

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



**LAB REPORT**  
**on**

## **Object Oriented Java Programming** **(23CS3PCOOJ)**

*Submitted by*

**B Vatsal (1BM23CS061)**

*in partial fulfillment for the award of the degree of*  
**BACHELOR OF ENGINEERING**  
*in*  
**COMPUTER SCIENCE AND ENGINEERING**



**B.M.S. COLLEGE OF ENGINEERING**  
(Autonomous Institution under VTU)

**BENGALURU-560019**  
**Sep-2024 to Jan-2025**

**B.M.S. College of Engineering,**  
**Bull Temple Road, Bangalore 560019**  
(Affiliated To Visvesvaraya Technological University, Belgaum)  
**Department of Computer Science and Engineering**



**CERTIFICATE**

This is to certify that the Lab work entitled “Object Oriented Java Programming (23CS3PCOOJ)” carried out by **B Vatsal (1BM23CS061)**, who is bonafide student of **B.M.S. College of Engineering**. It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

Swetha Sridharan Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE
---	---

## Index

<b>Sl. No.</b>	<b>Date</b>	<b>Experiment Title</b>	<b>Page No.</b>
1	1/10/24	Quadratic Equation	4-6
2	8/10/24	Calculating SGPA	7-12
3	15/10/24	Book Details	13-19
4	22/10/24	Abstract Class Shape	20-24
5	29/10/24	Bank Details	25-34
6	12/11/24	Packages	35-43
7	19/11/24	Interface	44-48
8	26/11/24	Exception Handling	49-54
9	3/12/24	Threads	55-58
10	3/12/24	GUI – Java Swing	59-65

Github Link:

<https://github.com/Vatsalshetty/1BM23CS061>

### **Program 1**

Implement Quadratic Equation

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

Algorithm:

Java program to solve the quadratic equations.

```
import java.util.Scanner;
```

```
class Quadratic {
```

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

```
        int a, b, c;
```

```
        double d, r1, r2;
```

```
        Scanner p = new Scanner (System.in);
```

```
        System.out.println ("Enter values of a, b, c:");
```

```
        a = p.nextInt();
```

```
        b = p.nextInt();
```

```
        c = p.nextInt();
```

```
        d = b*b - 4*a*c;
```

```
        if (d > 0.0)
```

```
        {  
            r1 = (-b + Math.sqrt(d)) / (2.0 * a);
```

```
            r2 = (-b - Math.sqrt(d)) / (2.0 * a);
```

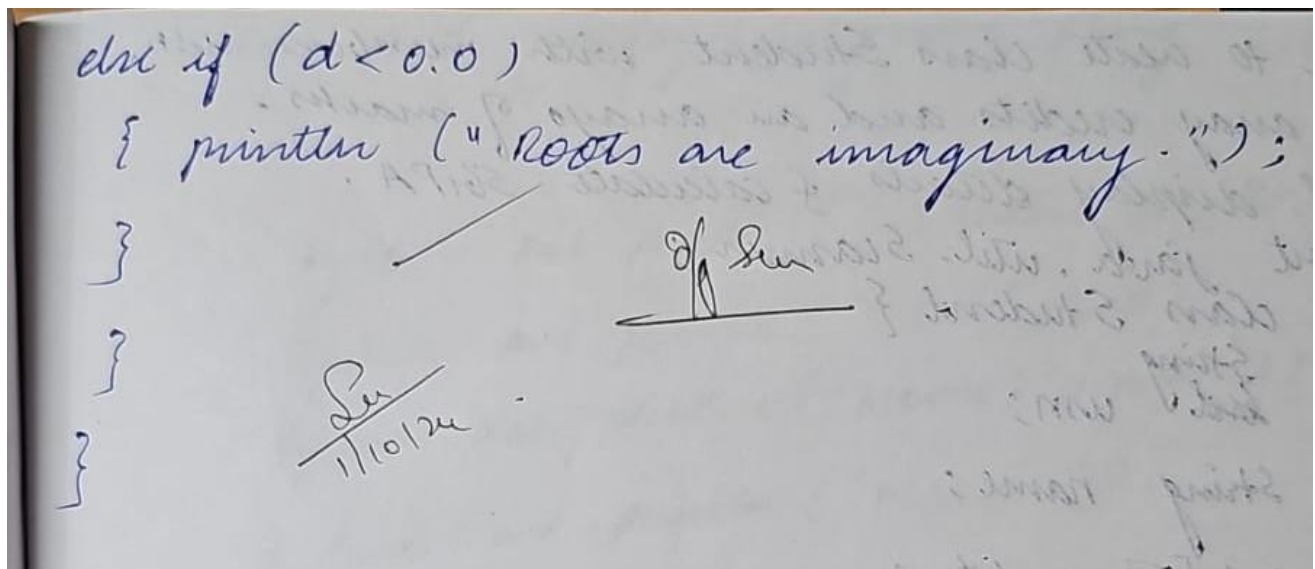
```
            System.out.println ("The roots are " + r1 + " and " + r2);  
        }
```

```
        else if (d == 0.0)
```

```
        {  
            r1 = -b / (2.0 * a);
```

```
            System.out.println ("The roots are equal, value is " + r1);  
        }
```

```
    }  
}
```



Code:

```
import java.util.Scanner;
public class Quad
{
    public static void main(String[] args)
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the coefficients:");
        int a=s.nextInt();
        int b=s.nextInt();
        int c=s.nextInt();
        int d=b*b-4*a*c;
        if(d>0)
        {
            System.out.println((-b+Math.sqrt(d))/(2*a));
            System.out.println((-b-Math.sqrt(d))/(2*a));
            System.out.println("Roots are unique");
        }
        else if(d==0)
        {
            System.out.println("Roots are equal");
            System.out.println("Roots are:");
            System.out.println(-b/2*a);
        }
        else
        {
            System.out.println("No real roots");
        }
        System.out.println("B VATSAL");
        System.out.println("1BM23CS061");
    }
}
```

```
}  
}
```

Output

```
C:\Users\Vatsal\OneDrive\Documents\java>java Quad.java  
Enter the coefficients:  
2  
1  
9  
No real roots  
B VATSAL  
1BM23CS061  
  
C:\Users\Vatsal\OneDrive\Documents\java>
```

### **Program 2**

Calculating SGPA

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.

Java program to create class Student with members name, an array credits and an array of marks. Accept and display details & calculate SGPA.

```
import java.util.Scanner;
→ public class Student {
    String usn;
    String name;
    int [5] credits;
    int [5] marks;
    public void studentInfo (String usn, String name) {
        this.usn = usn;
        this.name = name;
    }
```

```
    public void creditsandmarks (int [5] credits, int [5] marks)
```

```
    {
        Scanner p = new Scanner (System.in);
```

```
        System.out.println ("Enter 5 subject marks:");
```

```
        credits
```

```
        marks[0] = p.next Int();
```

```
        marks[1] = p.next Int();
```

```
        marks[2] = p.next Int();
```

```
        marks[3] = p.next Int();
```

```
        marks[4] = p.next Int();
```

for loop

```
        System.out.println ("Enter 5 credits:");
```

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

```
        {
            credits[i] = p.next Int();
```

```
        }
```



```
public void display(String name, String  
    ) {
```

```
    System.out.println("Student Details");
```

```
    System.out.println("USN:" + usn);
```

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

```
    System.out.println("Marks:");
```

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

```
        System.out.println(marks[i]);
```

```
    System.out.println("Credits:");
```

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

```
        System.out.println(credits[i]);
```

```
}
```

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

```
    Student sp = new Student();
```

```
    sp.studentInfo("IBM20CS461", "YZXH");
```

```
    sp.creditsAndMarks(int[5] credits,  
                       int[5] marks);
```

```
    sp.display();
```

```
    double  
int sum;
```

```
    sum = 0
```

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

```
        sum += credits[i];
```

```
    sum = sum / 5.000;
```

```
    System.out.println("The SGPA is " + sum);
```

```
}
```

```
}
```

Code:

```
import java.util.Scanner;

public class Student
{
    String usn;
    String name;
    int[] credits = new int[5];
    int[] marks = new int[5];

    public void studentInfo(String usn, String name)
    {
        this.usn = usn;
        this.name = name;
    }

    public void enterCreditsAndMarks()
    {
        Scanner scanner = new Scanner(System.in);

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

        System.out.println("Enter credits for 5 subjects: ");
        for (int i = 0; i < 5; i++)
        {
            System.out.print("Subject " + (i + 1) + " credits: ");
            credits[i] = scanner.nextInt();
        }
    }

    public void display()
    {
        System.out.println("\nStudent Information:");
        System.out.println("USN: " + usn);
    }
}
```

```

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

        System.out.println("Credits: ");
        for (int i = 0; i < 5; i++)
        {
            System.out.println("Subject " + (i + 1) + ": " + credits[i]);
        }

        // Calculate GPA
        int totalCredits = 0;
        int weightedSum = 0;
        for (int i = 0; i < 5; i++)
        {
            totalCredits += credits[i];
            weightedSum += marks[i] * credits[i];
        }
        float gpa = (float) weightedSum / totalCredits;

        System.out.println("GPA: " + gpa);
    }

    public static void main(String[] args)
    {
        Student student = new Student();
        student.studentInfo("1BMACS001", "ABC");
        student.enterCreditsAndMarks();
        student.display();
        System.out.println("B VATSAL");
        System.out.println("1BM23CS061");
    }
}

```

Output:

```
C:\Users\Vatsal\OneDrive\Documents\java\LAB-2>java Student.java
Enter marks for 5 subjects:
Subject 1 marks: 54
Subject 2 marks: 56
Subject 3 marks: 87
Subject 4 marks: 97
Subject 5 marks: 45
Enter credits for 5 subjects:
Subject 1 credits: 4
Subject 2 credits: 1
Subject 3 credits: 2
Subject 4 credits: 3
Subject 5 credits: 2

Student Information:
USN: 1BMACS001
Name: ABC
Marks:
Subject 1: 54
Subject 2: 56
Subject 3: 87
Subject 4: 97
Subject 5: 45
Credits:
Subject 1: 4
Subject 2: 1
Subject 3: 2
Subject 4: 3
Subject 5: 2
GPA: 68.916664
B VATSAL
1BM23CS061
```

### **Program 3**

#### Book Details

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.

Algorithm:

Write a program to take details of n books consisting of book name, author, price, no of pages.

```
import java.util.*;
```

```
class Book {
```

```
    private String name;
```

```
    private String author;
```

```
    private int price;
```

```
    private int numpages;
```

```
    public Book (String name, String author,  
                int price, int numpages) {
```

```
        this.name = name;
```

```
        this.author = author;
```

```
        this.price = price;
```

```
        this.numpages = numpages;
```

```
    }
```

```
    public void display () {
```

```
        System.out.println ("Student Details");
```

```
        System.out.println ("Name" + name);
```

```
        System.out.println ("Author" + author);
```

```
        System.out.println ("Price" + price);
```

```
        System.out.println ("Number of pages" + numpages);
```

```
    }
```

```

public void getname ( ) {
    this.name = name;
}

public void setname ( ) {
    // return name;
}

public void get & getauthor ( ) {
    this.author = author;
}

public void setauthor ( ) {
    // this return author;
}

public void get & setprice ( ) {
    this.price = price;
}

public void setprice ( ) {
    // return price;
}

public void getpages ( ) {
    this.numpages = numpages;
}

```



```
public void setpages() {
```

```
    return numpages;
```

```
}
```

```
public String toString() {
```

```
    return ("Name" + name + " Author" + author +  
        " Price" + price + " Pages" + numpages);
```

```
}
```

```
}
```

```
class Main {
```

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

```
        Scanner q = new Scanner (System.in);  
        int [] Books
```

```
        int n;
```

```
        System.out.println ("Enter the no. of books:");
```

```
        n = q.nextInt();
```

```
        Book [] books = new Book [n];
```

```
        for (int i=0; i<n; i++) {
```

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

```
            name = q.nextLine();
```

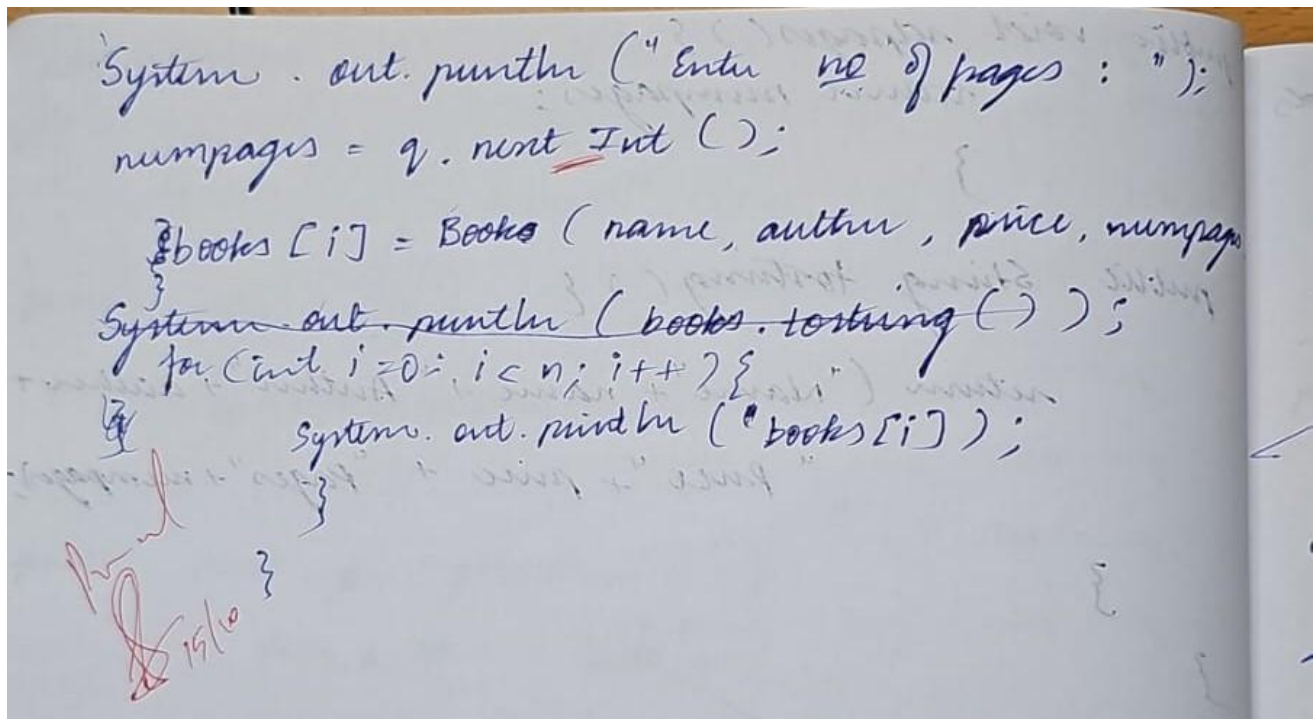
```
            System.out.println ("Enter author name:");
```

```
            author = q.nextLine();
```

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

```
            price = q.nextInt();
```





Code:

```
import java.util.*;
```

```
class Book {
```

```
    Scanner in = new Scanner(System.in);
```

```
    String name, author;
```

```
    int price, num_pages;
```

```
    Book(String n, String a, int p, int np) {
```

```
        name = n;
```

```
        author = a;
```

```
        price = p;
```

```
        num_pages = np;
```

```
    }
```

```
    void setDetails() {
```

```
        System.out.println("Enter Book Name: ");
```

```
        name = in.nextLine();
```

```
        System.out.println("Enter Author Name: ");
```

```
        author = in.nextLine();
```

```
        System.out.println("Price of the Book: ");
```

```
        price = in.nextInt();
```

```
        System.out.println("No of pages: ");
```

```
        num_pages = in.nextInt();
```

```
        in.nextLine();    }
```

```

void getDetails() {
    System.out.println("Book Name: " + name);
    System.out.println("Author Name: " + author);
    System.out.println("Book price: " + price);
    System.out.println("No of pages: " + num_pages);
}

public String toString() {
    return "Book Name: " + name + "\nAuthor Name: " + author + "\nBook price: " + price + "\nNo
of pages: " + num_pages;
}
}

public class Lab_3 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.println("Enter number of books:");
        int noBook = in.nextInt();
        in.nextLine();

        Book[] bk = new Book[noBook];
        for (int i = 0; i < noBook; i++) {
            System.out.println("Book: " + (i + 1));
            bk[i] = new Book("", "", 0, 0);
            bk[i].setDetails();
        }

        System.out.println("\nBook Details:");
        for (int k = 0; k < noBook; k++) {
            System.out.println();
            bk[k].getDetails();
            System.out.println(bk[k].toString());
        }
        System.out.println("B VATSAL");
        System.out.println("1BM23CS061");
    }
}

```

Output:

```

C:\Users\Vatsal\OneDrive\Documents\java\LAB-3>java Lab_3.java
Enter number of books:
2
Book: 1
Enter Book Name:
sdg
Enter Author Name:
sfdg
Price of the Book:
46
No of pages:
656
Book: 2
Enter Book Name:
dsf
Enter Author Name:
sfdg
Price of the Book:
546
No of pages:
656

Book Details:

Book Name: sdg
Author Name: sfdg
Book price: 46
No of pages: 656
Book Name: sdg
Author Name: sfdg
Book price: 46
No of pages: 656

Book Name: dsf
Author Name: sfdg
Book price: 546
No of pages: 656
Book Name: dsf
Author Name: sfdg
Book price: 546
No of pages: 656
B VATSAL
1BM23CS061

```

#### **Program 4**

##### **Abstract Class Shape**

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.

##### **Algorithm**

Java program to create abstract class called Shape that prints area of rectangle, triangle & circle.

```
import java.util.*;
```

```
abstract class Shape {
```

```
    int a, b;
```

```
    Shape (int a, int b) {
```

```
        this.a = this a;
```

```
        this.b = b;
```

```
    }
```

```
    abstract void printArea();
```

```
}
```

```
class Rectangle extends Shape {
```

```
    Rectangle (int length, int breadth) {
```

```
        super (length, breadth);
```

```
    }
```

```
    @Override
```

```
    void printArea() {
```

```
        System.out.println ("Area of rectangle : " + (a * b));
```

```
    }
```

```
}
```

```
class Triangle extends Shape {
```

```
    Triangle (int base, int height) {
```

```
        super (base, height);
```

```
    }
```

```
    @Override
```

```
    void printArea() {
```

```
        double area = 0.5 * a * b;
```

```
        System.out.println ("Area of Triangle: " + area);
```

```
    }
```

```

    }
    class Circle extends Shape {
        Circle (int radius) {
            super (radius, 0);
        }
        @Override
        void printArea() {
            double Area = Math.PI * a * a;
            System.out.println ("Area of circle : " + area);
        }
    }
}

```

```

public class Shapes {
    public void static void main (String[] args) {
        Scanner s = new Scanner (System.in);

        System.out.println ("Enter the length of rectangle :");
        int x = s.nextInt();

        System.out.println ("Enter the breadth of rectangle :");
        int y = s.nextInt();

        System.out.println ("Enter length of triangle :");
        int p = s.nextInt();

        System.out.println ("Enter the length of triangle :");
        int q = s.nextInt();

        System.out.println ("Enter radius of circle :");
        int r = s.nextInt();

        Shape rectangle = new Rectangle (x, y);
        Shape triangle = new Triangle (p, q);
    }
}

```

Shape

re

se

circ

}

}

Output

Enter

20

Enter

20

Enter

20

Enter

20

Enter

20

Area of

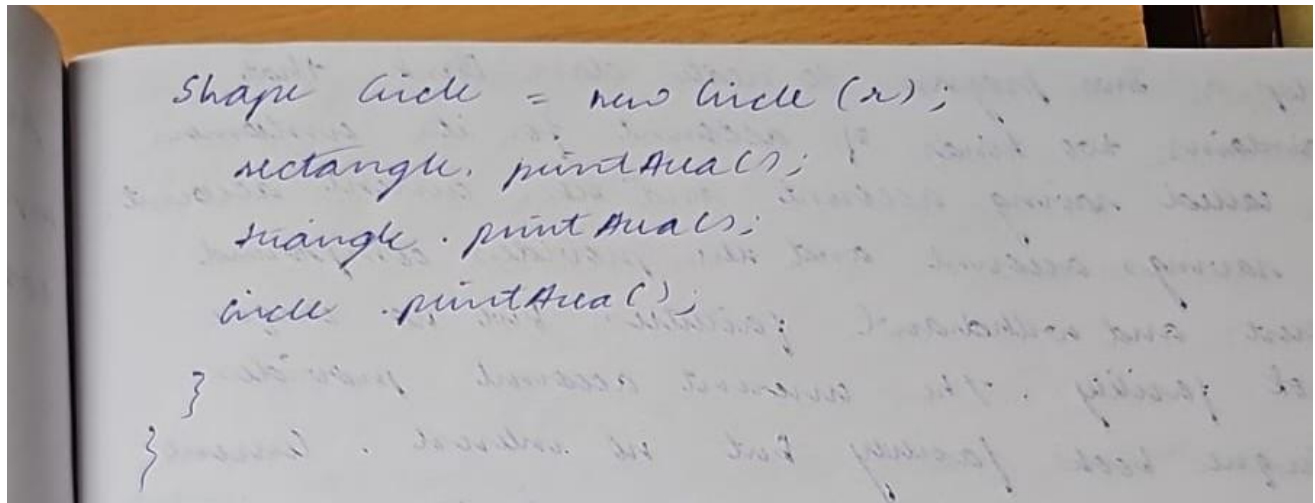
Area of

Area

Seen

Gt

22/10/24



Code:

```
import java.util.Scanner;
```

```
abstract class Shape {  
    int dim1, dim2;
```

```
    Shape(int dim1, int dim2) {  
        this.dim1 = dim1;  
        this.dim2 = dim2;  
    }
```

```
    abstract void printArea();  
}
```

```
class Rectangle extends Shape {  
    Rectangle(int length, int breadth)  
    {  
        super(length, breadth);  
    }
```

```
    @Override  
    void printArea() {  
        int area = dim1 * dim2;  
        System.out.println("Area of Rectangle: " + area);  
    }  
}
```

```
class Triangle extends Shape {  
    Triangle(int base, int height) {
```

```

        super(base, height);
    }

    @Override
    void printArea() {
        double area = 0.5 * dim1 * dim2;
        System.out.println("Area of Triangle: " + area);
    }
}

class Circle extends Shape {
    Circle(int radius) {
        super(radius, 0);
    }

    @Override
    void printArea()
    {
        double area = Math.PI * dim1 * dim1;
        System.out.println("Area of Circle: " + area);
    }
}

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

        System.out.print("Enter length and breadth of the rectangle: ");
        int rectLength = scanner.nextInt();
        int rectBreadth = scanner.nextInt();
        Shape rectangle = new Rectangle(rectLength, rectBreadth);

        System.out.print("Enter base and height of the triangle: ");
        int triBase = scanner.nextInt();
        int triHeight = scanner.nextInt();
        Shape triangle = new Triangle(triBase, triHeight);

        System.out.print("Enter radius of the circle: ");
        int circleRadius = scanner.nextInt();
        Shape circle = new Circle(circleRadius);

        rectangle.printArea();
        triangle.printArea();
        circle.printArea();
    }
}

```

```
        scanner.close();  
    }  
}
```

Output:

```
Enter length and breadth of the rectangle: 45  
12  
Enter base and height of the triangle: 10  
15  
Enter radius of the circle: 2  
Area of Rectangle: 540  
Area of Triangle: 75.0  
Area of Circle: 12.566370614359172
```



## **Program 5**

### **Bank Details**

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

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

Algorithm:

```

import java.util.*;

class Account {
    private String customer_name;
    private int Accno;
    private int balance;
    private String type;

    public Account (String customer_name, int Accno) {
        this.customer_name = customer_name;
        this.Accno = Accno;
        this.balance = balance;
    }
}

class Cur-acct & extends Account {
    System.out.println ("Cheque book available  
for this account");

    int min_bal = 2000;
    int serv_charge = 150;
    int deposit (int amount) {
        balance = balance + amount;
        return balance;
    }

    void min_bal () {
        if (balance < min_bal) {
            System.out.println ("Service charge is  
imposed due to  
min balance");
            balance = balance - serv_charge;
        }
    }
}

```

```

int withdrawl (int amount) {
    if (balance < amount) {
        System.out.println ("Can't withdraw");
        return 0;
    }
    else {
        balance -= amount;
        return balance;
    }
}

```

```

void display() {
    System.out.println ("Current balance is " + balance);
}
}

```

```

class Sav-act extends Account {
    double
    int no acc :
    double interest = (balance) * (10% / 100);
    balance
    balance no :
    int deposit
    Sav-act (String Customer-name, int Accno,
             double int balance)
    {
        super (Customer-name, Accno, balance);
        "savings";
    }
}

```

```

void deposit (int amount)
{
    double ror = 8;
    int interest = (balance) * (ror/100);
    balance += interest;
}

```

@ Override

```

void withdraw (int amount) {
    if (balance >= amount) {
        balance -= amount;
    }
    else {
        System.out.println ("Insufficient balance");
    }
}

```

```

public class Bank {
    public static void main (String args[]) {
        Scanner s = new Scanner (System.in);
        System.out.println ("Enter the customer name: ");
        String name = s.nextLine();
        System.out.println ("Enter account type: Savings / Current");
        String type = s.nextLine();
    }
}

```



```
System.out.println("Enter the balance");  
int balance = s.nextInt();
```

```
if (type == "Savings") {
```

```
    Account account = new Sav_acc  
        (customer_name, Account_type, balance);
```

```
}
```

```
else {
```

```
    Account account = new Cur_acc  
        (customer_name, Account_type, balance);
```

```
}
```

```
boolean exit = false;
```

```
while (!exit) {
```

```
    System.out.println("In. Choose an option In
```

```
    1. Deposit In 2. Display
```

```
    balance In 3. Compute
```

```
    Interest In 4. Withdraw
```

```
    In 5. Exit 6. Exit In
```

```
    int opt = s.nextInt();
```

```
    switch (opt) {
```

```
        case 1: System.out.println("Enter amount  
            to be  
            deposited");
```

```
int amount = s.nextInt();  
account.deposit(amount);  
break;
```

case 2 :

```
account.display();  
break;
```

case 3 :

```
account.deposit(0);  
break;
```

case 4 :

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

```
int amt = s.nextInt();
```

```
account.withdraw(amt);  
break;
```

case 5 :

```
break;
```

} default :

```
System.out.println("Invalid option");
```

```
}
```

```
}
```

```
}
```

```
}
```

29.10  
proceed

Code:

```
import java.util.Scanner;

class Account {
    String acc_name, acc_no, acc_type;
    double balance;

    Account(String name, String no, String acc, double bal) {
        this.acc_name = name;
        this.acc_no = no;
        this.acc_type = acc;
        this.balance = bal;
    }

    void deposit(double amt) {
        balance += amt;
        System.out.println("Deposit = " + amt);
    }

    void withdraw(double amt) {
        if (amt > balance) {
            System.out.println("Insufficient Balance");
        } else {
            balance -= amt;
            System.out.println("Withdrawal Amount = " + amt);
        }
    }

    void checkBalance() {
        System.out.println("Available Balance = " + balance);
    }
}

class CurAcct extends Account {
    CurAcct(String name, String no, double bal) {
        super(name, no, "Current Account", bal);
    }

    void minBalance() {
```

```

        if (balance < 5000) {
            System.out.println("Min Balance in Current Account should be 5000. Service charge of 100
will be charged.");
            this.balance -= 100.0;
        }
    }
}

```

```

void cheque(double amt) {
    balance -= amt;
    System.out.println("Cheque Amount = " + amt);
}
}

```

```

class SavAcct extends Account {
    SavAcct(String name, String no, double bal) {
        super(name, no, "Saving Account", bal);
    }
}

```

```

void computeInterest(int years) {
    double initialBalance = balance;
    double interest = balance * 5 / 100.0 * years;
    initialBalance += interest;
    System.out.println("Interest earned on savings for " + years + " years is " + interest);
    System.out.println("Balance after " + years + " years will be " + initialBalance);
}
}

```

```

public class Lab5 {
    public static void main(String[] args) {
        CurAcct currentAccount = new CurAcct("John Doe", "CA123", 10000);
        SavAcct savingAccount = new SavAcct("Jane Doe", "SA456", 15000);
        Scanner sc = new Scanner(System.in);

```

```

        System.out.println("Enter:\n1. To deposit in Current Account\n2. To withdraw from Current
Account\n3. To check balance in Current Account\n4. To deposit in Saving Account\n5. To withdraw
from Saving Account\n6. To check balance in Saving Account\n7. To compute interest in Saving
Account\n0. To exit");

```

```

        int option;
        do {
            System.out.print("\nChoose an option: ");
            option = sc.nextInt();

            switch (option) {
                case 1:
                    System.out.print("Enter amount to deposit: ");
                    double currentDeposit = sc.nextDouble();

```



```

        currentAccount.deposit(currentDeposit);
        break;
    case 2:
        System.out.print("Enter amount to withdraw: ");
        double currentWithdraw = sc.nextDouble();
        currentAccount.withdraw(currentWithdraw);
        break;
    case 3:
        currentAccount.checkBalance();
        break;
    case 4:
        System.out.print("Enter amount to deposit: ");
        double savingDeposit = sc.nextDouble();
        savingAccount.deposit(savingDeposit);
        break;
    case 5:
        System.out.print("Enter amount to withdraw: ");
        double savingWithdraw = sc.nextDouble();
        savingAccount.withdraw(savingWithdraw);
        break;
    case 6:
        savingAccount.checkBalance();
        break;
    case 7:
        System.out.print("Enter number of years to compute interest: ");
        int years = sc.nextInt();
        savingAccount.computeInterest(years);
        break;
    case 0:
        System.out.println("Exiting...");
        break;
    default:
        System.out.println("Invalid input");
    }
} while (option != 0);
}
}

```

Output:

Enter:

1. To deposit in Current Account
2. To withdraw from Current Account
3. To check balance in Current Account
4. To deposit in Saving Account
5. To withdraw from Saving Account
6. To check balance in Saving Account
7. To compute interest in Saving Account
0. To exit

Choose an option: 1

Enter amount to deposit: 455

Deposit = 455.0

Choose an option: 2

Enter amount to withdraw: 256

Withdrawal Amount = 256.0

Choose an option: |

Enter:

1. To deposit in Current Account
2. To withdraw from Current Account
3. To check balance in Current Account
4. To deposit in Saving Account
5. To withdraw from Saving Account
6. To check balance in Saving Account
7. To compute interest in Saving Account
0. To exit

Choose an option: 1

Enter amount to deposit: 455

Deposit = 455.0

Choose an option: 2

Enter amount to withdraw: 256

Withdrawal Amount = 256.0

Choose an option: |

## **Program 6**

### Packages

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.

Algorithm:

Create a package CIE which has two classes Student and Internals. The class Personal has members like usn, name, sem. The class

```
package CIE
import java.util.*;

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

    public void inputDetails() {
        Scanner s = new Scanner (System.in);
        System.out.println ("Enter the name of student");
        this.name = s.nextLine();
        System.out.println ("Enter the semester.");
        this.sem = s.nextInt();
    }

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

```

package CIE
import java.util.Scanner;
public class internal extends Student {
    private double ciem[] = new double [5];
    Scanner s = new Scanner (System.in);
    public void input CIE () {
        for (int i = 0; i < 5; i++) {
            System.out.println ("Enter the CIE
                                marks");
            this.ciem[i] = s.nextDouble();
        }
    }
}

```

```

package SEE
import CIE.*;
import java.util.Scanner;
public class External extends internal {
    private double sum = new double [5];
    private double finalm = new double [5];
    public void input SEE () {
        Scanner s = new Scanner (System.in);
        for (int i = 0; i < 5; i++) {
            System.out.println ("Enter SEE
                                marks");
            this.sum[i] = s.nextDouble();
        }
    }
}

```

```

public void calfinal() {
    for (int i = 0; i < 5; i++) {
        this.finalm[i] = ciem[i] + (xhm[i]) / 2;
    }
}

```

```

public void displayfinal() {
    System.out.println("Final marks of student is");
    for (int i = 0; i < 5; i++) {
        System.out.println("this finalm[i]");
    }
}

```

```

import java.util.Scanner;

```

```

class Main {

```

```

    public static void main (String[] args) {
        int n;
        Scanner s = new Scanner(System.in);

```

```

        System.out.println("Enter no of students.");
        n = s.nextInt();

```

```

        External e[] = new External[n];

```

```

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

```

```

            e[i] = new External();

```

```

            System.out.println("Enter the " + (i+1) + "th details of the marks");

```



```

e[i].input details ();
System.out.println ("Enter the " + (i+1) + "th CIE
marks");
e[i].input CIE ();
System.out.println ("The details of students is :");
e[i].display ();
System.out.println ("The final marks of the " + (i+1) +
"th student is ");
e[i].display final ();
}
}
}

```

Code:

```
package CIE;
```

```
public class Student {
    public String usn;
    public String name;
    public int sem;
}
```

```
package CIE;
```

```
package CIE;
```

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

```
    public Student(String usn, String name, int sem) {
        this.usn = usn;
```

```

        this.name = name;
        this.sem = sem;
    }
}

```

```

package CIE;
import CIE.Student;

```

```

public class Internals extends Student {
    public int[] internalMarks;

    public Internals(String usn, String name, int sem, int[] internalMarks) {
        super(usn, name, sem);
        this.internalMarks = internalMarks;
    }
}

```

```

package SEE;

```

```

import CIE.Student;

```

```

public class External extends Student {
    public int[] seeMarks;

    public External(String usn, String name, int sem, int[] seeMarks) {
        super(usn, name, sem);
        this.seeMarks = seeMarks;
    }
}

```

```

import CIE.Internals;
import SEE.External;

```

```

import java.util.Scanner;

```

```

public class Main
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the number of students: ");
        int n = sc.nextInt();
    }
}

```



```

Internals[] internals = new Internals[n];
External[] externals = new External[n];

for (int i = 0; i < n; i++)
{
    System.out.println("\nEnter details for student " + (i + 1) + ":");

    System.out.print("USN: ");
    String usn = sc.next();

    System.out.print("Name: ");
    String name = sc.next();

    System.out.print("Semester: ");
    int sem = sc.nextInt();

    int[] internalMarks = new int[5];
    System.out.println("Enter internal marks for 5 courses:");
    for (int j = 0; j < 5; j++)
    {
        internalMarks[j] = sc.nextInt();
    }
    internals[i] = new Internals(usn, name, sem, internalMarks);

    int[] seeMarks = new int[5];
    System.out.println("Enter SEE marks for 5 courses:");
    for (int j = 0; j < 5; j++)
    {
        seeMarks[j] = sc.nextInt();
    }
    externals[i] = new External(usn, name, sem, seeMarks);
}

System.out.println("\nFinal Marks for each student:");
for (int i = 0; i < n; i++)
{
    System.out.println("Student " + (i + 1) + ":");
    for (int j = 0; j < 5; j++)
    {
        int finalMarks = internals[i].internalMarks[j] + (externals[i].seeMarks[j] / 2);
        System.out.println("Course " + (j + 1) + ": " + finalMarks);
    }
}
}
}

import CIE.Internals;
import SEE.External;

```

```

import java.util.Scanner;

public class Main
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the number of students: ");
        int n = sc.nextInt();

        Internals[] internals = new Internals[n];
        External[] externals = new External[n];

        for (int i = 0; i < n; i++)
        {
            System.out.println("\nEnter details for student " + (i + 1) + ":");

            System.out.print("USN: ");
            String usn = sc.next();

            System.out.print("Name: ");
            String name = sc.next();

            System.out.print("Semester: ");
            int sem = sc.nextInt();

            int[] internalMarks = new int[5];
            System.out.println("Enter internal marks for 5 courses:");
            for (int j = 0; j < 5; j++)
            {
                internalMarks[j] = sc.nextInt();
            }
            internals[i] = new Internals(usn, name, sem, internalMarks);

            int[] seeMarks = new int[5];
            System.out.println("Enter SEE marks for 5 courses:");
            for (int j = 0; j < 5; j++)
            {
                seeMarks[j] = sc.nextInt();
            }
            externals[i] = new External(usn, name, sem, seeMarks);
        }

        System.out.println("\nFinal Marks for each student:");
        for (int i = 0; i < n; i++)

```

```

    {
        System.out.println("Student " + (i + 1) + ":");
        for (int j = 0; j < 5; j++)
        {
            int finalMarks = internals[i].internalMarks[j] + (externals[i].seeMarks[j] / 2);
            System.out.println("Course " + (j + 1) + ": " + finalMarks);
        }
    }
}

```

Output:

```

Enter the number of students: 1

Enter details for student 1:
USN: esr
Name: dsfg
Semester: 3
Enter internal marks for 5 courses:
23
23
45
56
56
Enter SEE marks for 5 courses:
987
78

788
89
899

Final Marks for each student:
Student 1:
Course 1: 516
Course 2: 62
Course 3: 439
Course 4: 100
Course 5: 505

```

## **Program 7**

Interfaces

Algorithm:

## Polygon

```
import java.util.*;
```

```
interface polygon {
```

```
    default void getperimeter (int n, int len) {
```

```
        System.out.println("The perimeter of polygon is " + (n * len));
```

```
    }
```

```
    void area (String shape, int len);
```

```
}
```

```
class Shape implements polygon {
```

```
    @Override
```

```
    public void area (String shape, int len) {
```

```
        if (shape.equals("square"))
```

```
            System.out.println("The area is " + (len * len));
```

```
        if (shape.equals("rectangle")) {
```

```
            System.out.println("Enter breadth ");
```

```
            int b = nextInt();
```

```
            System.out.println("The area is " + (len * b));
```

```
        }
```

```
        if (shape.equals("triangle")) {
```

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

```
            int h = nextInt();
```

```
            System.out.println("The area is " + (0.5 * len * h));
```

```
        }
```

```

else
    System.out.println("Shape not recognized");
}

public static void main (String[] args) {
    Scanner s = new Scanner (System.in);
    System.out.println("Enter the shape (square, rectangle, triangle):");
    String shape = s.nextLine();
    System.out.println("Enter the length of one side:");
    int side = s.nextInt();

    Shape sp = new Shape();
    sp.getPerimeter (side, den);
    sp.area (shape, den);
}
}

```

Code:

```
import java.util.Scanner;
```

```
interface Polygon {
    double getPerimeter();
    double getArea();
}
```

```
class Square implements Polygon {
    private double side;
```

```
    Square(double side) {
        this.side = side;
    }
```

```
    @Override
    public double getPerimeter() {
        return 4 * side;
    }
}
```



```

    @Override
    public double getArea() {
        return side * side;
    }
}

class Triangle implements Polygon {
    private double side;

    Triangle(double side) {
        this.side = side;
    }

    @Override
    public double getPerimeter() {
        return 3 * side;
    }

    @Override
    public double getArea() {
        return (Math.sqrt(3) / 4) * Math.pow(side, 2);
    }
}

public class Maininterface {
    public static void main(String[] args) {
        double s, t;
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the length of side of square: ");
        s = sc.nextDouble();

        System.out.print("Enter the length of side of triangle: ");
        t = sc.nextDouble();

        Square square = new Square(s);
        System.out.println("Square Perimeter: " + square.getPerimeter());
        System.out.println("Square Area: " + square.getArea());

        Triangle tri = new Triangle(t);
        System.out.println("Triangle Perimeter: " + tri.getPerimeter());
        System.out.println("Triangle Area: " + tri.getArea());
        sc.close();
    }
}

```

Output:

```
Enter the length of side of square: 5
Enter the length of side of triangle: 2
Square Perimeter: 20.0
Square Area: 25.0
Triangle Perimeter: 6.0
Triangle Area: 1.7320508075688772
```

## **Program 8**

### Exception Handling

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 uses both father and son's age and throws an exception if son's age is >=father's age.

Algorithm:

LAB

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

class Father
{
    int fa;
    public Father (int age) throws WrongAgeException
    {
        if (age < 0)
            throws new WrongAgeException ("Age is negative");
        fa = age;
        System.out.println ("Father age " + fa);
    }
}

class Son extends Father
{
    int sa;
    public Son (int fa, int sa) throws WrongAgeException
    {
        super (fa);
        if (sa < 0)
            throws new WrongAgeException ("Age is negative");
    }
}
```

```

if (sa >= fa)
    throw new WrongAgeException ("Son's age is greater than father's age");

this.sa = sa;
System.out.println ("Son's age is: " + sa);
}
}
}

public class Lab
{
    public static void main (String[] args)
    {
        try
        {
            Father f = new Father (40);
            Son s = new Son (40, 15);
        }
        catch (WrongAgeException e)
        {
            System.out.println ("Exception: " + e.getMessage());
        }
    }
}

```

Code:

```

class WrongAgeException extends Exception
{
    public WrongAgeException(String message)
    {

```

```

        super(message);
    }
}

class Father
{
    protected int age;

    public Father(int age) throws WrongAgeException
    {
        if (age < 0)
        {
            throw new WrongAgeException("Father's age is negative.");
        }
        this.age = age;
    }
}

class Son extends Father
{
    public int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAgeException
    {
        super(fatherAge);
        if (sonAge < 0)
        {
            throw new WrongAgeException("Son's age is negative.");
        }
        if (sonAge >= fatherAge)
        {
            throw new WrongAgeException("Son's age is greater than or equal to Father's age.");
        }
        this.sonAge = sonAge;
    }
}

public class Familytree
{
    public static void main(String[] args)
    {
        try
        {
            Son son1 = new Son(40, 15);
            System.out.println("Father's age: " + son1.age);
            System.out.println("Son's age: " + son1.sonAge);
        }
    }
}

```



```

        Son son2 = new Son(50, 70);
        System.out.println("Father's age: " + son2.age);
        System.out.println("Son's age: " + son2.sonAge);

    }
    catch (WrongAgeException e)
    {
        System.out.println("Error: " + e.getMessage());
    }
    System.out.println("B VATSAL");
    System.out.println("1BM23CS061");
}
}
class WrongAgeException extends Exception
{
    public WrongAgeException(String message)
    {
        super(message);
    }
}

class Father
{
    protected int age;

    public Father(int age) throws WrongAgeException
    {
        if (age < 0)
        {
            throw new WrongAgeException("Father's age is negative.");
        }
        this.age = age;
    }
}

class Son extends Father
{
    public int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAgeException
    {
        super(fatherAge);
        if (sonAge < 0)
        {
            throw new WrongAgeException("Son's age is negative.");
        }
    }
}

```

```

    }
    if (sonAge >= fatherAge)
    {
        throw new WrongAgeException("Son's age is greater than or equal to Father's age.");
    }
    this.sonAge = sonAge;
}
}

```

```

public class Familytree
{
    public static void main(String[] args)
    {
        try
        {

            Son son1 = new Son(40, 15);
            System.out.println("Father's age: " + son1.age);
            System.out.println("Son's age: " + son1.sonAge);

            Son son2 = new Son(50, 70);
            System.out.println("Father's age: " + son2.age);
            System.out.println("Son's age: " + son2.sonAge);

        }
        catch (WrongAgeException e)
        {
            System.out.println("Error: " + e.getMessage());
        }
        System.out.println("B VATSAL");
        System.out.println("1BM23CS061");
    }
}

```

Output:

```

C:\Users\Vatsal\OneDrive\Documents\java\LAB-8>java Familytree.java
Father's age: 40
Son's age: 15
Error: Son's age is greater than or equal to Father's age.
B VATSAL
1BM23CS061

```

### **Program 9**

#### Threads

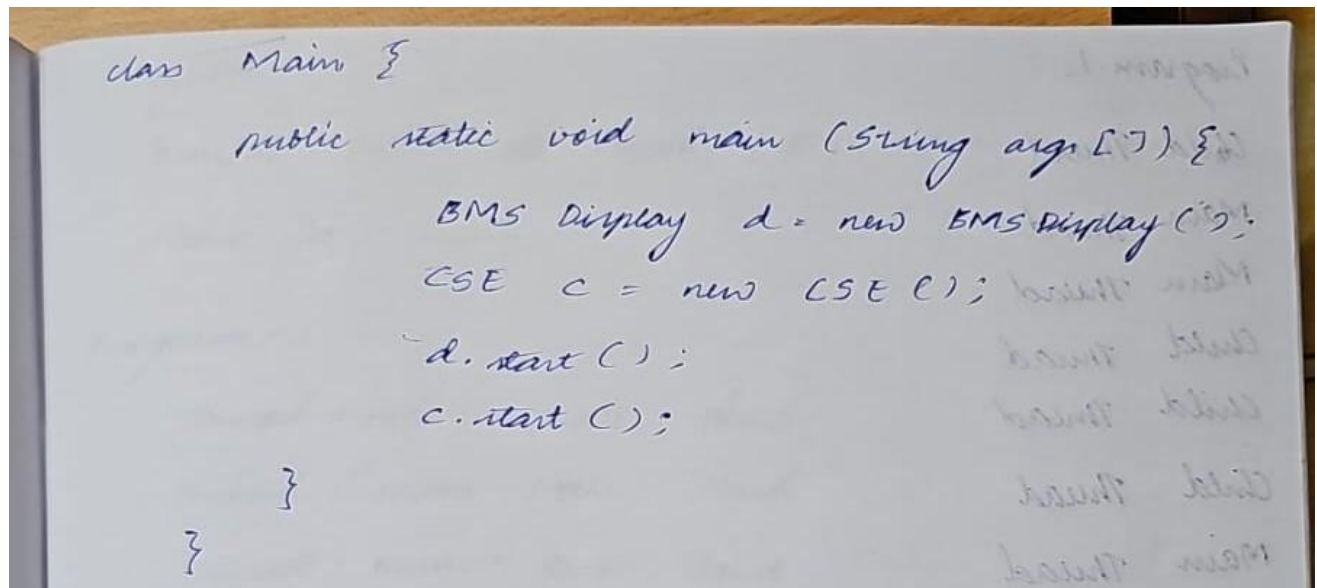
Write a program which creates two threads, one thread displaying “BMS College of Engineering” once every ten seconds and another displaying “CSE” once every two seconds.

Algorithm:

## Threads

```
class BMSDisplay extends Thread {  
    public void run() {  
        while (true) {  
            System.out.println("BMS College of Engineering");  
            try {  
                Thread.sleep(10000);  
            }  
            catch (InterruptedException e) {  
                System.out.println(e);  
            }  
        }  
    }  
}
```

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



Code:

```
class ThreadEx
{
    public static class BMSDisplayThread extends Thread
    {
        public void run()
        {
            int a=0;
            while (a<5)
            {
                System.out.println("BMS College of Engineering");
                try
                {
                    Thread.sleep(200);
                }
                catch (InterruptedException e)
                {
                    System.out.println(e);
                }
                a=a+1;
            }
        }
    }
}

public static class CSEDisplayThread extends Thread
{
    public void run()
    {
```

```

    int b=0;
    while (b<5)
    {
        System.out.println("CSE");
        try
        {
            Thread.sleep(200);
        }
        catch (Exception e)
        {
            System.out.println(e);
        }
        b=b+1;
    }
}
}
class Main
{
    public static void main(String[] args)
    {
        Thread bmsThread = new ThreadEx.BMSDisplayThread();
        Thread cseThread = new ThreadEx.CSEDisplayThread();
        bmsThread.start();
        cseThread.start();
        System.out.println("B VATSAL");
        System.out.println("1BM23CS061");
    }
}

```

Output:

```

C:\Users\Vatsal\OneDrive\Documents\java\LAB-9>java Main.java
B VATSAL
1BM23CS061
BMS College of Engineering
CSE
BMS College of Engineering
CSE
CSE
BMS College of Engineering
BMS College of Engineering
CSE
BMS College of Engineering
CSE

```



## **Program 10**

### GUI – Java Swing

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

1. create UI to divide 2 integers by  
boxes on clicking the divide  
button, result  
displayed in a square box. Display appropriate  
error message.

2. import javax.swing.\*;  
import java.awt.\*;  
~~import java.awt.\*;~~

3. class SwingDemo {  
 JFrame jfrm = new JFrame("Divident + Divisor");  
 jfrm.setSize(275, 150);  
 jfrm.setLayout(new FlowLayout());  
 jfrm.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  
 JLabel jlab = new JLabel("Enter the dividend  
and dividend");

~~JLabel jlab = new JLabel;~~

JTextField ajtf = new JTextField(8);  
JTextField bjtf = new JTextField(8);  
JButton button = new JButton("calculate");  
JLabel cr = new JLabel();  
JLabel alab = new JLabel();  
JLabel anslab = new JLabel();  
jfrm.add(cr);  
jfrm.add(jlab);  
jfrm.add(bjtf);  
jfrm.add(button);

```

ajtfm.add(alab);
bjtfm.add(blab);
ajrfm.add(anslab);

```

```

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

```

```

ajtf.add ActionListener(l);
bjtf.add ActionListener(l);
button.add ActionListener (new ActionListener() {
    try {
        int a = Integer.parseInt(ajtf.getText());
        int b = Integer.parseInt(bjtf.getText());
        int ans = a/b;
        alab.setText ("ln A = " + a);
        blab.setText ("ln B = " + b);
        ansLab.setText ("ln Ans = " + ans);
    }
    catch (NumberFormatException e) {
        alab.setText (" ");
        blab.setText (" ");
        ansLab.setText (" ");
        em.setText ("Enter only integers");
    }
});

```

```

catch (ArithmeticException e) {
    alab.setText(" ");
    blab.setText(" ");
    amlab.setText(" ");
    en.setText("B should be non zero");
}
}
};

ifrm.setVisible(true);
}

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

```

Code:

```
import javax.swing.*;
```

```

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class DivisionApp extends JFrame implements ActionListener
{
    private JTextField num1Field, num2Field, resultField;
    private JButton DivisionAppdeButton;

    public DivisionApp()
    {
        setTitle("Integer DivisionApp App");
        setLayout(new FlowLayout());
        setSize(300, 200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JLabel nameLabel = new JLabel("B VATSAL");
        JLabel usnLabel = new JLabel("1BM23CS061");

        JLabel num1Label = new JLabel("Num1:");
        num1Field = new JTextField(10);
        JLabel num2Label = new JLabel("Num2:");
        num2Field = new JTextField(10);
        JLabel resultLabel = new JLabel("Result:");
        resultField = new JTextField(10);
        resultField.setEditable(false);
        DivisionAppdeButton = new JButton("Divide");

        add(nameLabel);
        add(usnLabel);
        add(num1Label);
        add(num1Field);
        add(num2Label);
        add(num2Field);
        add(DivisionAppdeButton);
        add(resultLabel);
        add(resultField);

        DivisionAppdeButton.addActionListener(this);
    }

    @Override
    public void actionPerformed(ActionEvent e)
    {
        try

```

```

    {
        int num1 = Integer.parseInt(num1Field.getText());
        int num2 = Integer.parseInt(num2Field.getText());
        int result = num1 / num2;
        resultField.setText(String.valueOf(result));
    }
    catch (NumberFormatException ex)
    {
        JOptionPane.showMessageDialog(
            this,
            "Please enter valid integers.",
            "Input Error",
            JOptionPane.ERROR_MESSAGE
        );
    }
    catch (ArithmeticException ex)
    {
        JOptionPane.showMessageDialog(
            this,
            "Division by zero is not allowed.",
            "Arithmetic Error",
            JOptionPane.ERROR_MESSAGE
        );
    }
}

public static void main(String[] args)
{
    DivisionApp app = new DivisionApp();
    app.setVisible(true);
}
}

```

Output:



