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:

Rachana H D

1BM22CS212

Department of Computer Science and Engineering,

B.M.S College of Engineering,

Bull Temple Road, Basavanagudi, Bangalore, 560 019

2023-2024.

INDEX

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

LAB 1

Develop a Java program that prints all real solutions to the quadratic equation

ax 2 +bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminant b 2 -4ac**

is negative, display a message stating that there are no real solutions.

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

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

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

```
                q.getd();
                q.compute();
        }
}
```

LAB 2

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

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

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

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

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

```
                    sub[i].grade=0;
                numerator= sub[i].credits * sub[i].grade;
                n_sum = n_sum + numerator;
                d_sum = d_sum + sub[i].credits;
            }
        }

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

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

LAB 3

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

complete details of the book. Develop a Java program to create n book objects.

```
import java.util.Scanner;
class Books
{
	String title;
	String author;
	int price;
	int num_pages;

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

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

class Main
```

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

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

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

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

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

```
}}
```

LAB 4

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

```
import java.util.Scanner;

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

class Rectangle extends Shape
{
	void input()
	{
		System.out.println("Enter length and breadth:");
		dim1=s.nextInt();
		dim2=s.nextInt();
	}
```

```
    void printArea()
    {
        System.out.println("Area of rectangle= "+(dim1*dim2)+ " sq units");
    }
}

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

class Circle extends Shape
{
    void input()
    {
        System.out.println("Enter radius:");
        dim1=s.nextInt();
    }
    void printArea()
    {
```

```
                System.out.println("Area of circle= "+(3.14*dim1*dim1)+ " sq units");
        }
}

class Area
{
        public static void main (String args[])
        {
                Rectangle r= new Rectangle();
                Triangle t= new Triangle();
                Circle c= new Circle();
                Shape ref;
                ref=r;
                ref.input();
                ref.printArea();
                ref=t;
                ref.input();
                ref.printArea();
                ref=c;
                ref.input();
                ref.printArea();
        }
}
```

LAB 5

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

Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

a) Accept deposit from customer and update the balance.

b) Display the balance.

c) Compute and deposit interest

d) Permit withdrawal and update the balance

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

```
import java.util.Scanner;


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

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

```
                System.out.println("Balance = "+bal);
        }
}

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

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

```
}

class Account
{
	public static void main(String args[])
	{
		Scanner sc=new Scanner(System.in);
		int ch, type;
		Bank b=new Bank();
		b.get();
		Sav_acct a=new Sav_acct();
		Cur_acct c=new Cur_acct();
		System.out.println("Enter the account type (1.savings/2.current):");
		type=sc.nextInt();
		if(type==1)
		{
		 do
		{
		    System.out.println("-----Menu --- ");
		    System.out.println("1:Deposit  2:Withdrawal  3.Compound interest  4.Exit");
		    System.out.println("Enter your choice:");
		    ch=sc.nextInt();
		    switch(ch)
		    {
			case 1:
				a.deposit2();
				break;
			case 2:
				a.withdrawal();
```

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

```
                }
        }
}
```

LAB 6

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

```
// Package CIE
package CIE;
import java.util.Scanner;
public class student
{
        protected String usn = new String();
        protected String name = new String();
        protected int sem;
        public void inputDetails()
        {
                Scanner s=new Scanner(System.in);
                System.out.println("Enter student usn:");
                usn = s.next();
                System.out.println("Enter student name:");
                name = s.next();
                System.out.println("Enter student sem:");
```

```
                sem = s.nextInt();
        }

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

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

// Package SEE
package SEE;
import CIE.internals;
```

```
import java.util.Scanner;
public class externals extends internals
{
	protected int marks[];
	protected int finalMarks[];
	public externals()
	{
		marks = new int[5];
		finalMarks = new int[5];
	}
	public void SEEMarks()
	{
		Scanner s= new Scanner(System.in);
		for(int i=0;i<5;i++)
		{
			System.out.println("Subject "+(i+1)+" marks:");
			marks[i] = s.nextInt();
		}
	}

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

	public void displayFinalMarks()
	{
		for(int i=0;i<5;i++)
```

```
                System.out.println("Subject "+(i+1)+": "+finalMarks[i]);
        }
}

// Class Main
import SEE.externals;
class Main
{
        public static void main(String args[])
        {
                int num = 2;
                externals finalMarks[] = new externals[num];
                for(int i=0;i<num;i++)
                {
                        finalMarks[i] = new externals();
                        finalMarks[i].inputDetails();
                        System.out.println("Enter CIE marks:");
                        finalMarks[i].CIEmarks();
                        System.out.println("Enter SEE marks:");
                        finalMarks[i].SEEMarks();
                }
                System.out.println("Displaying data:\n");
                for (int i=0;i<num;i++)
                {
                        finalMarks[i].calMarks();
                        finalMarks[i].displayDetails();
                        finalMarks[i].displayFinalMarks();
                }
        }
```

LAB 7

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

```
import java.util.*;
class wrongAge extends Exception
{
        wrongAge(String s)
        {
                super(s);
        }

}

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



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

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

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

LAB 8

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

```
import java.util.*;
class NewThread implements Runnable
{
String name;
Thread t;
NewThread(String threadname)
{
```

```
        name = threadname;
        t = new Thread(this , name);
        t.start();
}

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

class MainThread
{
public static void main(String args[])
{
        new NewThread("Computer Science Engineering");
        try
        {
                while(true){
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000);
                System.out.print("\n");
```

```
        }
        }
        catch (InterruptedException e) {
        System.out.println("BMS Interrupted");
        }

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

}

}
```

LAB 9

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

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

class SwingDemo{
  SwingDemo(){
    // create jframe container
    JFrame jfrm = new JFrame("Divider App");
    jfrm.setSize(275, 150);
    jfrm.setLayout(new FlowLayout());
    // to terminate on close
    jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    // text label
    JLabel jlab = new JLabel("Enter the divider and divident:");
```

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

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

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

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

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

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

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

```
            }
            catch(ArithmeticException e){
               alab.setText("");
               blab.setText("");
               anslab.setText("");
               err.setText("B should be NON zero!");
            }
         }
      });

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

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

LAB 10

Demonstrate Inter process Communication and deadlock

```
// InterProcess Communication

class Q

{

        int n;

        boolean valueSet = false;

        synchronized int get()

        {

                while(!valueSet)

                try {

                System.out.println("Consumer waiting");

                wait();
```

```
        } catch(InterruptedException e) {
        System.out.println("InterruptedException caught");
        }
        System.out.println("Got: " + n);
        valueSet = false;
        System.out.println("Intimate Producer");
        notify();
        return n;
        }

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

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

```
        this.q = q;
        new Thread(this, "Producer").start();
        }
        public void run() {
        int i = 0;
        while(i<5) {
        q.put(i++);
        }
        }
}

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

class PCFixed {
        public static void main(String args[]) {
        Q q = new Q();
```

```
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-C to stop.");
        }
}
// 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);
System.out.println("Back in mainthread");
}
public void run() {
b.bar(a);
System.out.println("Back in other thread");
}
public static void main(String args[]) {
new Deadlock();
}
}
```

quadratic

```
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 coefficient
                                of a, b, c");
        a = s.nextInt();
        b = s.nextInt();
        c = s.nextInt();
    }
    void compute()
    {
        while(a==0)
        {
            System.out.println("Not a
                    quadratic equation");
            System.out.println("Enter a
                    non zero value for
                    a...");
            Scanner s = new Scanner(System.in);
            a = s.nextInt();
        }
        d = b*b - 4*a*c;
        if(d==0)
        {
```

```
r1 = (-b)/(2*a);
System.out.println(" Roots are real
                    and equal");
System.out.println(" "Root1 = Root2 = "+r1);
}
else if(d>0)
{
r1 = ((-b) + (Math.sqrt(d)))/(double)(2*a);
r2 = ((-b) - (Math.sqrt(d)))/(double)(2*a);
System.out.println("Roots are real and
                    distinct");
System.out.println("Root1 =" + r1 + "Root2"
                    + r2);
}
else if(d<0)
{
System.out.println("Roots are 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 QuadraticMain
{
public static void main(String args[])
{
Quadratic q = new Quadratic();
q.getd();
q.compute();
}
}
```

output : Enter the coefficients of a, b, c.

1

-2

1

Roots are real and equal

root1 = root2 = 1

20/20

```
import java.util.Scanner;

class subject
{
    int Marks;
    int credits;
    int Grades;
}

class student
{
    String name;
    String USN;
    double SGPA;
    Scanner s;

    Student(){
        int i;
        subjects = new subject[8];
        for(i=0; i<8; i++)
            subjects[i] = new subject();

        s = new Scanner(System.in);
    }
    void getStudentDetails() {
    System.out.print("Enter Student Name:");
    name = s.next();
    System.out.print("Enter USN: ");
    USN = s.next();
    }
```

```
void getMarks() {

for (int i=0; i<8; i++)
{
  System.out.println("Enter details of
                      subject");
System.out.println("Enter Marks:");
Subjects[i].SubjectMarks = s.nextInt();
System.out.print("enter credits:");
Subjects[i].credits = s:nextInt();
if (subjects[i].SubjectMarks>=90)
{
   subjects[i].grade = 10;
}
else if (subjects[i].subjectmarks>=80)
{
   subjects[i]·grade = 9;
}
else if (subjects[i].subjectmarks>=70)
{
   subjects[i].grade = 8;
}
else if (subjects[i].subjectmarks>=60)
{  subjects[i].grade = 7;
}
else if (subjects[i].subjectmarks>=50
{
  subjects[i].grade = 6;
}
```

```
else if (subjects[i].subjectmarks >= 40)
{
  subjects[i].grade = 5;
}
else {
  subjects[i].grade = 0;
}
}
}

void computeSGPA()
{
  double totalcredits = 0;
  double weightedsum = 0;

  for(int i=0; i<8; i++)
{
  totalcredits += subjects[i].credits;
  weightedsum += subjects[i].grade
                 * subjects[i].credits;
}

  SGPA = weightedsum / totalcredits
}

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

```
public class studentMain {

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

output.

Enter name
Rachana

Enter USN
212

Enter ~~credits~~ Marks
94
98
100
100
89
89
100
100

PAGE NO.
DATE

Enter Credits
4
4
3
3
3
1
1
1

SGPA = 9.8 .

PAGE NO:
DATE:

Lab program No 3

1) 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.*;

class Books
{
 int price;
 int numpages
 String name;
 String author;
 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;
  String author;
  String price;
  String numpages;
  name = "BOOK name: " + this.name + "\n
  author = "Author name: " + this.author + "\n";
  price = "price" + this.price + "\n";
  numpages = "Numbers of Page" + this.numpages
  return name + author + price + numpages + "\n";
 }

}
```

```
class main
{
  public static void main (String args
  {
    Scanner s = new Scanner (System.in
    int n;
    String name;
    string author;
    int price;
    int numpages;
    System.out.println ("enter no of bo
    n = S.nextInt();
    Books b[];
    b = new Books[n];

    for (int i=0; i<n; i++)
    {
    System.out.println ("Enter name,
                          author, price,
                          no of pages");
    name = s.next();
    author = s.next();
    price = s.nextInt();
    numpages = s.nextInt();
    b[i] = New Books (name, author,
                          price, numpage
    }
    for (int i=0; i<n; i++)
    {
    System.out.println (b[i]);
    }
}
```

PAGE NO
DATE

output :-

enter no of books
2

Enter name, author, price, numbers of ages
ikigai yen 100 200
C reema 200 300

Book name: ikigai
Author name: yen
price : 100
Number of pages : 80

Book name: C
Author name: reema
price : 200
Number of pages : 300

26/12/23

lab - 4.

```
import java.util.scanner;

abstract class shape
{
    int a, b;
    abstract void printarea();
    abstract void input();
    Scanner s = new Scanner(System.in)
}

class rectangle extends shape
{
    void input()
    {
        System.out.println("Enter l and b");
        a = s.nextInt();
        b = s.nextInt();
    }

    void printarea()
    {
        System.out.println("Area of rectangle
                is: " + (a*b) + " sq units");
    }
}
```

```
class Triangle extends shape.
{
    void input()
    {
        System.out.println ("Enter b and h");
        a = s.nextInt();
        b = s.nextInt();
    }
    void printarea()
    {
        System.out.println("Area of Triangle is:"
        +((a*b)/2) + "sq units");
    }
}

class circle extends shape
{
    void input()
    {
        println("enter the radius of circle");
        a=S.nextInt();
    }
    void printarea()
    {
        System.out.println("Area of circle is"
        + (3.14)*a*a) + "sq units");
    }
}
```

```
class area
{
  public static void main (String args[])
  {
    rectangle r = new rectangle();
    r.input();
    r.printarea();

    triangle t = new triangle();
    t.input();
    t.printarea();

    Circle c = new circle();
    c.input();
    c.printarea();

  }
}
```

output

Enter l and b

4

5

The area of triangle is : 20 sq units.

Enter b and h

4 5

The area of triangle is : 10 sq units

Enter the radius

2

The area of circle is : 12.56 sq units.

PAGE NO
DATE

Lab - 5:

Develop a Java program to create a class Bank that maintains 2 kinds of account-for its customer.

```
import java.util.scanner;

class Account
{
  string customerName;
  long accountNumber;
  String accountType;
  double balance;

  public Account (String customerName,
          long accountNumber, String accountTyp
            double balance)

  {
    this.customerName = customerName.
    this.accountNumber = accountNumber;
    this.accountType = accountType;
    this.balance = balance;
  }
public void deposit (double amount)
{
  balance += amount;
  System.out.println ("Deposit
          successful. updated balance:"+
                balance);

}
```

```
public void displayBalance()
{ System.out.println("Deposit Successful.
                      updated balance: "+bala
}
{
    System.out.println("Account Balance: "+bala
}
}

class currAcct extends Account
{
    double minBalance;
    double serviceCharge;
    public currAcct(String customerName,
                    long accountNumber, double bala
super(customerName, accountNumber, "current", balance
this.minBalance = 100;
this.serviceCharge = 50;
}
```

```
private void checkMinBalance()
{
  if (balance < minBalance)
  {
    balance -= serviceCharge;
    System.out.println("service charges
                        imposed. updated
                        balance: " + balance);
  }
}

public void withdraw(double amount)
{
  if (amount <= balance)
  {
    balance -= amount;
    System.out.println("withdrawal
                        successful. updated
                        balance: " + balance);
    checkMinBalance();
  }
  else
  {
    System.out.println("Insufficient
                        funds for withdrawal.");
  }
}

}
```

```
class SavAcct extends Account
{
    double interestRate;

    public SavAcct(String customerName,
                   long accountNumber, double balance)
    {
        super(customerName, accountNumber,
                   "saving", balance);
        this.interestRate = 0.05;
    }

    public void depositInterest()
    {
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest deposited
                   updated balance: " + balance);
    }

    public void withdraw(double amount)
    {
        if (amount <= balance)
        {
            balance -= amount;
            System.out.println("withdrawal successful
                   updated balance: " + balance);
        }
        else
        {
            System.out.println("Insufficient funds
                   for withdrawal");
        }
    }
}
```

```
public class Bank
{
    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);

        CurrAcct currentAccount = new currAcct
                ("John Doe", 1234, 1500);
        CurrentAccount.deposit(500);
        currentAccount.displayBalance();
        currentAccount.withdraw(1000);

        SavAcct SavingsAccount = new SavAcct
                ("John Doe", 9345, 2000);
        SavingsAccount.deposit(1000);
        SavingsAccount.displaybalance();
        SavingsAccount.depositInterest();
        SavingAccount.displayBalance();
        savingsAccount.withdraw(500);

        scanner.close();
    }
}
```

PAGE NO:
DATE

output:

Deposit successful
update balance : 2000

Account Balance : 2000
withdrawal succesful

update balance : 1000

Deposit successful

9/1/24

PAGE NO
DATE

Week - 6

Student.java.

```
package cie;

import java.util.Scanner;
public class student
{
    protected String usn = new String();
    protected String name = new String();
    protected int sem;

    public void inputstudentdetails()
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the Us of
                                student");
        usn = s.next();
        System.out.println("enter the name of the
                                student");
        name = s.next();
        System.out.println("enter the sem in
        which student is studying");
        sem = s.nextInt();
    }

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

Internals.java

```
package cie;
import java.util.Scanner;
public class internals extends Student
{
    protected int marks[] = new int[5];

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

Externals.java

```
package see;

import cie.internal
import java.util.Scanner;

public class externals extends internals
{
    protected int marks[];
    protected int finalmarks[];
```

```
public external ()
{
    marks = new int[5];
    finalmarks = new int[5];
}
public void inputsubmarks()
{
    Scanner s = new Scanner (System.in
    for (i=0; i<5; i++)
    {
        system.out.print ("subject" + (i+1)
                          + "marks ");
        marks[i] = s.nextInt();
    }
}
public void calculatefinalmarks()
{
    for (int i=0; i<5; i++)
        finalmarks[i] = marks[i]/2 +
                        super.marks(i);
}
public void displayfinalmarks()
{
    displayStudentDetails()
    for (int i=0; i<5; i++)
        system.out.println("subject" + (i+1) +
                           finalmarks[i]
    }
}
```

Main.java

```
import see.externals;
class main
{
  public static void main (String args[])
  {
    int numofstudents = 2;
    externals finalmarks[]
              = new externals[
                numofstudents];
    for (int i=0; i<numofstudents; i++)
    {
      finalmarks[i] = new externals();
      finalmarks[i].inputstudentdetails();
      System.out.println("enter the marks");
      finalmarks[i].inputinternalmarks();
      System.out.println("enter see marks");
      finalmarks[i].inputseemarks();
    }
    System.out.println("displaying data:\n");
    for(int i=0; i<numofstudents; i++)
    {
      finalmarks[i].calculatefinalmarks();
      finalmarks[i].displayfinalmarks();
    }
  }
}
```

Enter cie marks
enter the marks
marks 0 = 10
marks 1 = 10
marks 2 = 10
Marks 3 = 10
Marks 4 = 10

Enter see marks
subject 1 marks 10
subject 2 marks 10
subject 3 marks 10
subject 4 marks 10
subject 5 marks 10

Displaying data:

final marks [0] = 15
final marks [1] = 15
final marks [2] = 15
final marks [3] = 15
final mark [4] = 15

USN: 1BM22CS212
Name: Raehana
Sem : 3.

23/01/24

30/1/24

LAB - 7

```
import java.util.Scanner;

class wrongage extends Exception
{
    public wrongage (String s)
    {
        super (s);
    }
}
class input
{
    Scanner a = new Scanner (System.in);
}
class father extends input
{
    int fatherage;
    void fathercheck() throws wrongage
    {
        System.out.println("enter the age of father");
        fatherage = a.nextInt();
        if (fatherage < 0)
        {
            throw new wrongage ("Age cannot be negative");
        }
    }
    void display()
    {
        System.out.println("Father's age is " + fatherage);
    }
}
```

```
class son extends father
{
  int sonage;
  void soncheck() throws wrongage
  {
    System.out.println("enter son's age");
    sonage = a.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()
  {
    System.out.println("son's age is "+sonage);
  }
}
```

```
class Main
{
    public static void main (String args[])
    {
        father f = new father();
        son s = new son();
        try
        {
            f.fathercheck();
            f.display()
            s.soncheck();
            s.display();
        }
        catch (wrongage e)
        {
            System.out.println("caught" +e)
        }
    }
}
```

output :-

```
Enter the age of Father
50
Father's age is 50
Enter the age of son
20
Son's age is 20
```

Lab - 8

1) write a program which

```
class displayBMS implements Runnable
{
  public void run()
  {
    while (true)
    {
      System.out.println("BMS College of
                          Engineering");
      try
      {
        Thread.sleep(10000);
      }
      catch (InterruptedException e)
      {  e.printStackTrace();
      }
    }
  }
}
class DisplayCSE implements Runnable
{
  public void run()
  {
    while (true)
    {
      System.out.println("CSE");
    }
    try
    {
      Thread.sleep(2000);
    }
```

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

public class main
{
    public static void main(String[] args)
    {
        Thread threadBMS = new Thread(new Display
                                        BMS());
        Thread threadCSE = new Thread(new DisplayCSE
                                        ());

        threadBMS.start();
        threadCSE.start();
    }
}
```

output

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

Lab - 10

```
class Q
{
int n;
synchronized int get()
{
System.out.println("Got: " + n);
return n;
}
synchronized void put(int n)
{
this.n = n;
System.out.println("Put: " + n);
}
}
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();
      i++;
    }
  }
}
class PC
{
  public static void main(String args[])
  {
    Q q = new Q();
    new producer(q);
    new consumer(q);
    System.out.println("Press Ctrl+C to stop.");
  }
}
```

output:-

put: 1	put: 2	put: 6
Got: 1	put: 3	put: 7
Got: 1	put: 4	put: 7
Got: 1	put: 5	Got: 7

correct implementation of Producer and Consumer

```
class Q
{ int n;
  boolean valueset = false;
  synchronized int get()
  {
    while (!valueset)
    {
      try
      {
        System.out.println("\n consumer
                                waiting \n");
      wait();
      }
      catch(InterruptedException e)
      {
        System.out.println("Interrupted
                                Exception caught");
      }
    system.out.println("Got: " + n);
    valueset = false;
    System.out.println("put: " + n);
    notify();
    return n;
  }
```

```
Synchronized void put(int n)
{
  while (valueset)
  {
    try
    {
      System.out.println("\n producer
                                waiting \n");
      wait();
    }
    catch (InterruptedException caught");
  }
  this.n = n;
  valueset = true;
  System.out.println("Put: "+n);
  System.out.println("Intimate consumer");
  notify();
}
}

class producer implements Runnable
{
  Q q;
  producer(Q q)
  { this.q = q
    new Thread(this, "Producer").start();
  }
  public void run()
  {
    int i = 0;
    while (i<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 Ctrl-C to stop");
  }
}
```

output:

Put: 1
Got: 1
Put: 2
Got: 2
Put: 3
Got: 3.

06/02/24

LAB - 10

Deadlock.

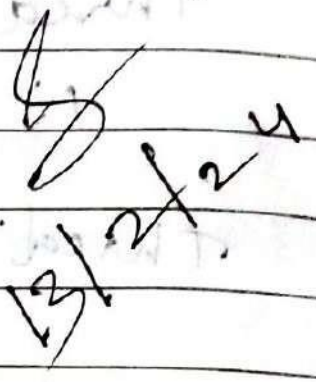
```
class A
{
  Synchronized void foo(B b)
  {
    String name = Thread.currentThread().
                                getName();
    System.out.println(name + "entered A foo");
    try
    {
      Thread.sleep(1000);
    }
    catch(Exception e)
    {
      System.out.println("A Interrupted");
  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

    Thread t = new Thread(this, "Racing
                                Thread");
    t.start();
    a.foo(b);
    Thread

    System.out.println("Back in
                        main Thread");
  }

public void run()
{   b.bar(a);
    System.out.println("Back in other
                        thread");
}
public static void main(String args[])
{
  new Deadlock();
}
}
```

output

Main Thread entered A. foo
Racing Thread entered B. bar
main Thread trying to call B.last()
Inside A.last
Back in main Thread
Inside A.last
Back in other thread.



20/2/24

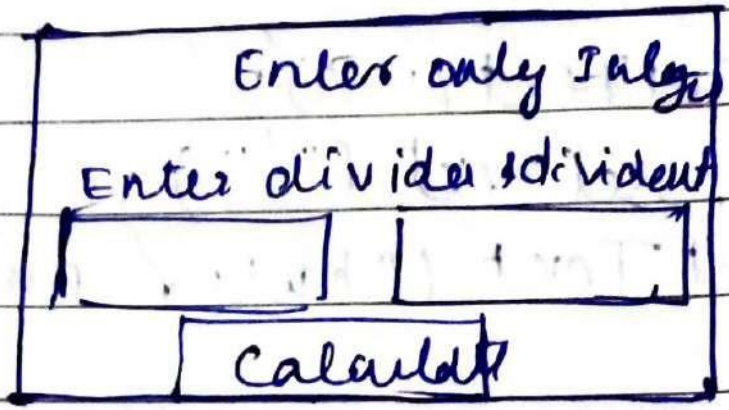
LAB-9

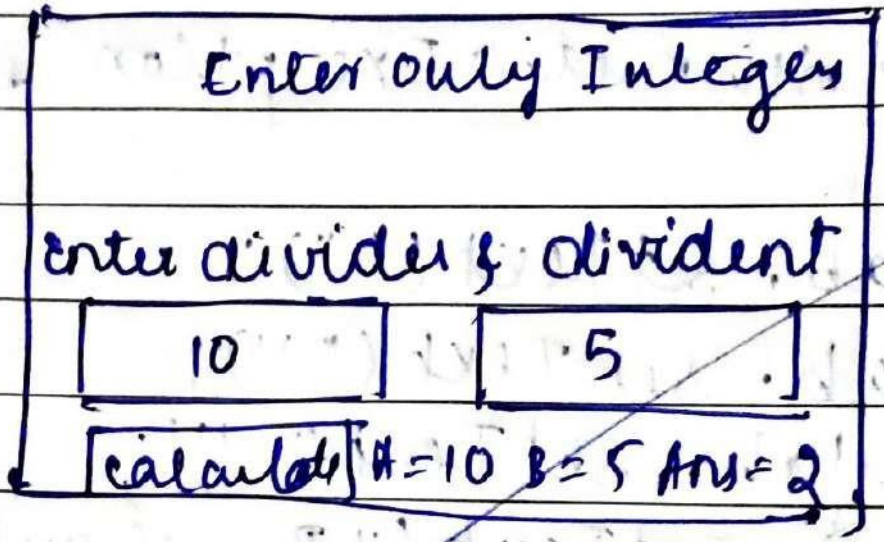
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

class SwingDemo
{
    SwingDemo()
    {
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.
                                      EXIT_ON_
                                      CLOSE);
        JLabel jlab = new JLabel("Enter the
                                  divider and
                                  divident:");
        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);
        JButton button = new JButton("Calculate");
        JLabel err = new JLabel();
        JLabel alab = new JLabel();
        JLabel blab = new JLabel();
        JLabel anslab = new JLabel();
        //add in order
        jfrm.add(err);
        jfrm.add(jlab);
        jfrm.add(ajtf);
        jfrm.add(bjtf);
        jfrm.add(button);
        jfrm.add(alab);
        jfrm.add(blab);
        jfrm.add(anslab);
```

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

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

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

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

output :-


Enter only Integ
Enter divider & divident
Calculate


Enter only Integers
enter divider & divident
10
5
calculate
A=10 B=5 Ans=2