Root 1 = Root 2 = -1.0
Shamanth - 1BM22CS251

Entu coefficients of a, b, c
1
3
2.
Roots are real and distict
Root1 = -1.0 Root2 = -2.0
Shamanth - 1BM22CS251

Entu coefficients of a, b, c
2
1
3
Roots are imaginary
Root 1 = 0.1 + i 1.198578808
Root 2 = 0.0 - i 1.198578808
Shamanth - 1BM22CS251

LAB - 2.

```
import java.util.Scanner;

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

class Student
{
    Subject subject[];
    String name;
    String usn;
    double sgpa = 0;
    Scanner s;

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

    void getStudentDetails()
    {
        System.out.println("Enter your name");
        name = s.next();
        System.out.println("Enter your usn");
        usn = s.next();
    }
```

```
void getMarks()
{
    for(int i=0; i<8; i++)
    {
        System.out.println("Enter marks for subject"
                + (i+1) + ":");
        subject[i].subjectMarks = s.nextInt();
        System.out.println("Enter credits");
        subject[i].credits = s.nextInt();
        subject[i].grade = (subject[i].subjectMarks
        if (subject[i].grade > 11)
            subject[i].grade = 10;
        if (subject[i].grade <= 4)
            subject[i].grade = 0;
    }
}
void computeSGPA()
{
    int totalCredits = 0;
    for(int i = 0; i<8; i++)
    {
        sgpa += (subject[i].credits * subject[i].grade);
        totalCredits += subject[i].credits;
    }
    sgpa = sgpa / totalCredits;
}
}
Class main
{
public static void main(String args[])
{
    Student s1 = new Student();
    s1.getStudentDetails();
```

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

Output:

Entu your Name
Shamanth
Entu your Usn
1BM22CS251
Entu submarks of subject 1
91
Entu credits of subject 1
4
Entu marks of subject 2
85
Entu Credits of Subject 2
4
Entu marks of subject 3
85
Entu Credits of subjcd 3
3
Entu marks of subject 4
86
Entu Credits of Subject 4
3
Entu marks of subject 5
81
Entu credits of subject 5
3

Enter marks of subject 6
85
Enter credits of subject 6
1
Enter marks of subject 7
95
Enter credits of subject 7
1
Enter marks of subject 8
98
Enter credits of subject 8
1

Name: shamanth
USN: 1BM22CS251
SGPA: 9.3

19.1

LAB - 3

Q) 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() that could display the complete details of the book. Develop a java program to create n book objects.

```
import java.util.Scanner;

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

    Books (String name, String author, int price,
            int numPages)
    {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }
    public String toString()
    {
        String name, author, price, numPages;
        name = "Book Name: " + this.name + "\n";
        author = "Author Name: " + this.author + "\n";
        price = "Price: " + this.price + "\n";
```

```
numPages = "Number of Pages : "+this.numPages
                                    +"\n";
return name + author + price + numPages;
}
};

class Main{

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

Output:

Enter the Number of Books.
2
Enter the name, author, price and number of pages of the book:
Java
xyz
1200
100

Enter the name, author, price and number of pages of the book:
C++
abc
1500
243

The Book details:
Book name : Java
Author name : xyz
Price : 1200
Number of Pages : 100

Book Name : C++
Author Name : abc
Price : 1500
Number of Pages : 243

26/2/23

Shamanth K Murthy
1BM22CS251

02/01/2024

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 & Circle such that each one of the classes extend the class Shape.

```
import java.util.Scanner;

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

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

class Rectangle extends Shape{
    void getInput(){
        System.out.println("Enter the dimensions
            for Rectangle");
        a = ss.nextDouble();
        b = ss.nextDouble();
    }
```

```
    void printArea(){
        System.out.println("Area of Rectangle
                = " + (a*b));
    }
};

class Triangle extends Shape{
    void getInput(){
        System.out.println("Enter dimensions of
                Triangle");
        a = s.nextDouble();
        b = s.nextDouble();
    }

    void printArea(){
        System.out.println("Area of triangle="
                + (0.5*a*b));
    }
};

class Circle extends Shape{
    void getInput(){
        System.out.println("Enter the dimensions
            of circle");
        a = s.nextDouble();
    }

    void printArea()
    {
        System.out.println("Area of Circle=" +
                (3.14*a*a));
    }
};
```

```
class Main{
    public static void main (String [] args)
    {
        Rectangle r = new Rectangle();
        Triangle t = new Triangle();
        Circle c = new Circle();

        r.getInput();
        t.getInput();
        c = getInput();

        r.getprintArea();
        t.printArea();
        c.printArea();
    }
};
```

Output:

```
Enter the dimensions of rectangle
2 3
Enter the dimensions of triangle
2 4
Enter the dimensions of circle
2
Area of Rectangle = 6.0
Area of Triangle = 4.0
Area of Circle = 12.56.
```

2/1/2024

Shamanth K Murthy 1BM22CS251

09/01/2024

SURYA Gold
Date Page

LAB-5
Bank Account

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. 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

```
import java.util.Scanner;

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

class Account extends Input{
    String name;
    int accNo;
    double balance;

    void getDetails(){
        System.out.println("Enter Name:");
        name = sc.nextLine();
        System.out.print("Enter acc No:");
        accNo = sc.nextInt();
    }
    void deposit(){
        System.out.println("Enter amount to deposit:");
        double amt = sc.nextDouble();
        balance += amt;
```

```
        System.out.println("Amount Deposited");
    }

    void withdraw(){
        System.out.print("Enter amount:");
        double amt = sc.nextDouble();
        if(balance >= amt){
            balance -= amt;
            System.out.println("Amount Withdrawn");
        } else {
            System.out.println("Insufficient Balance");
        }
    }

    void display(){
        System.out.println("Name: "+name);
        System.out.println("Account Number:"
                           +accNO);
        System.out.println("Balance: "+balance);
    }
}

Current
class Savings extends Account {
    double minbal = 500;
    double penalty = 100;

    void withdraw(){
        super.withdraw();
        checkMinBalance();
    }
    Private void checkMinBalance(){
        if(balance < minBal){
            balance -= penalty;
            System.out.println("Penalty applied for
                low balance");
        }
    }
```

```
class Savings extends Account {
    double interestRate = 0.04;

    void compute Interest() {
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest Credited : " + interest);
    }
}

class Bank extends Account {
    public static void main(String[] args)
    {
        Savings ob1 = new Savings();
        Current ob2 = new Current();
        ob1.getDetails();
        ob2.getDetails();
        int choice;
        String acc;
        System.out.println("Menu:");
        System.out.println("1. Deposit \n
        2. Withdraw \n 3. Display Balance \n
        4. Compute Interest (Savings Only)" \n
        5. Exit");
        do {
            System.out.print("Enter you choice:");
            choice = sc.nextInt();
            System.out.print("Enter the acctype");
            acc = sc.next();
            switch (choice) {
            case 1:
                if (acc.equals("saving"))
                    ob1.deposit();
```

```
else {
    Ob2.deposit();
    break;
case 2:
    if (acc.equals("saving"))
        Ob1.withdraw();
    else
        Ob2.withdraw();
    break;
case 3:
    if (acc.equals("saving"))
        Ob1.display();
    else
        Ob2.display();
    break;
case 4:
    Ob1.computeInterest();
    break;
case 5:
    break;
default:
    System.out.println("Invalid choice"
}
} while (option != 5);
}
}
```

SURYA Gold
Date
Page

OUTPUT:

Enter name: Shamanth
Enter accNO: 1
Enter name: Shreyas
Enter accNo: 2

Menu:
1. Deposit
2. Withdraw
3. Display
4. Compute Interest (Savings Only)
5. Exit

Enter your choice: 1
Enter acc Type: saving
Enter deposit amount: 1000
Enter your choice: 1
Enter acc type: current
Enter deposit amount: 5000
Enter your choice: 2
Enter acc type: saving
Enter withdraw amount: 500
Enter your choice: 2
Enter acc type: current
Enter withdraw amount: 4600
Penalty applied.
Enter your choice: 3
Enter acc type: current
Name: Shreyas
Acc number: 2
Balance: 300
Enter your choice: 4
Interest credited: 40
Enter your choice: 2
Enter acc type: saving
Enter withdraw amount: 1050 Insufficient Balance

Shamanth K Murthy
1BM22CS251

16/12/2024

23/01/24

SURYA Gold
Date Page

LAB-6

Create a package CIE which has two classes Student & Internals. The class student has members usn, name, sem. The class internals derived from student has an array that stores the internal marks scored in 5 subjects. Create another package SEE containing a class External which has an array to store SEE marks.

Student.java

```
Package CIE;
import java.util.*;
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);
        this.usn = sc.nextLine();
        this.name = sc.nextLine();
        this.sem = sc.nextInt();
    }

    public void displayStudentDetails(){
        System.out.println(this.usn+" "+this.name);
    }
}
```

Internals.java.

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

Externals.java.

```
Package SEE;
import CIE.Internals;
import java.util.Scanner;
public class Externals extends Internals {
    protected int marks[];
    protected int finalMarks[];
    public Externals() {
        marks = new int[5];
        finalMarks = new int[5];
    }
    public void inputSEEmarks() {
        for(int i=0; i<5; i++)
            marks[i] = sc.nextInt();
    }
    public void calcFinalMarks() {
        for(int i=0; i<5; i++)
            finalMarks[i] = marks[i]/2;
                + super.marks[i];
    }
```

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

Main.java.

```
import SEE.Externals;
class Main{
    public static void main(String args[]){
        int numOfStudents = 2;
        Externals finalMarks[] = new Externals[];
        for(int i=0; i<numOfStudents; i++){
            finalMarks[i] = new Externals();
            finalMarks[i].inputStudentDetails();
            System.out.println("Enter CIE Marks");
            finalMarks[i].inputCIEmarks();
            System.out.println("Enter SEE Marks");
            finalMarks[i].inputSEEmarks();
        }
        for(int i=0; i<numOfStudents; i++){
            final[i].calculateFinalMarks();
            final[i].displayFinalMarks();
        }
    }
}
```

SURYA Gold
Date
Page

OUTPUT:

Enter Student Details:
$BM22CS251
Shamanth
2
Enter CIE Marks
40
45
41
48
44
Enter SEE Marks
80
90
86
80
84

Enter Student Details
1BM22CS256
Shashank
2
Enter CIE marks
50
45
40
48
46
Enter SEE marks
90
96
100
90
92

Displaying Details:
USN: 1BM22CS251
Name: Shamanth
Sem: 2
Subject 1: 80
Subject 2: 90
Subject 3: 84
Subject 4: 88
Subject 5: 86
USN: 1BM22CS256
Name: Shashank
Sem: 2
Subject 1: 95
Subject 2: 93
Subject 3: 90
Subject 4: 96
Subject 5: 92

LAB - 7
Exception Handling

Write a Program that demonstrates Handling of exceptions in Inheritance tree. Create a base class Father & derive class Son. In Father class implement a constructor which throws WrongAge when age < 0. In Son class implement a constructor which throws WrongAge() when age >= Father age

```
import java.util.Scanner;

class WrongAge extends Exception{
    public WrongAge(String s){
        super(s);
    }
}
class Input{
   Scanner sc = new Scanner(System.in);
}
class Father extends Input{
      int fatherAge;
      Father(){
        fatherAge = sc.nextInt();
        try{
            check();
        } catch (WrongAge e){
            System.out.println(e);
        }
        void check() throws WrongAge{
            if (fatherAge < 0)
              { throw new WrongAge("Cannot be
                                    negative");
}   }
```

```
    void display(){
        System.out.println("Father Age: "+fatherAge);
    }
}
class Son extends Father{
    int SonAge;
    Son(){
        super();
        try{
            check();
        }
        catch(WrongAge e){
            System.out.println(e);
        }
    }
    void check() throws WrongAge{
        if(SonAge<0)
            throw new WrongAge("cannot be negative");
        else if(sonAge>fatherAge)
            throw new WrongAge("Son Age can't
                be greater than father's Age");
        else if(sonAge==fatherAge)
            throw new WrongAge("Son Age can't be
                equal to father's age");
    }
    void display(){
        System.out.println(e);
    }
}
```

```
class Main{
public static void main (String args[])
  {
      Son so = new Son();
      so.display();
  }
}
```

OUTPUT:
50
30
Father Age : 50
Son Age : 30

-12
Age cannot be negative

20
40
Son's Age cannot be greater than Father's Age

1/2/2024

Shamanth K Murthy 1BM22CS251

6/2/24

SURYA Gold
Date
Page

LAB-8

Write a Program which creates two threads, one displaying "BMS College of Engineering" every 10 sec & another displaying "CSE" every 2 sec

```
class displayMessage implements Runnable {
    String Message;
    int interval;

    displayMusage (String message, int interval){
        this.message = message;
        this.intu val = intu val;
    }

    public void run (){
        while(true){
            System.out.println('message);
            try{
                Thread.sleep(interval *1000);
            } catch (InterruptedException e){
                System.out.println(e);
            }
        }
    }
}

public class multi{

    public static void main (String args[]){
        Thread t1 = new Thread( new displayMessage
            ("BMS College of Eng", 10);
        Thread t2 = new Thread( new Displaymessage
```

```
("CSE", 2);

t1.start();
t2.start();
}
}
```

OUTPUT:

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

6/2/2024

20/4/24

SURYA Gold
Date Page

LAB - 9

WAP that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, num1 & num2, The division of Num1 & Num2 is displayed in Result field when the divide is if Num1 & Num2 were not an Integer, the Program would throw an arithmetic exception display the messag dialog box.

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

class SwingDemo {

    SwingDemo() {
        JFrame jfm = new JFrame("Divider App");
        jfm.setSize(275, 180);
        jfm.setLayout(new FlowLayout());
        jfm.setDefaultCloseOperation(JFrame.EXIT
                                     ON_CLOSE);
        JLabel jlab = new JLabel("Enter the Divider
                                  and dividend");
        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);

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

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

```
ActionListener l = new ActionListener().
{.
    public void actionPerformed(ActionEvent ev
    {
        System.out.println("Action Event from a tex
    }
};
ajtf.addActionListener(l);
bjtf.addActionListener(l);

buttonActionListener(new ActionListener) {
    public void actionPerformed.
    {
        try{.
            int a = Integer.parseInt(ajit.getText());
            int b = Intiger.parseInt(bjit.getText())
            alab.setText("\nA = "+a).
            blab.set Text("\nans="+ans).
        } catch(NumberFormatException c)
        {
            alab.setText(" n");
            blab.setText(" ");
            anslab.setText(");
        } catch (Arithematic Exception e) {
            alab.set Text(" n")
            blab.set Text(" n");
            cri.set Text("B should be Non zero");
        }
    }
});
jfm.(set Visible (true);
}
```

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

OUTPUT:

Enter the divider and dividend

10 2

Calculate A=10 B=2 Ans=5

Shamanth K Murthy
1BM22CS251

Functions

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

setSize(int width, int height) - used to resize a frame using width & height parameters

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

setDefault(CloseOperation()- methods is used to specify one of several options for the close button JFrame. EXIT_ON_CLOSE

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

JTextField - The object of a JTextFeild class is a text component that allows the editions of a single line text. It inherits JText component class.

add(frame) - adds new frame to the existing frames.

Action Listener - The Java ActionListener is notified whenever you click on the button or menu item It is notified against ActionEvent

SetText() - This method substitutes new text for all or part of the text in the text field.

SetVisible() is a method that has return type boolean.

23/2/2024

13/02/24

SURYA Gold

Date Page

LAB - 10a

Demonstrate Inter Process Communication

```
class Q{
    int n;
    boolean valueSet = false;
    synchronized int get(){
        while(!valueSet)
            try{
                System.out.println("\nConsumer Waiting");
                wait();
            } catch(InterruptedException e){
                System.out.println("\nInterrupted
                    Exception caught");
            }
        System.out.println("Got : "+n);
        valueSet = false;
        System.out.println("\nIntimate Producer");
        notify();
        return n;
    }
    synchronized void put(int n){
        while(valueSet)
            try{
                System.out.println("\nProducer Waiting");
                wait();
            } catch(InterruptedException e){
                System.out.println("\nInterrupted
                    Exception caught");
            }
        this.n = n;
        valueSet = true;
```

```
        System.out.println("Put: "+n);
        System.out.println("Intimate consumer");
        notify();
    }
}

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

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

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

OUTPUT:

```
Put : 0
Got : 0
Put : 1
Got : 1
Put : 2
Got : 2
Put : 3
Got : 3
Put : 4
Got : 4
```

13/2/24

SURYA Gold
Date Page

LAB - 10 b

Demonstrate Deadlock

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

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

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

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

class Deadlock implements Runnable{
    A a = new A();
    B b = new B();
    Deadlock(){
        Thread.currentThread().setName("MainThread");
        Thread t = new Thread(this, "RacingThread");
        t.start();
        a.foo(b);
        System.out.println("Back in mainthread");
    }
    public void run(){
        b.bar(a);
        System.out.println("Back in otherthread");
    }
    public static void main(String args[]){
        new Deadlock();
    }
}
```

OUTPUT:

```
Mainthread entered A.foo
Racingthread entered B.bar
Racing thread trying to call A.last()
Inside A.last
Back in otherhead
Main Thread trying to call B.last()
Inside A.last
Back in mainthread
```

LAB: 1

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

```
import java.util.Scanner;
class Quadratic
{
	int a, b, c;
	double r1, r2, d;
	void getd()
	{
		Scanner s = new Scanner(System.in);
		System.out.println("Enter the coefficients of a,b,c");
		a = s.nextInt();
		b = s.nextInt();
		c = s.nextInt();
	}
	void compute()
	{
		while(a==0)
		{
			System.out.println("Not a quadratic equation");
			System.out.println("Enter a non zero value for a:");
			Scanner s = new Scanner(System.in);
			a = s.nextInt();
		}
		d = b*b-4*a*c;
		if(d==0)
		{
			r1 = (-b)/(2*a);
			System.out.println("Roots are real and equal");
			System.out.println("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

Develop a Java program to create a class Student with members usn, name, an array credits and an array mark. Include methods to accept and display details and a method to calculate SGPA of a student.
//Develop a java program to create a class Student with members usn,name, an array creadits 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 Student{
   String name;
   String usn;
   double SGPA;
   Subject subject[];
   Scanner s;

   Student(){
      int i;
      subject=new Subject[9];
      for(i=0;i<9;i++){
         subject[i]=new Subject();
         s=new Scanner(System.in);
      }

   }

   void getStudentDetails(){
      System.out.println("Enter your Name:");
      name=s.next();
      System.out.println("Enter your USN:");
      usn=s.next();
   }

   void getMarks(){
      for(int i=0;i<9;i++){
         System.out.println("Enter 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].subjectMarks/10)+1;

         if (subject[i].grade==11){
            subject[i].grade=10;
         }
         if (subject[i].grade<=4){
            subject[i].grade=0;
         }

      }
   }

   void computeSGPA(){
      int effectiveScore=0;
      int totalCreadits=0;
```

```
      for(int i=0;i<9;i++){
         effectiveScore += (subject[i].grade*subject[i].credits);
         totalCreadits += subject[i].credits;
      }

      SGPA=(double)effectiveScore/(double)totalCreadits;
   }
}
class Subject{
         int subjectMarks;
         int credits;
         int grade;

}
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);
         }
}
```

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,page_num;

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

   String toStrings(){
      String name,author,price,page_num;
      name="Book name: "+this.name+"\n";
      author="Author name: "+this.author+"\n";
      price="Price: "+this.price+"\n";
      page_num="Number of pages: "+this.page_num+"\n";

      return (name+author+price+page_num);
   }

   public static void main(String args[]){
      int n;
      String name,author;
      int price,page_num;

      Scanner sc=new Scanner(System.in);

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

      Book b[]=new Book[n];

      for (int i=0;i<n;i++){
         System.out.println("Enter name of book:");
         sc.nextLine();
         name=sc.nextLine();

         System.out.println("Enter author of a book:");
         author=sc.nextLine();

         System.out.println("Enter the price of book:");
         price=sc.nextInt();

         System.out.println("Enter the no. of pages of book:");
         page_num=sc.nextInt();
```

```
            b[i]=new Book(name,author,price,page_num);
        }

        System.out.println("Book Details:");
        for(int i=0;i<n;i++){
            System.out.println("Book "+(i+1)+" :\n"+b[i].toStrings());
        }
    }
}
```

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;
public class InputScanner {
	Scanner s;
	InputScanner(){
		s=new Scanner(System.in);
	}
}
abstract public class Shape extends InputScanner{
	double a,b;
	abstract void getInput();
	abstract void displayArea();
}
public class Rectangle extends Shape {
	void getInput() {
		System.out.println("Enter the dimension of rectangle:");
		a=s.nextDouble();
		b=s.nextDouble();
	}

	void displayArea() {
		System.out.println("Area of reactangle:"+(a*b));
	}
}
public class Circle extends Shape{
	void getInput() {
		System.out.println("Enter the dimension of circle:");
		a=s.nextDouble();
	}

	void displayArea() {
		System.out.println("Area of circle:"+(3.14*a*a));
	}
}
public class Triangle extends Shape {
	void getInput() {
		System.out.println("Enter the dimension of triangle:");
		a=s.nextDouble();
		b=s.nextDouble();
	}

	void displayArea() {
		System.out.println("Area of triangle:"+((a*b)/2));
	}
}
public class MainMethod {

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

```
                Rectangle R=new Rectangle();
                R.getInput();
                R.displayArea();

                Triangle T=new Triangle();
                T.getInput();
                T.displayArea();

                Circle C=new Circle();
                C.getInput();
                C.displayArea();
        }

}
```

LAB: 5

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

```
import java.util.Scanner;
public class Account {
	String customer_name;
	int account_no;
	String type_acc;
	double balance=0;
	Scanner sc;

	Account(String customer_name,int account_no,String type_acc){
		this.customer_name=customer_name;
		this.account_no=account_no;
		this.type_acc=type_acc;
		sc=new Scanner(System.in);
	}

	void diposite() {
		System.out.println("Enter the deposit amount:");
		int dipo=sc.nextInt();
		balance+=dipo;
	}

	void withdrawal() {
		System.out.println("Enter the withdrawl amount:");
		int with=sc.nextInt();
		if(with>balance) {
			System.out.println("Insufficient Balance");
		}
		else{
			balance-=with;
		}
	}

	void display() {
		System.out.println("Customer name:"+customer_name);
		System.out.println("Account number:"+account_no);
		System.out.println("Type of Account:"+type_acc);
		System.out.println("Balance:"+balance);
	}
	void applyinterest() {}

}

public class Cur_acct extends Account {

	Cur_acct(String customer_name,int account_no,String type_acc){
		super(customer_name,account_no,type_acc);}
```

```
	void withdrawal() {
		System.out.println("Enter the withdrawl amount:");
		int with=sc.nextInt();
		if (balance<=2000) {
			double pen=balance/(0.06);
			System.out.println("Insuffiecient balance penalty to be paid:"+pen);
			balance+=pen;
		}
		else{
			balance-=with;
		}
	}
}

public class Sav_acct extends Account{
Sav_acct(String customer_name,int account_no,String type_acc){
super(customer_name,account_no,type_acc);
}

void applyinterest() {
System.out.println("Enter the interest rate:");
int rate=sc.nextInt();
double interest=balance*rate;
balance+=interest;
System.out.println("Balance after interest:"+balance);
}
}

import java.util.Scanner;
public class Bank {
	public static void main(String[] args) {
		Scanner sc=new Scanner(System.in);
		System.out.println("Enter customer name:");
		String cus_name=sc.nextLine();
		System.out.println("Enter account number:");
		int acc_no=sc.nextInt();

		Account ca=new Cur_acct(cus_name,acc_no,"Current Account");
		Account sa=new Sav_acct(cus_name,acc_no,"Saving Account");

		int choice;
		while(true){
			System.out.println("----MENU----");
			System.out.println("1.Deposite\n2.Withdrawl\n3.Compute intereset for Saving
			account\n4.Display account details\n5.Exit");
			choice=sc.nextInt();

			switch(choice) {
			case 1:
			System.out.println("Enter the type of account:\n1.Saving Account \n2.Current Account");
			int acc=sc.nextInt();
			if(acc==1) {
```

```
                    sa.diposite();
                    }
                    else {
                    ca.diposite();
                    }
                    break;
                    case 2:
                    System.out.println("Enter the type of account:\n1.Saving Account \n2.Current Account");
                    int acc1=sc.nextInt();
                    if(acc1==1) {
                    sa.withdrawal();
                    }
                    else {
                    ca.withdrawal();
                    }
                    break;
                    case 3:
                    sa.applyinterest();
                    break;
                    case 4:
                    System.out.println("Enter the type of account:\n1.Saving Account \n2.Current Account");
                    int acc2=sc.nextInt();
                    if(acc2==1) {
                    sa.display();
                    }
                    else {
                    ca.display();
                    }
                    break;
                    case 5:
                    break;
                    default:
                    System.out.println("Invalid Choice");
                    break;
                    }

                    if(choice==5) {
                    break;
                    }
                }
        }
}
```

LAB: 6

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

```
package CIE;
import java.util.Scanner;
public class Student{
   public String usn,name;
   public int sem;
   public void inputStudentDetails(){
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter the student usn:");
      usn=sc.nextLine();
      System.out.println("Enter the student name:");
      name=sc.nextLine();
      System.out.println("Enter student semester:");
      sem=sc.nextInt();
   }

   public void disply(){
      System.out.println("Student USN:"+usn);
      System.out.println("Student Name:"+name);
      System.out.println("Student Sem:"+sem);
   }
}

package CIE;
import java.util.Scanner;
public class Internals extends Student{
   public int marks[]=new int[5];
   public void inputCIEmarks(){
      Scanner sc=new Scanner(System.in);
      for(int i=0;i<5;i++){
         System.out.println("Enter the marks for subject "+(i+1)+":");
         marks[i]=sc.nextInt();
      }
   }
}
package SEE;
import CIE.Internals;
import java.util.Scanner;
public class Externals extends CIE.Internals{
   public int marks[];
   public int finalMarks[];

   public Externals(){
      marks=new int[5];
      finalMarks=new int[5];
   }
```

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

    public void displayFinalMarks(){
        inputStudentDetails();
        for(int i=0;i<5;i++){
            System.out.println("Subject "+(i+1)+" final marks:"+finalMarks[i]);
        }
    }
}
import SEE.Externals;
class PackageMain{
    public static void main(String args[]){
        int numOfStudent=2;
        Externals finalMarks[]=new Externals[numOfStudent];
        for(int i=0;i<numOfStudent;i++){
            finalMarks[i]=new Externals();
            finalMarks[i].inputStudentDetails();
            System.out.println("Enter CIE Marks:");
            finalMarks[i].inputCIEmarks();
            System.out.println("Enter SEE Marks:");
            finalMarks[i].inputSEEMarks();
        }
        System.out.println("Displaying data:\n");
        for(int i=0;i<numOfStudent;i++){
            finalMarks[i].calculateFinalMarks();
            finalMarks[i].disply();
            finalMarks[i].displayFinalMarks();
        }
    }
}
```

LAB: 7

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "Son" which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age<0. In Son class, implement a constructor that cases both father and son's age and throws an exception if son's age is >=father's age.

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

   }
   void display(){
      System.out.println("Father's age:"+fatherAge);
   }
}
class Son extends Father{
   int sonAge;
   Son() throws WrongAge{
      System.out.println("Enter Son's age:");
      sonAge=sc.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:"+sonAge);
   }

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

```
        try{
            Son son=new Son();
            son.display();
        }
        catch(WrongAge e){
            System.out.println(e);

        }
    }
}
```

LAB: 8

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

```
class BMS_College_of_Engineering implements Runnable {
   public void run() {
      while (true) {
         try {
            System.out.println("BMS College of Engineering");
            Thread.sleep(10000); // Sleep for 10000 milliseconds (10 seconds)
         } catch (InterruptedException e) {
            e.printStackTrace();
         }
      }
   }
}

class CSE implements Runnable {
   public void run() {
      while (true) {
         try {
            System.out.println("CSE");
            Thread.sleep(2000); // Sleep for 2000 milliseconds (2 seconds)
         } catch (InterruptedException e) {
            e.printStackTrace();
         }
      }
   }
}

public class ThreadMain {
   public static void main(String[] args) {
      Thread t1 = new Thread(new BMS_College_of_Engineering());
      Thread t2 = new Thread(new CSE());
      t1.start();
      t2.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 divident:");

      // add text field for both numbers
      JTextField ajtf = new JTextField(8);
      JTextField bjtf = new JTextField(8);

      // calc button
      JButton button = new JButton("Calculate");

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

      // add in order :)
      jfrm.add(err);  // to display error bois
      jfrm.add(jlab);
      jfrm.add(ajtf);
      jfrm.add(bjtf);
      jfrm.add(button);
      jfrm.add(alab);
      jfrm.add(blab);
      jfrm.add(anslab);

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

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

                    alab.setText("\nA = " + a);
                    blab.setText("\nB = " + b);
                    anslab.setText("\nAns = "+ ans);
                }
                catch(NumberFormatException e){
                    alab.setText("");
                    blab.setText("");
                    anslab.setText("");
                    err.setText("Enter Only Integers!");
                }
                catch(ArithmeticException e){
                    alab.setText("");
                    blab.setText("");
                    anslab.setText("");
                    err.setText("B should be NON zero!");
                }
            }
        });

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

    public static void main(String args[]){
        // create frame on event dispatching thread
        SwingUtilities.invokeLater(new Runnable(){
            public void run(){
                new SwingDemo();
            }
        });
    }
}
```

LAB: 10

Demonstrate Inter process Communication and deadlock.

```
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<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();
			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.");
	}
}
```

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 other thread");
        }

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

}
```