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:

RAGHAVENDRA R

1BM22CS214

Department of Computer Science and Engineering,

B.M.S College of Engineering,

Bull Temple Road, Basavanagudi, Bangalore, 560 019

2023-2024.

LAB - 1

Develop a Java program that prints all real solutions to the quadratic equation ax2+bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminate b2 -4ac is negative, display a message stating that there are no real solutions.

```
import java.util.Scanner;

class Quadratic
{
    int a, b, c;
    double r1, r2, d;
    void getd()
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the coefficients of a,b,c");
        a = s.nextInt();
        b = s.nextInt();
        c = s.nextInt();
    }
    void compute()
    {
    while(a==0)
        {
            System.out.println("Not a quadratic equation");
            System.out.println("Enter a non zero value for a:");
            Scanner s = new Scanner(System.in);
            a = s.nextInt();
        }
        d = b*b-4*a*c;
        if(d==0)
        {
            r1 = (-b)/(2*a);
            System.out.println("Roots are real and equal");
            System.out.println("Root1 = Root2 = " + r1);
```

```
        }
        else if(d>0)
        {
            r1 = ((-b)+(Math.sqrt(d)))/(double)(2*a);
            r2 = ((-b)-(Math.sqrt(d)))/(double)(2*a);
            System.out.println("Roots are real and distinct");
            System.out.println("Roo1 = " + r1 + " Root2 = " +
r2);
        }
        else if(d<0)
        {
            System.out.println("Roots are imaginary");
            r1 = (-b)/(2*a);
            r2 = Math.sqrt(-d)/(2*a);
            System.out.println("Root1 = " + r1 + " + i"+r2);
            System.out.println("Root1 = " + r1 + " - i"+r2);
        }
    }
}

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

LAB - 2

Sgpa calculator.

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

```
import java.util.Scanner;

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

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

    void getStudentDetails()
    {
        System.out.println("enter your name : ");
        name = s.nextLine();
        System.out.println("enter your usn : ");
        usn = s.nextLine();
    }
```

```
    void getMarks()
    {
        int i;
        for(i=0;i<8;i++)
        {
            System.out.println("enter the marks and credits for course " +
(i+1) + ":");
            System.out.println("marks : ");
            int marks = s.nextInt();
            System.out.println("credits : ");
            int credit = s.nextInt();
            subject[i].subjectMarks = marks;
            subject[i].credits = credit;

            if(marks >= 90 && marks<=100)
            {
                subject[i].grade = "O";
            }
            else if(marks>=80 && marks<90)
            {
                subject[i].grade = "A+";
            }
            else if(marks>=70 && marks<80)
            {
                subject[i].grade = "A";
            }
            else if(marks>=60 && marks<70)
            {
                subject[i].grade = "B+";
            }
            else if(marks>=50 && marks<60)
            {
                subject[i].grade = "B";
            }
            else if(marks>=40 && marks<50)
            {
                subject[i].grade = "C";
            }
            else if(marks>=0 && marks<40)
            {
                subject[i].grade = "F";
            }
```

```
        }
    }
    void computeSGPA()
    {
        int i;
        double sgpa;
        double totalcredits = 0;
        double totalgradepoints = 0;

        for(i=0;i<8;i++)
        {
            totalcredits += subject[i].credits;
            switch(subject[i].grade)
            {
                case "O" : totalgradepoints += 10*subject[i].credits;
                break;
                case "A+" : totalgradepoints += 9*subject[i].credits;
                break;
                case "A" : totalgradepoints += 8*subject[i].credits;
                break;
                case "B+" : totalgradepoints += 7*subject[i].credits;
                break;
                case "B" : totalgradepoints += 6*subject[i].credits;
                break;
                case "C" : totalgradepoints += 5*subject[i].credits;
                break;
                case "F" : totalgradepoints += 0*subject[i].credits;
                break;
            }
        }
        sgpa = totalgradepoints/totalcredits;
        System.out.println("the sgpa is : "+sgpa);
    }
}
class sgpa
{
    public static void main(String args[])
    {
        Student s1 = new Student();
        s1.getStudentDetails();
        s1.getMarks();
        s1.computeSGPA();
```

```
    }
}
```

```
                                    LAB - 3

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

import java.util.Scanner;

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

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

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

public class Mainbook
```

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

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

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

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

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

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

LAB - 4

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

```
import java.util.Scanner;

class inputScanner
{

    protected Scanner s;

    public inputScanner()
    {
        s = new Scanner(System.in);
    }

    public int getInput(String message)
    {
        System.out.println(message);
        return scanner.nextInt();
    }



}

abstract class Shape extends inputScanner
{
    protected int a,b;

    public Shape()
    {
        super();
    }

    abstract public void printArea();
}

class Rectangle extends Shape
{
```

```
    protected int a,b;
    public Rectangle()
    {
        super();
    }

    public void printArea()
    {

        a=getInput("Enter the length:");
        b=getInput("Enter the bredth:");
        int area= a*b;
        System.out.println("Area of the Rectangle:" +area);
    }
}

class Triangle extends Shape
{
    protected int a,b;
    public Triangle()
    {
        super();
    }

    public void printArea()
    {
        a=getInput("Enter the side1:");
        b=getInput("Enter the side2:");
        double area=0.5*a*b;
        System.out.println("Area of the Triangle:" +area);
    }

}

class Circle extends Shape
{
    protected int a;
    public Circle()
    {
        super();
    }
```

```
    public void printArea()
    {
        a=getInput("Enter the radius:");
        double area=3.14*a*a;
        System.out.println("Area of the Circle:" +area);

    }

}

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

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

LAB - 5

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

a) Accept deposit from customer and update the balance.

b) Display the balance.

c) Compute and deposit interest

d) Permit withdrawal and update the balance Check for the minimum balance, impose penalty if necessary and update the balance.

```
import java.util.Scanner;
class account
{
    String name;
    int accno;
    String type;
    double balance;

    account(String name,int accno,String type,double balance)
    {
        this.name=name;
        this.accno=accno;
        this.type=type;
        this.balance=balance;
    }
    void deposit(double amount)
    {
        balance+=amount;
    }
    void withdraw(double amount)
    {
        if((balance-amount)>=0)
        {
            balance-=amount;
        }
        else
        {
            System.out.println("insufficient balance,cant withdraw");
        }
```

```
    }

    void display()
    {

System.out.println("name:"+name+"accno:"+accno+"type:"+type+"balance:"+balance
);
    }
}
class savAcct extends account
{

    private static double rate=5;
    savAcct(String name,int accno,double balance)
    {
        super(name,accno,"savings",balance);

    }

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


}
class curAcct extends account
{

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

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

    }


    void checkmin()
    {
```

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

    }

}
class accountMain
{
    public static void main(String a[])
    {
        Scanner s=new Scanner(System.in);
        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 accno=s.nextInt();
        System.out.println("enter the intial balance:");
        double balance=s.nextDouble();
        int ch;
        double amount1,amount2;
        account acc=new account(name,accno,type,balance);
        savAcct sa=new savAcct(name,accno,balance);
        curAcct ca=new curAcct(name,accno,balance);
        while(true)
        {
            if(acc.type.equals("savings"))
            {
                System.out.println("\nMenu\n1.deposit 2.withdraw 3.compute
interest 4.display");
                System.out.println("enter the choice:");
                ch=s.nextInt();
                switch(ch)
                {
                    case 1:System.out.println("enter the amount:");
                        amount1=s.nextInt();
```

```
                        sa.deposit(amount1);
                        break;
                    case 2:System.out.println("enter the amount:");
                        amount2=s.nextInt();
                        sa.withdraw(amount2);
                        break;
                    case 3:sa.interest();
                        break;
                    case 4:sa.display();
                        break;
                    case 5:System.exit(0);
                    default:System.out.println("invalid input");
                        break;
                }
            }
            else
            {
                System.out.println("\nMenu\n1.deposit 2.withdraw  3.display");
                System.out.println("enter the choice:");
                ch=s.nextInt();
                switch(ch)
                {
                    case 1:System.out.println("enter the amount:");
                        amount1=s.nextInt();
                        ca.deposit(amount1);
                        break;
                    case 2:System.out.println("enter the amount:");
                        amount2=s.nextInt();
                        ca.withdraw(amount2);
                        ca.checkmin();
                        break;

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

LAB - 6

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

```
// Internals.java
package CIE;

import java.util.Scanner;

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



    public void inputCIEmarks() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter Internal Marks for " + name);
        for (int i = 0; i < 5; i++) {
            System.out.print("Subject " + (i + 1) + " marks: ");
            marks[i] = scanner.nextInt();
        }
    }
}
```

```
// 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 scanner = new Scanner(System.in);
        System.out.print("Enter USN: ");
        usn = scanner.next();
        System.out.print("Enter Name: ");
        name = scanner.next();
        System.out.print("Enter Semester: ");
        sem = scanner.nextInt();
    }

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

```
// 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() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter SEE Marks for " + name);
        for (int i = 0; i < 5; i++) {
```

```
            System.out.print("Subject " + (i + 1) + " marks: ");
            marks[i] = scanner.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]);
    }
}
```

LAB - 7

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

```
import java.util.Scanner;

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

class Father
{
    protected int fatAge;
    public Father() throws WrongAge
    {
        Scanner s = new Scanner(System.in);
        System.out.println("enter father's age : ");
        fatAge = s.nextInt();
        if(fatAge < 0)
            throw new WrongAge("Age cannot be negative");
    }
    public void displayfat()
    {
            System.out.println("father's age is : "+fatAge);
    }
}

class Son extends Father
{
    private int sonAge;
    public Son() throws WrongAge
    {
        super();
```

```
        Scanner s = new Scanner(System.in);
        System.out.println("enter the son's age : ");
        sonAge = s.nextInt();
        if(sonAge >= fatAge)
        {
            throw new WrongAge("son's age is more than or equal to father's
age");
        }
        else if(sonAge<0)
        {
            throw new WrongAge("age cannot be negative");
        }
    }
    public void display()
    {
        System.out.println("son's age is : "+sonAge);
    }
}

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

LAB - 8

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

```
class BMSThread extends Thread {
    @Override
    public void run() {
        while(true) {
            System.out.println("BMS college of engineering");
            try {
                Thread.sleep(10000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

class CSEThread extends Thread {
    @Override
    public void run() {
        while(true) {
            System.out.println("CSE");
            try {
                Thread.sleep(2000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

public class threadEx {
    public static void main(String[] args) {
        BMSThread bms = new BMSThread();
        bms.start();
        CSEThread cse = new CSEThread();
        cse.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 UserInterface {
    UserInterface() {
        // 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 UserInterface();
            }
        });
    }
}
```

LAB - 10

Demonstrate Inter process Communication and deadlock

IPC

```
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<2) {
    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();
    }
}
```

12-12-23

classmate
Date
Page 12/12/23

1)
```
class helloworld {
    public static void main(String [] args) {
        System.out.println("hello world!");
    }
}
```

o/p: hello world!

2)
```
class rectangleArea {
    public static void main(String [] args) {
        int length, Breadth;
        length = Integer.parseInt(args[0]);
        Breadth = Integer.parseInt(args[1]);
        int area = length * Breadth;
        System.out.println("length = " + length);
        System.out.println("Breadth = " + Breadth);
        System.out.println("Area = " + Area);
    }
}
```

o/p: length = 10
Breadth = 5
Area = 50

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

6.

```
import java.util.*;

class palindrome
{
    public static void main(String[] args)
    {
        int n, t, sum, rev = 0;
        Scanner s = new Scanner(System.in);
        System.out.println("Enter a 5 digit number");
        n = s.nextInt();
        t = n;
        while (t > 0)
        {
            sum = t%10
            rev = rev*10 + sum;
            t = t/10;
        }
        if (rev == n)
        {
            System.out.println("It is a palindrome");
        }
        else
        {
            System.out.println("It is not a palindrome");
        }
```

Op: Enter a 5 digit number:
12321
It is a palindrome.

4)

```
import java.util.*;

class factorial
{
    public static void main(String[]args)
    {
        int fac = 1;
        System.out.println("Enter a number");
        Scanner s = new Scanner(System.in);
        int n = s.nextInt();
        for(int i=0; i<=n; i++)
        {
            fac = fac * i;
        }
        System.out.println("The factorial is " + fac);
```

o/p:

```
Enter a number
5
The factorial is 120
```

5)

```
class Array
{
    public static void main(String [] args)
    {
        int month[] = {31,28,31,30,31,30,31,31,30,31,30,31};
        System.out.println("April has " + math[3] + "days");
    }
}
```

o/p:

```
April has 30 days.
```

Date 12/12/23
Page

LAB - 1 PROGRAM:

1. // Quadratic Equation.

```
import java.util.*;
class Quadratic
{
    int a,b,c;
    double r1,r2,d;
    void getd()
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the value of a, b & c");
        a = s.nextInt();
        b = s.nextInt();
        c = s.nextInt();
    }
    void compute()
    {
        while(d==0)
        {
            System.out.println("not a quadratic function");
            System.out.println("Enter a value of a > 0");
            a = s.nextInt();
            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 & 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("root 1 = " + r1 + "root 2 = " + r2);
        }
        else if(d<0)
        {
            System.out.println("roots are imaginary");
            r1 = (-b)/(2*a);
            r2 = Math.sqrt(-d)/(2*a);
            System.out.println("root 1 = " + r1 + " + i " + r2);
            System.out.println("root2 = " + r2 + " - i " + r2);
        }
    }
}

class QuadraticMain
{
    public static void main(String[] args)
    {
        Quadratic q = new Quadratic();
        q.getd();
        q.compute();
        System.out.println("1BM22CS214, Raghavendra R");
    }
}
```

output:

```
Enter the coefficient of a, b, c:
1 2 1
roots are real & Equal.
root 1 = root 2 = -1.0.
```

iii) Enter the values of a, b & c:
2 4 5.
roots are imaginary
root1 = -1.0 + i1.2247448713915 89
root2 = -1.0 - i1.2247448 713915 89

iii) Enter the values of a, b & c:
-1 4 1
roots are real & distinct.
root1 = -0.2679491924311228
root 2 = -3.7320508075688772

iv) Enter the values of a, b & c:
0 2 1
Not a quadratic fuction
Enter a value of a > 0.
1
roots are real & Equal
root1 = root2 = -1.0

1BM22CS214, Raghavendra.R

LAB - 2 :

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

```
import java.util.Scanner;

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

class Student
{
    String name;
    String usn;
    double SGPA;
    Scanner s;
    Subject subject[];
    Student()
    {
        int i;
        subject = new Subject[9];
        for(i=0; i<9; i++)
        {
            Subject[i] = new Subject();
        }
        s = new Scanner(System.in);
    }
```

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

void getMarks()
{
    int i;
    for(i=0; i<8; i++)
    {
        System.out.println("Enter the marks and
        credits for course "+(i+1)+" : ");
        System.out.println("marks : ");
        int marks = s.nextInt();
        System.out.println("credits : ");
        int credits = s.nextInt();

        if(marks >= 90 && marks <= 100)
        {
            subject[i].grade = "O";
        }
        else if(marks >= 80 && marks < 90)
        {
            subject[i].grade = "A+";
        }
        else if (marks >= 70 && marks < 80)
        {
            subject[i].grade = "A";
        }
```

```
        else if(marks >=60 && marks < 70)
        {
            Subject[i].grade = "B+";
        }
        else if (marks >=50 && marks <60)
        {
            Subjelt[i].grade = "B";
        }
        else if (marks >=40 && marks <50)
        {
            Subject[i].grade = "C";
        }
        else if (marks >=0 && marks <40)
        {
            Subject[i].grade = "F";
        }
    }
}
void computeSGPA()
{
    int i;
    double sgpa;
    double totalcredits = 0;
    double totalgradepoints = 0;
    for(i=0; i<8; i++)
    {
        totalcredits += Subject[i].credits;
        switch(subject[i].grade)
        {
            case "O": totalgradepoints += 10*Student[i].credits;
            break;
            case "A+": totalgradepoints += 9*Students[i].credits;
            break;
```

```
            case "A": totalgradepoints += 8*student[i].credits;
            break;
            case "B+": totalgradepoints += 7*student[i].credits;
            break;
            case "B": totalgradepoints += 6*student[i].credits;
            break;
            case "C": totalgradepoints += 5*student[i].credits;
            break;
            case "F": totalgradepoints += 4*student[i].credits;
            break;
        }
    }
    sgpa = totalgradepoints/totalcredits;
    System.out.println("name: "+name);
    System.out.println("USN:" + usn);
    System.out.println("SGPA:" + sgpa);
  }
}

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

OUTPUT:

name: Raghavendra R
USN : 1BM22CS214

Enter marks for course 1:
marks: 90
credits: 4

Enter marks and credits for course 2:
marks: 91
credits: 4

Enter marks and credits for course 3:
marks: 99
credits: 3

Enter marks and credits for course 4:
marks: 94
credits: 3

Enter marks and credits for course 5:
marks: 95
credits: 2

Enter marks and credits for course 6:
marks: 96
credits: 01

Enter marks and credits for course 7:
mark: 97
credits: 01

14/12/23

The SGPA is: 10.000000

LAB - 3.

Page No.
Date 26/12/23

Create a class Book which contains objects & constructor & methods. Develop a Java program to create n book objects.

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

    Books(String name, String author, int price, int numPages)
    {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }
public String toString()
{
    String name, author, price, numPages;
    name = "Book name : " + this.name + "\n";
    author = "author name : " + this.author + "\n";
    price = " price : " + this.price + "\n";
    numPages = "number of pages : " + this.numPages + "\n";
    return name + author + price + numpages;
}
}
```

```
public class Mainbooks
{
    public static void main(String[] args)
    {
        Scanner s = new Scanner(System.in);
        int n;
        int i;
        String name;
        String authan;
        int price;
        int numPages;
        System.out.println("Enter number of books : ");
        n = s.nextInt();
        Books b[];
        b = new Books[n];

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

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

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

OUTPUT:

```
Enter the number of books: 2
Enter the details of book 1:
Enter the name of the book: javaCompleteReference
Enter the author name: Herbert
Enter the price: 1000
Enter the number of pages: 1034

Enter the details of book 2:
Enter the name of the book: Ramayana
Enter the author name: valmiki
Enter the price: 1200
Enter the number of pages: 1200.

Book details:
Book name: JavaCompliteRifrence
author name: hurbert
price: 1000
number of pages: 1034

Book name: Ramayana
author name: valmiki
price: 1200
number of pages: 1200
```

26/12/23

LAB-4

Page No.
Date 2/01/24

Q: Abstract class for shapes (rectangle, triangle & circle).

```
import java.util.Scanner;

class InputScanner
{
    protected Scanner s;
    Public InputScanner()
    {
        s = new Scanner(System.in);
    }
    Public int getInput(String message)
    {
        System.out.println(message);
        return s.nextInt();
    }

abstract class Shape extends InputScanner
{
    protected int a, b;
    Public Shape()
    {
        super();
    }
    abstract public void printArea();
}

class Rectangle extends Shape
{
    protected int a, b;
    Public Rectangle()
    {
        Super();
    }
```

```
    Public void printArea()
    {
        a = getInput("Enter the length of :");
        b = getInput("Enter the breadth : ");
        int area = a * b;
        System.out.println("Area of rectangle:" + area);
    }
}

class Triangle extends Shape
{
    protected int a,b;
    public Triangle()
    {
        super();
    }
    public void printArea()
    {
        a = getInput("Enter the Side1 : ");
        b = getInput("Enter the Side2 : ");
        double area = 0.5 * a * b;
        System.out.println("Area of Triangle : " + area);
    }
}

class Circle extends Shape
{
    protected int a;
    public circle()
    {
        super();
    }
```

```
    public void printArea()
    {
        a = getInput("Enter the radius: ");
        double area = 3.14*a*a;
        System.out.println("Area of circle: "+area);
    }
}

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

OUTPUT:

```
Enter the length : 3
Enter the breadth : 4
Area of rectangle : 12
Enter the side1 : 5
Enter the side2 : 6
Area of Triangle : 30
Enter the radius : 2
Area of circle : 12.6
```

02.1.24

Java : Lab - 6 16/01/24

Question 1 :
type 1 : BMSCE
type 2 : BMSCE
type 3 : BMSCE
type 4 : MS
type 5 : abcd

Question 2 :
length of s1 : 5
Concatenation of s1 & s2 : BMSCEBMSCE

Question 3 :
hashString() : 20

Question 5 :
65 66 67 68 69 70 71
B M S C E

Question 4 :
* The given string is : welcome to bmsce college.
* The srcBegin, srcEnd, and dstBegin values are :
11, 16 & 0
* The value of character array :
[b, m, s, c, e, , , , , ,].

Question 6 :
Bmsce equals Bmsce → true
Bmsce equals college → false
Bmsce. equalsIgnoreCase BMSCE → true

Question 7:
Substring & matched
S1 = "BMSCE college";
S2 = "welcome to BMSCE college of engineering";

Question 8:
true
False

Question 9:
False
True.

Question 10:
Hello equals Hello → True
Hello == Hello → False

Question 11:
The names in alphabetical order are:
apple
ball
cat
van
watch

Question 12:
Sorted numbers: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10].

Question 13:
Thwas was a test Thwas was, bob.

Question 14 :
hello world

Question 15 : :
commege.

question 16 :
befor trim: Hello friends
after trim : Hello friends

R:0 | Page No.
Date 23/01/24

LAB - 7

// 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 scanner = new Scanner(System.in);
        System.out.println("ENTER USN :");
        usn = scanner.nextInt();
        System.out.println("Enter name :");
        name = scanner.next();
        System.out.println("Enter semester :");
        sem = Scanner.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 Students {
    protected int marks[] = new int[5];
    public void inputElements() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter internal marks for "+name);
        for(int i=0; i<5; i++) {
            System.out.println("subject " + (i+1) + " marks:");
            marks[i] = s.nextInt();
        }
    }
}
```

// Externals.java

```
package SEE;
import CIE.Internals;
import java.util.Scanner;

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

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

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

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

```
import SEE.Externals;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        int num = 2;
        Externals finalMarks[] = new Externals[num];

        for(int i=0; i< num; i++) {
            finalMarks[i] = new Externals();
            finalMarks[i].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<num; i++) {
            finalMarks[i].calculateFinalMarks();
            finalMarks[i].displayFindMarks();
        }
    }
}
```

OUTPUT:

Enter USN : 777 Enter Name : thala Enter sem : 7
Enter CIE marks :
Enter Internal Marks for "thala"
Subject 1 marks : 45
Subject 2 marks : 46
Subject 3 marks : 47
Subject 4 marks : 48
Subject 5 marks : 50
Enter SEE marks :
Subject 1 marks : 90
Subject 2 marks : 92
Subject 3 marks : 95
Subject 4 marks : 98
Subject 5 marks : 100

Enter USN : 18 Enter name : Kohli Enter Sem : 8
Enter CIE marks :
Enter Intimal marks for "Kohli" :
Subject 1 marks : 45
Subject 2 marks : 47
Subject 3 marks : 48
Subject 4 marks : 49
Subject 5 marks : 50
Enter SEE marks :
Subject 1 marks : 98
Subject 2 marks : 97
Subject 3 marks : 96
Subject 4 marks : 95
Subject 5 marks : 100

USN : 777 Name : thala Semester : 7

Subject 1 : 90

Subject 2 : 92

Subject 3 : 94

Subject 4 : 97

Subject 5 : 100

~~Subject~~ 6

USN : 18 name : kohli semester : 8

Subject 1 : 94

Subject 2 : 95

Subject 3 : 96

Subject 4 : 96

Subject 5 : 100

2/8/24

LAB - 10

Page No.
Date 13/02/24

```
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(" caught ");
            }
            System.out.println("Got: " + n);
            valueset = false;
            System.out.println("\n Inform 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(" caught ");
    }
    this.n = n;
    valueset = true;
    System.out.println("Put: " + n);
    System.out.println("\n Inform consumer\n");
    notify();
    }
}
```

```
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 < 2) {
            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.");
    }
}
```

Put : 0
Intimate consumer
producer waiting
Got : 0
Intimate producer

Put : 1
Intimate Consumer
Consumer : 0
Got : 1
Intimate producer
Consumer : 1
consumer waiting

LAB-10:

// Deadlock.

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

O/P:

```
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.
```

Lab - 9.

Page No.
Date 20/2/24

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

class UserInterface {
    UserInterface() {
        // create jframe container
        JFrame jfrm = new JFrame("Dividin 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 dividen & 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);
```

```
jfram.add(jlab);
jfram.add(ajtf);
jfram.add(bjtf);
jfram.add(button);
jfram.add(alab);
jfram.add(blab);
jfram.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("\n A = ", a);
            blab.setText("\n B = ", b);
            anslab.setText("\n Ans = " + ans);
        } catch (NumberFormatException e) {
            alab.setText(" ");
            blab.setText(" ");
            anslab.setText(" ");
            jfram.setText("Enter only integers");
        }
```

```
            catch(ArithmeticException e) {
                alab.setText(" ");
                blab.setText(" ");
                ans lab.setText(" ");
                ena.setText("B should be Non zero!");
            }
        }
    });

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

Public static void main(String[] args) {
    // create frame on event dipatching thread
    Swing Utilities.invokeLater(new Runnable() {
        Public run() {
            new UserInterface();
        }
    });
}
}
```

output:

Enter the divider & divident:

6 3

Calculate : A = 6 ; B = 3 Ans = 2

LAB - 8:

Page No.
Date 06 / 02 / 2024

```
class BMSThread extends Thread {
    @override
    public void run() {
        while (true) {
            System.out.println("BMS college");
            try {
                Thread.sleep(10000);
            } catch (InterruptedException e){
                e.printStackTrace();
            }
        }
    }
}

class CSEThread extends Thread {
    @Override
    public void run() {
        while(true) {
            System.out.println("CSE");
            try {
                Thread.sleep(2000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}
```

```
Public class threadEx {
    public static void main(String[] args) {
        BMSthread bms = new BMSThread();
        bms.start();
        CSEThread cse = new CSEThread();
        cse.thread();
    }
}
```

o/p:

BMS college
CSE
CSE
CSE
CSE
CSE
BMS college
CSE
CSE
CSE
CSE
CSE
BMS college
CSE

8/02/24

WEEK - 7.

```
import java.util.*;

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

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

class Father extends InputScanner{
    protected int fatherAge;
    public Father() throws WrongAge{
        System.out.println("Enter Father's age : ");
        fatherAge = s.nextInt();
        if(fatherAge < 0){
            throw new WrongAge("Age can't be negative");
        }
    }
    public void display(){
        System.out.println("Father's Age : " + fatherAge);
    }
}
```

```
class Son extends Father {
    private int sonAge;
    public Son() throws WrongAge {
        super();
        System.out.println("Enter Son's age : ");
        sonAge = s.nextInt();
        if (sonAge > fatherAge) {
            throw new exception("Son's age cannot
            be greater than father's age");
        else if (sonAge < 0) {
            throw new Exception ("Age can't be negative");
        }
    }
    public void display() {
        super.display();
        System.out.println("Son's age : " + sonAge);
    }
}

Public class FatherSonAge {
    Public static void main(String[] Args) {
        try {
            Son son = new Son();
            son.display();
        } catch (WrongAge e) {
            System.out.println("Error : " + e.getMessage());
        }
    }
}
```

O/P:

* Enter ~~Son~~ Father's age: 40
Enter Son's age: 20
Father's age: 40
Son's age: 20

* Enter Father's age: 10
Enter Son's age: 20
Error, Son's age cannot be greater than father's age.

* Enter father's age: -10
Error: age cannot be negative

30/01/24