

**VISVESVARAYA TECHNOLOGICAL
UNIVERSITY**

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT

on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

LAKSHITH P (1BM23CS164)

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 **LAKSHITH P (1BM23CS164)**, 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.

Ms. Ambuja K Assistant Professor Department of CSE, BMSCE	Dr. Seema patil Assistant Professor Department of CSE, BMSCE
---	--

Index

Sl. No.	Date	Experiment Title	Page No.
1	23/09/24	QUADRATIC EQUATIONS	4
2	07/10/24	SGPA CALCULATOR	8
3	14/10/24	BOOK DETAILS	16
4	21/10/24	AREA TRIANGLE RECTANGLE CIRCLE	21
5	28/10/24	BANK ACCOUNT	26
6	11/11/24	FINAL MARKS PROGRAM PACKAGES	33
7	28/11/24	FATHER AND SON EXCEPTION	38
8	28/11/24	MULTITHREADING	42
9	28/11/24	DIVISION GUI	45
10	28/11/24	A)PCFIXED &B)DEADLOCK	48

Github Link:

<https://github.com/LakshithP994/Lab-programs-ooj-lab>

Program 1

Implement Quadratic Equation

20/09/24

DATE: PAGE

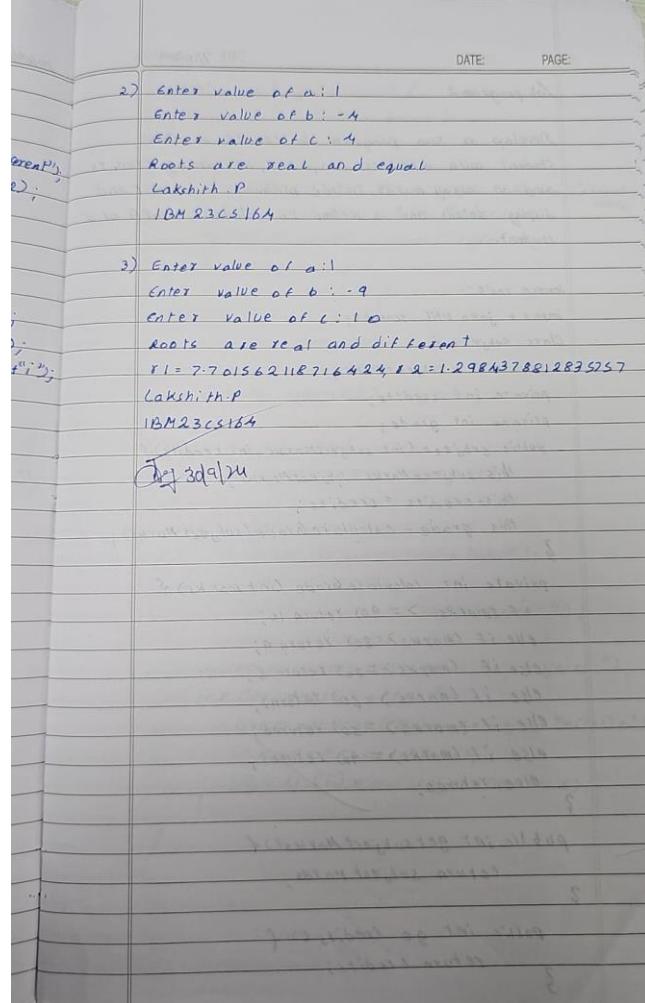
```
Lab program 1
Develop a java program that prints all real solutions to the quadratic equation ax2+bx+c=0. Read in a,b,c and use the quadratic formula. If the discriminant b2-4ac is negative, display a message stating that there are no real solutions.

import static java.lang.Math.sqrt;
import java.util.Scanner;
class quadratic {
    int a,b,c;
    double r1,r2,d;
    void input() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter value of a:");
        a=sc.nextInt();
        while(a==0) {
            System.out.println("Enter a non zero number for a:");
            a=sc.nextInt();
        }
        System.out.print("Enter value of b:");
        b = sc.nextInt();
        System.out.print("Enter value of c:");
        c = sc.nextInt();
        d=b*b-4*a*c;
    }
    void display() {
        if (d==0) {
            r1 = -b/(2*a);
            System.out.println("Roots are real and equal");
            System.out.println("Root : " + r1);
        }
    }
}
```

DATE: PAGE

```
else if (d>0) {
    r1 = (-b+sqrt(d))/(2*a);
    r2 = (-b-sqrt(d))/(2*a);
    System.out.println("Roots are real and different");
    System.out.println("r1 = " + r1 + ", r2 = " + r2);
}
else {
    r1 = -b/(2*a);
    r2 = sqrt(-d)/(2*a);
    System.out.println("Roots are imaginary");
    System.out.println("r1 = " + r1 + " + " + r2 + "i");
    System.out.println("r2 = " + r1 + " - " + r2 + "i");
}
public static void main(String args[]) {
    quadratic qe = new quadratic();
    qe.input();
    qe.display();
    System.out.println("Lakshith.P");
    System.out.println("IBM23CS164");
}

out put
1) Enter value of a:0
Enter a non zero number for a:
1
Enter value of b:2
Enter value of c:1
Roots are real and equal
Root : -1.0
Lakshith P
IBM23CS164
```



Code:

```
import static java.lang.Math.sqrt;  
  
import java.util.Scanner;  
  
  
class quadratic {  
  
    int a, b, c;  
  
    double r1, r2, d;  
  
  
    void input() {  
  
        Scanner sc = new Scanner(System.in);  
  
        System.out.print("Enter value of a: ");  
  
        a = sc.nextInt();
```

```

while (a == 0) {

    System.out.println("Enter a non-zero number for a:");

    a = sc.nextInt();

}

System.out.print("Enter value of b: ");

b = sc.nextInt();

System.out.print("Enter value of c: ");

c = sc.nextInt();

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

}

void display() {

    if (d == 0) {

        r1 = -b / (2.0 * a);

        System.out.println("Roots are real and equal");

        System.out.println("Root: " + r1);

    } else if (d > 0) {

        r1 = (-b + sqrt(d)) / (2.0 * a);

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

        System.out.println("Roots are real and different");
    }
}

```

```

        System.out.println("r1 = " + r1 + ", r2 = " + r2);

    } else {

        r1 = -b / (2.0 * a);

        r2 = sqrt(-d) / (2.0 * a);

        System.out.println("Roots are imaginary");

        System.out.println("r1 = " + r1 + " + " + r2 + "i");

        System.out.println("r2 = " + r1 + " - " + r2 + "i");

    }

}

```

```

public static void main(String[] args) {

    quadratic qe = new quadratic();

    qe.input();

    qe.display();

    System.out.println("Lakshith.P");

    System.out.println("1BM23CS164");

}

```

```

D:\1BM23CS164>java quadratic
Enter value of a: 1
Enter value of b: -9
Enter value of c: 10
Roots are real and different
r1 = 7.701562118716424, r2 = 1.2984378812835757
Lakshith.P
1BM23CS164

```

```

D:\1BM23CS164>java quadratic
Enter value of a: 1
Enter value of b: 2
Enter value of c: 1
Roots are real and equal
Root: -1.0
Lakshith.P
1BM23CS164

```

```

D:\1BM23CS164>java quadratic
Enter value of a: 1
Enter value of b: -4
Enter value of c: 4
Roots are real and equal
Root: 2.0
Lakshith.P
1BM23CS164

```

Program 2

SGPA CALCULATOR

Lab program 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.

Source code :-

```

import java.util.Scanner;
class Subject {
    private int subjectMarks;
    private int credits;
    private int grade;
    public Subject (int subjectMarks, int credits) {
        this.subjectMarks = subjectMarks;
        this.credits = credits;
        this.grade = calculateGrade(subjectMarks);
    }
    private int calculateGrade (int marks) {
        if (marks >= 90) return 10;
        else if (marks >= 80) return 9;
        else if (marks >= 70) return 8;
        else if (marks >= 60) return 7;
        else if (marks >= 50) return 6;
        else if (marks >= 40) return 5;
        else return 0;
    }
    public int getSubjectMarks() {
        return subjectMarks;
    }
    public int getcredits() {
        return credits;
    }
}

```

```

public int getGrade() {
    return grade;
}

public void display () {
    System.out.println ("subject Marks :" + subject
    Marks);
    System.out.println ("credits :" + credits);
    System.out.println ("Grade :" + grade);
}

class Student {
    private String usn;
    private String name;
    private Subject [] subjects;
    public Student () {
        subjects = new Subject [3];
    }
    void getDetails (Scanner sc) {
        System.out.print ("Enter USN : ");
        usn = sc.nextLine();
        System.out.print ("Enter Name : ");
        name = sc.nextLine();
        System.out.println ("Enter marks for 3 subjects : ");
        for (int i = 0; i < 3; i++) {
            System.out.print ("Marks for subject " + (i+1) + " : ");
            int Marks = sc.nextInt();
            subjects [i] = new Subject (Marks, 4);
        }
    }
}

```

```

DATE: PAGE: Sy
double calculateSAPAdt
double totalPoints=0;
double totalCredits=0;
for (Subject subject : subjects) {
    totalPoints += subject.getGrade() * subject.get
    credits();
    totalCredits += subject.getCredits();
}
return totalCredits > 0 ? totalPoints / totalCredits : 0;
}

void display() {
    System.out.println("USN: " + usn);
    System.out.println("Name: " + name);
    System.out.println("Subjects: ");
    for (Subject subject : subjects) {
        subject.display();
        System.out.println();
    }
    System.out.printf("SGPA: %.2f\n", calculateSAPAdt);
}

public class SAPACalculator {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Student[] students = new Student[3];
        for (int j = 0; j < 3; j++) {
            System.out.println("Enter the details of student " + (j + 1));
            students[j] = new Student();
            students[j].getDetails(sc);
        }
    }
}

```

DATE: PAGE:
 System.out.println("In Student Details");
 for (Student student : students) {
 student.display();
 }
 System.out.println();
 System.out.println("Lakshith P");
 System.out.println("IBM23CS164");
 sc.close();

Output
 Enter the details of student 1:
 Enter USN : IBM23CS160
 Enter Name: Arvind
 Enter marks for 3 subjects:
 Marks for subject 1 : 20
 Marks for subject 2 : 30
 Marks for subject 3 : 35
 Enter the details of student 2:
 Enter USN : IBM23CS164
 Enter name: Lakshith
 Enter marks of 3 subjects
 Marks for subject 1 : 89
 Marks for subject 2 : 90
 Marks for subject 3 : 95
 Enter the details of student 3:
 Enter USN : IBM23CS162
 Enter name: Raj
 Enter marks for 3 subjects:
 Marks for subject 1 : 50
 Marks for subject 2 : 67
 Marks for subject 3 : 70

DATE:
 Student Details:
 USN: IBM23CS160
 Name: Arvind
 Subjects:
 Subject Marks: 20
 Credits: 3
 Grade: A
 Subject Marks: 30
 Credits: 4
 Grade: A
 Subject Marks: 25
 Credits: 4
 Grade: A
 Subject Marks: 30
 Credits: 4
 Grade: A
 SGPA: 0.00

USN: IBM23CS164
 Name: Lakshith
 Subjects:
 Subject Marks: 89
 Credits: 4
 Grade: A
 Subject Marks: 90
 Credits: 4
 Grade: A
 Subject Marks: 95
 Credits: 4
 Grade: A
 SGPA: 9.67

LAKSHITH P
 IBM23CS164
 Date: 9/10/24

CODE:

```
import java.util.Scanner;

class Subject {

    private int subjectMarks;
    private int credits;
    private int grade;

    public Subject(int subjectMarks, int credits) {
        this.subjectMarks = subjectMarks;
        this.credits = credits;
        this.grade = calculateGrade(subjectMarks);
    }

    private int calculateGrade(int marks) {
        if (marks >= 90) return 10;
        else if (marks >= 80) return 9;
        else if (marks >= 70) return 8;
        else if (marks >= 60) return 7;
        else if (marks >= 50) return 6;
        else if (marks >= 40) return 5;
        else return 0;
    }
}
```

```
public int getSubjectMarks() {  
    return subjectMarks;  
}  
  
public int getCredits() {  
    return credits;  
}  
  
public int getGrade() {  
    return grade;  
}  
  
public void display() {  
    System.out.println("Subject Marks: " + subjectMarks);  
    System.out.println("Credits: " + credits);  
    System.out.println("Grade: " + grade);  
}  
  
}  
  
class Student {  
    private String usn;  
    private String name;
```

```
private Subject[] subjects;

public Student() {
    subjects = new Subject[3];
}

void getDetails(Scanner sc) {
    System.out.print("Enter USN: ");
    usn = sc.next();
    System.out.print("Enter Name: ");
    name = sc.next();

    System.out.println("Enter marks for 3 subjects:");
    for (int i = 0; i < 3; i++) {
        System.out.print("Marks for subject " + (i + 1) + ": ");
        int marks = sc.nextInt();
        subjects[i] = new Subject(marks, 4);
    }
}

double calculateSGPA() {
    double totalPoints = 0;
    double totalCredits = 0;
```

```

for (Subject subject : subjects) {

    totalPoints += subject.getGrade() * subject.getCredits();

    totalCredits += subject.getCredits();

}

return totalCredits > 0 ? totalPoints / totalCredits : 0;
}

void display() {

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

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

    System.out.println("Subjects:");

    for (Subject subject : subjects) {

        subject.display();

        System.out.println();

    }

    System.out.printf("SGPA: %.2f%n", calculateSGPA());
}

}

```

```

public class SGPAcalculator {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        Student[] students = new Student[3];

```

```
for (int j = 0; j < 3; j++) {  
    System.out.println("Enter the details of student " + (j + 1) + ":");  
    students[j] = new Student();  
    students[j].getDetails(sc);  
}  
  
System.out.println("\nStudent Details:");  
for (Student student : students) {  
    student.display();  
    System.out.println();  
}  
System.out.println("Lakshith P");  
System.out.println("1BM23CS164");  
  
sc.close();  
}  
}
```

OUT PUT:

```
Enter the details of student 1:  
Enter USN: 1BM23CS160  
Enter Name: Arvind  
Enter marks for 3 subjects:  
Marks for subject 1: 20  
Marks for subject 2: 30  
Marks for subject 3: 35  
Enter the details of student 2:  
Enter USN: 1BM23CS164  
Enter Name: LAKSHITH  
Enter marks for 3 subjects:  
Marks for subject 1: 89  
Marks for subject 2: 90  
Marks for subject 3: 95  
Enter the details of student 3:  
Enter USN: 1BMCS162  
Enter Name: RAJ  
Enter marks for 3 subjects:  
Marks for subject 1: 50  
Marks for subject 2: 67  
Marks for subject 3: 70  
  
Student Details:  
USN: 1BM23CS160  
Name: Arvind  
Subjects:  
Subject Marks: 20  
Credits: 4  
Grade: 0  
  
Subject Marks: 30  
Credits: 4  
Grade: 0  
  
Subject Marks: 35  
Credits: 4  
Grade: 0  
  
SGPA: 0.00  
  
USN: 1BM23CS164  
Name: LAKSHITH  
Subjects:  
Subject Marks: 89  
Credits: 4  
Grade: 9  
  
Subject Marks: 90  
Credits: 4  
Grade: 10  
  
Subject Marks: 95  
Credits: 4  
Grade: 10  
  
SGPA: 9.67  
  
USN: 1BMCS162  
Name: RAJ  
Subjects:  
Subject Marks: 50  
Credits: 4  
Grade: 6  
  
Subject Marks: 67  
Credits: 4  
Grade: 7  
  
Subject Marks: 70  
Credits: 4  
Grade: 8  
  
SGPA: 7.00  
  
Lakshith P  
1BM23CS164
```

Program 3

BOOK LIST

DATE: 14/08/24 PAGE:

Lab program 3

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

source code

```

import java.util.Scanner;
class books {
    String name;
    String author;
    int price, numPages;
    books(String name, String author, int price, int numPages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }
    public String to_string() {
        String name, author, price, numPages;
        name = "Book name:" + this.name + "\n";
        author = "Author name:" + this.author + "\n";
        price = "Price:" + this.price + "\n";
        numPages = "Number of pages:" + this.numPages + "\n";
        return name + author + price + numPages;
    }
}

```

DATE: PAGE:

```

public static void main(String args[]) {
    Scanner s = new Scanner(System.in);
    int n;
    String name;
    String author;
    int price;
    int numPages;
    System.out.println("Enter the number of books:");
    n = s.nextInt();
    books b[];
    b = new books[n];
    for (int i = 0; i < n; i++) {
        System.out.println("Enter the book " + (i + 1) + " name:");
        name = s.nextLine();
        System.out.println("Enter the book " + (i + 1) + " author:");
        author = s.nextLine();
        System.out.println("Enter the book " + (i + 1) + " price:");
        price = s.nextInt();
        System.out.println("Enter the book " + (i + 1) + " number of pages:");
        numPages = s.nextInt();
        b[i] = new books(name, author, price, numPages);
    }
    for (int i = 0; i < n; i++) {
        b[i].toString();
        System.out.println(b[i]);
    }
    System.out.println("Lakshith P");
    System.out.println("IBM23CS184");
}

```

DATE: PAGE:

Output

Enter the number of books:
2

Enter the book 1 name:
percy

Enter the book 1 author:
rick

Enter the book 1 price:
300

Enter the book 1 number of pages:
500

Enter the book 2 name:
don

Enter the book 2 author:
miguel

Enter the book 2 price:
500

Enter the book 2 number of pages:
350

Book name: percy
Author name: rick
price: 300
Number of pages: 500

Book name: don
Author name: miguel
price: 500
Number of pages: 350

Lakshith P
1 BM23CS164
14/10/24

CODE:

```

import java.util.Scanner;

class books{

    String name;
    String author;
    int price,numPages;

    books(String name,String author,int price,int numPages)

    {
        this.name=name;
    }
}

```

```
this.author=author;  
this.price=price;  
this.numPages=numPages;  
}  
  
public String toString()  
{  
    String name,author,price,numPages;  
    name="Book name:"+ this.name +"\n";  
    author="Author name:"+ this.author +"\n";  
    price="Price:"+ this.price +"\n";  
    numPages="Number of pages:"+ this.numPages +"\n";  
    return name+author+price+numPages;  
}  
  
public static void main(String args[])  
{  
    Scanner s=new Scanner(System.in);  
    int n;  
    String name;  
    String author;  
    int price;  
    int numPages;  
    System.out.println("Enter the number of books;");
```

```

n=s.nextInt();

books b[];

b=new books[n];

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

{

    System.out.println("Enter the book "+(i+1)+"name:");

    name=s.next();

    System.out.println("Enter the book "+(i+1)+" author:");

    author=s.next();

    System.out.println("Enter the book "+(i+1)+" price:");

    price=s.nextInt();

    System.out.println("Enter the book "+(i+1)+" number of pages:");

    numPages=s.nextInt();

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

}

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

{

    b[i].toString();

    System.out.println(b[i]);

}

System.out.println("Lakshith P");

System.out.println("1BM23CS164");

}}

```

OUT PUT:

```
D:\1BM23CS164>java books
Enter the number of books;
2
Enter the book 1name:
percy
Enter the book 1 author:
rick
Enter the book 1 price:
300
Enter the book 1 number of pages:
500
Enter the book 2name:
don
Enter the book 2 author:
miguel
Enter the book 2 price:
500
Enter the book 2 number of pages:
350
Book name:percy
Author name:rick
Price:300
Number of pages:500
```

```
Book name:don
Author name:miguel
Price:500
Number of pages:350
```

Lakshith P
1BM23CS164

Program 4

AREA OF TRAINGLE RECTANGLE AND CIRCLE

Lab program 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 {
    double dim1;
    double dim2;
    public Shape(double dim1, double dim2) {
        this.dim1 = dim1;
        this.dim2 = dim2;
    }
    abstract void printArea();
}
class Rectangle extends Shape {
    public Rectangle(double length, double breadth) {
        super(length, breadth);
    }
    void printArea() {
        double area = dim1 * dim2;
        System.out.println("Area of Rectangle = " + area);
    }
}
class Triangle extends Shape {
    public Triangle(double base, double height) {
        super(base, height);
    }
}
```

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

class Circle extends Shape {
    public Circle(double radius) {
        super(radius, 0);
    }
}
```

```
void printArea() {
    double area = 3.14 * dim1 * dim1;
    System.out.println("Area of circle = " + area);
}
}

public class Area {
    public static void main(String args[]) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter the dimensions of rectangle (length and breadth): ");
        double rectLength = scanner.nextDouble();
        double rectBreadth = scanner.nextDouble();
        Shape rectangle = new Rectangle(rectLength, rectBreadth);
        rectangle.printArea();
        System.out.println("Enter the dimensions of the triangle (base and height): ");
        double triBase = scanner.nextDouble();
        double triHeight = scanner.nextDouble();
        Shape triangle = new Triangle(triBase, triHeight);
        triangle.printArea();
    }
}
```

```
System.out.println("Enter the dimensions of the circle (radius): ");
double cirRadius = scanner.nextDouble();
Shape circle = new Circle(cirRadius);
```

DATE: PAGE:

```

        circle.printArea();
        System.out.println("Lakshith P");
        System.out.println("IBM23LS164");
        }
    }

    Output
    Enter the dimensions of rectangle (length and breadth):
    2 4
    Area of Rectangle=8.0
    Enter the dimensions of the triangle (base and height):
    2 5
    Area of Triangle=5.0
    Enter the dimensions of the circle (radius):
    6
    Area of Circle = 113.03999999999998
    Lakshith P
    IBM23LS164
    
```

Date 20/10

CODE:

```

import java.util.Scanner;

abstract class shape{
    double dim1;
    double dim2;
    public shape(double dim1,double dim2){
        this.dim1=dim1;
        this.dim2=dim2;
    }
    abstract void printArea();
}
    
```

```
}

class Rectangle extends shape{

public Rectangle(double length, double breadth){

super(length ,breadth);

}

void printArea(){

double area=dim1*dim2;

System.out.println("Area of Rectangle="+area);

}

}

class Triangle extends shape{

public Triangle(double base,double height){

super(base,height);

}

void printArea(){

double area=0.5*dim1*dim2;

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

}

}

class Circle extends shape{

public Circle(double radius){

super(radius,0);

}
```

```
void printArea(){

    double area=3.14*dim1*dim1;

    System.out.println("Area of Circle="+area);

}

}

public class Area{
```

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

    Scanner scanner=new Scanner(System.in);

    System.out.println("Enter the dimensions of rectangle(length and breadth):");

    double rectlength=scanner.nextDouble();

    double rectbreadth=scanner.nextDouble();

    shape rectangle=new Rectangle(rectlength,rectbreadth);

    rectangle.printArea();
```

```
System.out.println("Enter the dimensions of the triangle (base and height):");

double tribase=scanner.nextDouble();

double triheight=scanner.nextDouble();

shape triangle=new Triangle(tribase,triheight);

triangle.printArea();
```

```
System.out.println("Enter the dimensions of the circle (radius):");

double cirradius=scanner.nextDouble();

shape circle=new Circle(cirradius);
```

```
circle.printArea();

System.out.println("Lakshith P");

System.out.println("1BM23CS164");

}

}
```

OUT PUT:

```
D:\1BM23CS164>java Area
Enter the dimensions of rectangle(length and breadth):
6 8
Area of Rectangle=48.0
Enter the dimensions of the triangle (base and height):
34 56
Area of Triangle=952.0
Enter the dimensions of the circle (radius):
45
Area of Circle=6358.500000000001
Lakshith P
1BM23CS164
```

```
C:\Users\student>d:

D:\>cd 1BM23CS164

D:\1BM23CS164>javac Area.java

D:\1BM23CS164>java Area
Enter the dimensions of rectangle(length and breadth):
2 4
Area of Rectangle=8.0
Enter the dimensions of the triangle (base and height):
2 5
Area of Triangle=5.0
Enter the dimensions of the circle (radius):
6
Area of Circle=113.0399999999999
Lakshith P
1BM23CS164
```

Program 5

AREA OF TRAINGLE RECTANGLE AND CIRCLE

LAB PROGRAM 5
 Develop a Java program to create a class Bank that maintains two kinds of accounts 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-account and sv-account to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

- Accept deposit from customer and update the balance.
- Display the balance.
- Compute and deposit interest.
- Permit withdrawal and update the balance.
- Check for the minimum balance, impose penalty if necessary and update the balance.

```

import java.util.Scanner;
abstract class Account {
    String customerName;
    String accountNumber;
    double balance;
    public Account (String customerName, String accountNumber,
                    double initialBalance) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.balance = initialBalance;
    }
}
  
```

DATE: PAGE:

public abstract void deposit (double amount);
 public abstract void withdraw (double amount);
 public abstract void displayBalance();
}
class SavingsAccount extends Account {
 public SavingsAccount (String customerName, String accountNumber, double initialBalance) {
 super (customerName, accountNumber, initialBalance);
 }
 public void deposit (double amount) {
 balance += amount;
 System.out.println ("Deposited: " + amount);
 }
 public void withdraw (double amount) {
 if (amount <= balance) {
 balance -= amount;
 System.out.println ("Withdraw: " + amount);
 } else {
 System.out.println ("Insufficient balance.");
 }
 }
 public void computeAndDepositInterest() {
 double interest = balance * 0.04;
 balance += interest;
 System.out.println ("Interest deposited: " + interest);
 }
 public void displayBalance() {
 System.out.println ("Saving Account Balance: " + balance);
 }
}

```

class CurrentAccount extends Account {
    private double minimumBalance;
    public CurrentAccount(String customerName, String accountNumber, double initialBalance, double minimumBalance) {
        super(customerName, accountNumber, initialBalance);
        this.minimumBalance = minimumBalance;
    }
    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited: " + amount);
        checkMinimumBalance();
    }
    void checkMinimumBalance() {
        if (balance < 1000) {
            balance -= 50;
            System.out.println("Minimum balance not maintained. Service charge imposed: 50");
        }
    }
    public void displayBalance() {
        System.out.println("Current Account Balance: " + balance);
    }
}

public class Bank1 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        while (true) {
            System.out.println("Enter 1 for Saving Account and 2 for Current Account 3 to Exit:");
            String input1 = scanner.nextLine();
            if (input1 == "1") {
                System.out.print("Enter customer name for SA:");
            }
        }
    }
}

```

```

DATE: PAGE:
string savingName=scanner.nextLine();
System.out.println("Enter account number");
String accountNumber = scanner.nextLine();
double initialSavingsBalance=0;
SavingsAccount savingsAccount = new SavingsAccount(
    savingsName, accountNumber, initialSavingsBalance);
handleSavingsAccount(scanner, savingsAccount);
else if (input == "2") {
    System.out.println("Enter customer name for CA:");
    String currentName=scanner.nextLine();
    System.out.print("Enter account number:");
    String accountNumber2=scanner.nextLine();
    double minimumBalance=1000;
    CurrentAccount currentAccount = new CurrentAccount(
        currentName, accountNumber2, initialCurrentBalance,
        minimumBalance);
} else {
    System.out.println("Exited");
    break;
}
static void handleCurrentAccount(Scanner scanner,
    CurrentAccount account) {
    while(true) {
        System.out.println("Enter 1 for deposit 2 for withdrawl
        3 for balance, or any other number to exit:");
        String input = scanner.nextLine();
        if (input == "1") {
            System.out.println("Enter amount to be deposited:");
            account.deposit(scanner.nextDouble());
            scanner.nextLine();
        } else if (input == "2") {
            System.out.print("Enter amount to withdrawl:");
        }
    }
}

```

DATE: _____ PAGE: _____

```

3 account.withdraw(scanner.nextDouble());
else if (input == "3") {
    account.displayBalance();
}
else {
    break;
}
333
(May 10/24

output:
Enter 1 for savings account 2 for current account 3 to exit:
1
Enter customer name: sa
Enter account number: 12
Enter 1 for deposit, 2 for withdraw, 3 for balance, 4 for interest
or any other number to exit:
1
Deposited 1000
2
300
4
28
3
728
6
Enter 2
Enter customer name: ca
Enter Acc no: 12
1
1000.0
2
5000
3
5000

```

CODE:

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

```

abstract class Account {
    String customerName;
    String accountNumber;
    double balance;

    public Account(String customerName, String accountNumber, double initialBalance) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.balance = initialBalance;
    }

    public abstract void deposit(double amount);
    public abstract void withdraw(double amount);
    public abstract void displayBalance();
}
```

```

class SavingsAccount extends Account {
    public SavingsAccount(String customerName, String accountNumber, double initialBalance) {
        super(customerName, accountNumber, initialBalance);
    }

    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited: " + amount);
    }

    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrew: " + amount);
        } else {
            System.out.println("Insufficient balance.");
        }
    }

    public void computeAndDepositInterest() {
        double interest = balance * 0.04;
        balance += interest;
        System.out.println("Interest deposited: " + interest);
    }

    public void displayBalance() {
        System.out.println("Savings Account Balance: " + balance);
    }
}

class CurrentAccount extends Account {
    private double minimumBalance;

    public CurrentAccount(String customerName, String accountNumber, double initialBalance, double minimumBalance) {
        super(customerName, accountNumber, initialBalance);
        this.minimumBalance = minimumBalance;
    }

    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited: " + amount);
        checkMinimumBalance();
    }

    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrew: " + amount);
        }
    }
}

```

```

        checkMinimumBalance();
    } else {
        System.out.println("Insufficient balance.");
    }
}

private void checkMinimumBalance() {
    if (balance < 1000) {
        balance -= 50;
        System.out.println("Minimum balance not maintained. Service charge imposed: 50");
    }
}

public void displayBalance() {
    System.out.println("Current Account Balance: " + balance);
}
}

public class Bank1 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        while (true) {
            System.out.println("Enter 1 for Savings Account and 2 for Current Account 3 to Exit:");
            String input1 = scanner.nextLine();
            if (input1.equals("1")) {
                System.out.print("Enter customer name for Savings Account: ");
                String savingsName = scanner.nextLine();
                System.out.print("Enter account number: ");
                String accountNumber = scanner.nextLine();
                double initialSavingsBalance = 0;
                SavingsAccount savingsAccount = new SavingsAccount(savingsName, accountNumber,
initialSavingsBalance);
                handleSavingsAccount(scanner, savingsAccount);
            } else if (input1.equals("2")) {
                System.out.print("Enter customer name for Current Account: ");
                String currentName = scanner.nextLine();
                System.out.print("Enter account number: ");
                String accountNumber2 = scanner.nextLine();
                double initialCurrentBalance = 0;
                double minimumBalance = 1000;
                CurrentAccount currentAccount = new CurrentAccount(currentName, accountNumber2,
initialCurrentBalance, minimumBalance);
                handleCurrentAccount(scanner, currentAccount);
            } else {
                System.out.println("Exited");
                break;
            }
        }
        scanner.close();
    }
}

```

```

private static void handleSavingsAccount(Scanner scanner, SavingsAccount account) {
    while (true) {
        System.out.println("Enter 1 for deposit, 2 for withdrawal, 3 for balance, 4 for interest, or any other number to exit:");
        String input = scanner.nextLine();
        if (input.equals("1")) {
            System.out.print("Enter amount to deposit into Savings Account: ");
            account.deposit(scanner.nextDouble());
            scanner.nextLine();
        } else if (input.equals("2")) {
            System.out.print("Enter amount to withdraw from Savings Account: ");
            account.withdraw(scanner.nextDouble());
            scanner.nextLine();
        } else if (input.equals("3")) {
            account.displayBalance();
        } else if (input.equals("4")) {
            account.computeAndDepositInterest();
        }
        else {
            break;
        }
    }
}

```

```

private static void handleCurrentAccount(Scanner scanner, CurrentAccount account) {
    while (true) {
        System.out.println("Enter 1 for deposit, 2 for withdrawal, 3 for balance, or any other number to exit:");
        String input = scanner.nextLine();
        if (input.equals("1")) {
            System.out.print("Enter amount to deposit into Current Account: ");
            account.deposit(scanner.nextDouble());
            scanner.nextLine();
        } else if (input.equals("2")) {
            System.out.print("Enter amount to withdraw from Current Account: ");
            account.withdraw(scanner.nextDouble());
            scanner.nextLine();
        } else if (input.equals("3")) {
            account.displayBalance();
        } else {
            break;
        }
    }
}

```

OUT PUT:

```
C:\Users\STUDENT>d:  
D:>cd 1BM23CS164  
D:\1BM23CS164>javac Bank1.java  
D:\1BM23CS164>java Bank1  
Enter 1 for Savings Account and 2 for Current Account 3 to Exit:  
1  
Enter customer name for Savings Account: sa  
Enter account number: 12  
Enter 1 for deposit, 2 for withdrawal, 3 for balance, 4 for interest,or any other number to exit:  
1  
Enter amount to deposit into Savings Account: 1000  
Deposited: 1000.0  
Enter 1 for deposit, 2 for withdrawal, 3 for balance, 4 for interest,or any other number to exit:  
2  
Enter amount to withdraw from Savings Account: 300  
Withdrew: 300.0  
Enter 1 for deposit, 2 for withdrawal, 3 for balance, 4 for interest,or any other number to exit:  
4  
Interest deposited: 28.0  
Enter 1 for deposit, 2 for withdrawal, 3 for balance, 4 for interest,or any other number to exit:  
3  
Savings Account Balance: 728.0  
Enter 1 for deposit, 2 for withdrawal, 3 for balance, 4 for interest,or any other number to exit:  
4  
Interest deposited: 29.12  
Enter 1 for deposit, 2 for withdrawal, 3 for balance, 4 for interest,or any other number to exit:  
8  
Enter 1 for Savings Account and 2 for Current Account 3 to Exit:  
2  
Enter customer name for Current Account: ca  
Enter account number: 12  
Enter 1 for deposit, 2 for withdrawal, 3 for balance, or any other number to exit:  
1  
Enter amount to deposit into Current Account: 10000  
Deposited: 10000.0  
Enter 1 for deposit, 2 for withdrawal, 3 for balance, or any other number to exit:  
5000  
Enter 1 for Savings Account and 2 for Current Account 3 to Exit:  
3  
Exited
```

Program 6

FINAL MARKS CALCULATOR

DATE: _____ PAGE: _____

Lab program 6

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

```

package cie;
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;
    }
    public void display () {
        System.out.println ("USN:" + usn);
        System.out.println ("Name:" + name);
        System.out.println ("semester:" + sem);
    }
}

```

DATE: _____ PAGE: _____

```

package cie;
public class Internal extends Student {
    public int [] internalMarks = new int [5];
    public Internal (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 = new int [5];
    public External (String usn, String name, int sem, int [] seeMarks) {
        super (usn, name, sem);
        this.seeMarks = seeMarks;
    }
}

import cie.*;
import see.*;
import java.util.Scanner;
public class FinalMarksCalculator {
    public static void main (String [] args) {
        Scanner scanner = new Scanner (System.in);
        System.out.print ("Enter number of students:");
        int n = scanner.nextInt ();
        Internal [] internalStudents = new Internal [n];
        External [] externalStudents = new External [n];
        for (int i = 0; i < n; i++) {
            System.out.println ("USN:");
            String usn = scanner.next ();

```

```

DATE: PAGE:
system.out.print("Name: ");
String name = scanner.nextLine();
system.out.print("semester: ");
int sem = scanner.nextInt();
int[] internalMarks = new int[5];
system.out.println("Enter 5 internal marks (out of 50):");
for (int j = 0; j < 5; j++) {
    internalMarks[j] = scanner.nextInt();
}
internalStudents[i] = new InternalStudents(usn, name, sem,
internalMarks);
int[] secMarks = new int[5];
System.out.println("Final marks in 5 courses:");
for (int mark : finalMarks)
{
    system.out.print(mark + " ");
}
system.out.println();
}
Scanner.close();
}

```

Output

Enter number of students : 2

Enter details for student 1 :

USN: 123

Name: LUCK

Semester: 3

Enter 5 internal marks (out of 50);

34
43
45
43
45

DATE: PAGE:

Enter 5 sec marks (out of 100);

76
78
87
98
87

Enter details for student 2:

USN: 321

Name: BUCK

Semester: 3

Enter 5 internal marks (out of 50):

23
21
12
32
38

Enter 5 sec marks (out of 100):

54
56
78
62
87

~~Final marks of students:~~

~~Details of student 1:~~

USN: 123

Name: LUCK

Semester: 3

Final Marks in 5 Courses

72 82 88 92 88

~~Details of student 2:~~

USN: 321

Name: BUCK

Semester: 3

Final Marks in 5 Courses:

50 49 50 66 75

CODE:

INTERNALS.JAVA:

```
package cie;
```

```
public class Internals extends Student {  
    public int[] internalMarks = new int[5];  
  
    public Internals(String usn, String name, int sem, int[] internalMarks) {  
        super(usn, name, sem);  
        this.internalMarks = internalMarks;  
    }  
}
```

STUDENT.JAVA

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

EXTERNAL.JAVA:

```
package see;  
  
import cie.Student;  
  
public class External extends Student {  
    public int[] seeMarks = new int[5];  
  
    public External(String usn, String name, int sem, int[] seeMarks) {  
        super(usn, name, sem);  
        this.seeMarks = seeMarks;  
    }  
}
```

MAIN CODE:

```

import cie.*;
import see.*;
import java.util.Scanner;

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

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

        Internals[] internalStudents = new Internals[n];
        External[] externalStudents = new External[n];

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

            System.out.print("USN: ");
            String usn = scanner.next();
            System.out.print("Name: ");
            String name = scanner.next();
            System.out.print("Semester: ");
            int sem = scanner.nextInt();

            int[] internalMarks = new int[5];
            System.out.println("Enter 5 internal marks(Out of 50): ");
            for (int j = 0; j < 5; j++) {
                internalMarks[j] = scanner.nextInt();
            }
            internalStudents[i] = new Internals(usn, name, sem, internalMarks);

            int[] seeMarks = new int[5];
            System.out.println("Enter 5 SEE marks(Out of 100): ");
            for (int j = 0; j < 5; j++) {
                seeMarks[j] = scanner.nextInt();
            }
            externalStudents[i] = new External(usn, name, sem, seeMarks);
        }

        System.out.println("\nFinal Marks of Students:");
        for (int i = 0; i < n; i++) {
            System.out.println("\nDetails of Student " + (i + 1) + ":");

            internalStudents[i].display();

            int[] finalMarks = new int[5];
            for (int j = 0; j < 5; j++) {
                finalMarks[j] = internalStudents[i].internalMarks[j] + (externalStudents[i].seeMarks[j] / 2);
            }

            System.out.println("Final Marks in 5 Courses:");
        }
    }
}

```

```

        for (int mark : finalMarks)
    {
        System.out.print(mark + " ");
    }
    System.out.println();
}

scanner.close();
}
}

```

OUT PUT:

```

Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\STUDENT>d:
D:\>cd lbm23cs164
D:\lbm23cs164>javac cie/*.java see/*.java FinalMarksCalculator.java
D:\lbm23cs164>java FinalMarksCalculator
Enter number of students: 2
Enter details for Student 1:
USN: 123
Name: luck
Semester: 3
Enter 5 internal marks(Out of 50):
34
43
45
43
45
Enter 5 SEE marks(Out of 100):
76
78
87
98
87
Enter details for Student 2:
USN: 321
Name: buck
Semester: 3
Enter 5 internal marks(Out of 50):
23
21
12
32
32
Enter 5 SEE marks(Out of 100):
54
56
76
68
87

Final Marks of Students:

Details of Student 1:
USN: 123
Name: luck
Semester: 3
Final Marks in 5 Courses:
72 82 88 92 88

Details of Student 2:
USN: 321
Name: buck
Semester: 3
Final Marks in 5 Courses:
50 49 50 66 75

```

Program 7

FATHER SON AGE EXCEPTION

DATE: PAGE:

SUB PROGRAM ?

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

```

import java.util.Scanner;
class wrongAge {
    super ("Age Error");
}
public wrongAge (String message) {
    super (message);
}

class father {
    protected int fatherAge;
    public father () throws wrongAge {
        Scanner s = new Scanner (System.in);
        System.out.print ("Enter father's age : ");
        fatherAge = s.nextInt();
        if (fatherAge < 0) {
            throw new wrongAge ("Age cannot be negative");
        }
    }
    public void display () {
        System.out.println ("Father's Age :" + fatherAge);
    }
}

class son extends father {
    private int sonAge;
}

```

DATE: PAGE:

```

public son () throws wrongAge {
    Scanner s = new Scanner (System.in);
    System.out.print ("Enter son's age : ");
    sonAge = s.nextInt();
    if (sonAge < 0) {
        throw new wrongAge ("Age cannot be negative");
    }
    if (sonAge >= fatherAge) {
        throw new wrongAge ("Son's age cannot be greater than or equal to father's age");
    }
}

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

public class AgeValidationDemo {
    public static void main (String [] args) {
        try {
            son son = new son ();
            son.display ();
        } catch (wrongAge e) {
            System.out.println ("Exception : " + e.getMessage ());
        }
    }
}

```

11/11

Output:-
Enter father's age : 50
Enter son's age : 20
father's Age : 50
son's Age : 20

Enter father's age : 23
Enter son's age : 43
Exception: son's age cannot be greater than or
equal to father's age

Enter father's age : 22
Enter son's age : 22
Exception: son's age cannot be greater than or
equal to father's age

Q
1/11

CODE:

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

```
class WrongAge extends Exception {  
    public WrongAge() {  
        super("Age Error");  
    }
```

```
    public WrongAge(String message) {  
        super(message);  
    }  
}
```

```
class Father {  
    protected int fatherAge;
```

```

public Father() throws WrongAge {
    Scanner s = new Scanner(System.in);
    System.out.print("Enter Father's age: ");
    fatherAge = s.nextInt();

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

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

class Son extends Father {
    private int sonAge;

    public Son() throws WrongAge {
        super();
        Scanner s = new Scanner(System.in);
        System.out.print("Enter Son's age: ");
        sonAge = s.nextInt();

        if (sonAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
        if (sonAge >= fatherAge) {
            throw new WrongAge("Son's age cannot be greater than or equal to Father's age");
        }
    }

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

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

```

OUT PUT:

```
C:\Users\LAKSHITH.P>cd..  
C:\Users>cd..  
C:\>cd 3rd sem java/LAB PROGRAM 7  
C:\3rd sem java\LAB PROGRAM 7>javac AgeValidationDemo.java  
C:\3rd sem java\LAB PROGRAM 7>java AgeValidationDemo  
Enter Father's age: 50  
Enter Son's age: 20  
Father's Age: 50  
Son's Age: 20  
  
C:\3rd sem java\LAB PROGRAM 7>java AgeValidationDemo  
Enter Father's age: 23  
Enter Son's age: 43  
Exception: Son's age cannot be greater than or equal to Father's age  
  
C:\3rd sem java\LAB PROGRAM 7>java AgeValidationDemo  
Enter Father's age: 22  
Enter Son's age: 22  
Exception: Son's age cannot be greater than or equal to Father's age
```

Program 8

MULTI THREADING

LAB Program 8

DATE:

PAGE:

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

```
class collegeThread extends Thread {  
    @Override  
    public void run() {  
        try {  
            while (true) {  
                System.out.println("BMS college of  
                Engineering");  
                Thread.sleep(10000);  
            }  
        } catch (InterruptedException e) {  
            System.out.println("College Thread interrupted."  
                + e.getMessage());  
        }  
    }  
}
```

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

```
public class MultiThreadDemo {  
    public static void main (String [] args) {  
        CollegeThread collegeThread = new CollegeThread();  
        DepartmentThread departmentThread = new  
        DepartmentThread();  
        collegeThread.start();  
        departmentThread.start();  
    }  
}
```

OUTPUT

BMS college of engineering

CSE

CSE

CSE

CSE

BMS college of engineering

CSE

CSE

CSE

CSE

PAGE:

CODE:

```
class CollegeThread extends Thread {  
    public void run() {  
        try {  
            while (true) {  
                System.out.println("BMS College of Engineering");  
                Thread.sleep(10000);  
            }  
        } catch (InterruptedException e) {  
            System.out.println("CollegeThread interrupted: " + e.getMessage());  
        }  
    }  
}  
  
class DepartmentThread extends Thread {  
  
    public void run() {  
        try {  
            while (true) {  
                System.out.println("CSE");  
                Thread.sleep(2000);  
            }  
        } catch (InterruptedException e) {  
            System.out.println("DepartmentThread interrupted: " + e.getMessage());  
        }  
    }  
}  
  
public class MultiThreadDemo {  
    public static void main(String[] args) {  
        CollegeThread collegeThread = new CollegeThread();  
        DepartmentThread departmentThread = new DepartmentThread();  
  
        collegeThread.start();  
        departmentThread.start();  
    }  
}
```

OUTPUT:

```
C:\Users\LAKSHITH.P>cd..  
C:\Users>cd..  
C:\>cd 3rd sem java  
C:\3rd sem java>cd LAB PROGRAM 8  
C:\3rd sem java\LAB PROGRAM 8>javac MultiThreadingDemo.java  
error: file not found: MultiThreadingDemo.java  
Usage: javac <options> <source files>  
use --help for a list of possible options  
C:\3rd sem java\LAB PROGRAM 8>javac MultiThreadDemo.java  
C:\3rd sem java\LAB PROGRAM 8>java MultiThreadDemo  
BMS College of Engineering  
CSE  
CSE  
CSE  
CSE  
CSE  
BMS College of Engineering  
CSE  
CSE  
CSE  
CSE  
CSE  
BMS College of Engineering  
CSE  
CSE  
CSE  
CSE  
CSE  
BMS College of Engineering  
CSE  
CSE  
CSE  
CSE  
CSE  
BMS College of Engineering  
CSE  
CSE  
CSE  
CSE  
CSE  
BMS College of Engineering  
CSE  
CSE  
CSE  
CSE  
CSE  
BMS College of Engineering  
CSE  
CSE  
CSE  
CSE  
CSE  
BMS College of Engineering  
CSE  
CSE  
CSE  
CSE  
CSE  
BMS College of Engineering  
CSE  
CSE  
CSE  
CSE  
CSE  
BMS College of Engineering  
CSE  
CSE  
CSE  
CSE  
CSE
```

Program 9

DIVISION GUI

XAB Program 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 a integer, the program would throw a NumberFormatException. If NUM2 were zero, the program would throw an ArithmeticException. Display the exception in a message dialog box.

```
import java.swing.*;
import java.awt.*;
import java.awt.event.*;
class swingdemo{
    swing demo() {
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 200);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JLabel qlab = new JLabel("Enter the dividend and divisor:");
        JTextField qjtf = new JTextField(2);
        JTextField bjtf = new JTextField(2);
        JButton button = new JButton("calculate");
        JLabel err = new JLabel();
        JLabel alab = new JLabel();
        JLabel blab = new JLabel();
        JLabel anslab = new JLabel();
        jfrm.add(qlab);
        jfrm.add(qjtf);
        jfrm.add(bjtf);
        jfrm.add(button);
        jfrm.add(err);
    }
}
```

a)

```
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        try {
            int a = Integer.parseInt(qjtf.getText());
            int b = Integer.parseInt(bjtf.getText());
            int ans = a/b;
            err.setText("A = " + a);
            blab.setText("B = " + b);
            anslab.setText("Ans = " + 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!");
        }
    }
});

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

enter the dividend and divisor:
  
A=1 B=2 Ans=2
```

CODE:

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

class SwingDemo {
    SwingDemo() {
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 200);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JLabel jlab = new JLabel("Enter the dividend and divisor:");
        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);
        JButton button = new JButton("Calculate");

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

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

        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("A = " + a);
                    blab.setText("B = " + b);
                    anslab.setText("Ans = " + ans);
                    err.setText("");
                } catch (NumberFormatException e) {
                    alab.setText("");
                    blab.setText("");
                    anslab.setText("");
                    err.setText("Enter Only Integers!");
                } catch (ArithmetricException 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();
        }
    });
    System.out.println("LAKSHITH P \n1BM23CS164");
}
}

```

OUT PUT:

```

Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\Users\STUDENT>D:

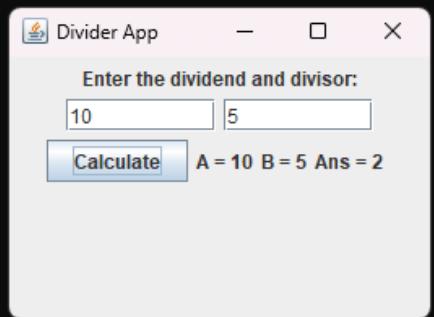
D:\>cd 1bm23cs164/LAB-PROGRAM 9

D:\1BM23CS164\LAB-PROGRAM 9>javac SwingDemo.java

D:\1BM23CS164\LAB-PROGRAM 9>javac SwingDemo.java

D:\1BM23CS164\LAB-PROGRAM 9>java SwingDemo.java
LAKSHITH P
1BM23CS164

```



Program 10

A) PC FIXED

LAB PROGRAM 10

```

a) write java program for inter process communication
continued, to avoid polling Java includes an elegant interprocess
communication mechanism via the wait(), notify()
and notifyAll() methods. These methods are
implemented as final methods in Object, so all classes
have them.

class Q {
    int n;
    boolean valueSet = false;
    synchronized int get() {
        while (!valueSet)
            try {
                System.out.println("In consumer waiting \n");
                wait();
            } catch (InterruptedException e) {
                System.out.println("\n Intimate procedures \n");
                notify();
            }
        return n;
    }

    synchronized void put(int n) {
        while (valueSet)
            try {
                System.out.println("Interrupted exception caught");
            }
        this.n = n;
        valueSet = true;
        System.out.println("put;" + n);
        System.out.println("\n Intimate consumer \n");
        notify();
    }
}

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

```

DATE: PAGE:

DATE: PAGE:

```

this.q = q;
new Thread(this, "Producer").start();
}

public void run() {
    int i = 0;
    while (i < 10) {
        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 < 10) {
            int y = q.get();
            System.out.println("consumed." + y);
            i++;
        }
    }
}

class PC fixed {
    public static void main
        (String args[]) {
        Q q = new Q();
        new producer(q);
        new consumer(q);
        System.out.println("Press control - c to stop");
    }
}

```

OUTPUT:-

press control-c to stop

put: 0

Intimate consumer

producer waiting

Got: 0

Intimate producer

put: 1

Intimate consumer

producer waiting

consumed 0

not: 1

Intimate consumer

consumed: 1

put: 2

Intimate consumer

producer waiting

not: 2

Intimate producer

consumer: 2

put: 3

Intimate consumer

producer waiting

not: 3

Intimate consumer

producer waiting

not: 4

CODE:

```

class Q {
    int n;
    boolean valueSet = false;

    synchronized int get() {
        while (!valueSet) {
            try {
                wait();
            } catch (InterruptedException e) {
                System.out.println("InterruptedException caught");
            }
        }
        System.out.println("Got: " + n);
        valueSet = false;
        notify();
        return n;
    }

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

```

```

        notify();
    }
}

class Producer implements Runnable {
    Q q;

    Producer(Q q) {
        this.q = q;
        new Thread(this, "Producer").start();
    }

    public void run() {
        int i = 0;
        while (i < 15) {
            q.put(i++);
        }
    }
}

class Consumer implements Runnable {
    Q q;

    Consumer(Q q) {
        this.q = q;
        new Thread(this, "Consumer").start();
    }

    public void run() {
        int i = 0;
        while (i < 15) {
            int r = q.get();
            System.out.println("Consumed: " + r);
            i++;
        }
    }
}

public 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.");
        System.out.println("LAKSHITH P \n1BM23CS164");
    }
}

```

OUTPUT:

```
D:\1BM23CS164\LAB-PROGRAM 10 A>javac PCFixed.java
D:\1BM23CS164\LAB-PROGRAM 10 A>java PCFixed
Press Control-C to stop.
LAKSHITH P
1BM23CS164
Put: 0

Intimate Consumer

Producer waiting
Got: 0

Intimate Producer
Put: 1

Intimate Consumer

Producer waiting
Consumed: 0
Got: 1

Intimate Producer
Consumed: 1
Put: 2

Intimate Consumer

Producer waiting
Got: 2

Intimate Producer
Consumed: 2
Put: 3

Intimate Consumer

Producer waiting
Got: 3

Intimate Producer
Consumed: 3
Put: 4

Intimate Consumer

Producer waiting
Got: 4
```

```
Got: 4
Intimate Producer
Consumed: 4
Put: 5

Intimate Consumer

Producer waiting
Got: 5

Intimate Producer
Consumed: 5
Put: 6

Intimate Consumer

Producer waiting
Got: 6

Intimate Producer
Consumed: 6
Put: 7

Intimate Consumer

Producer waiting
Got: 7

Intimate Producer
Consumed: 7
Put: 8

Intimate Consumer

Producer waiting
Got: 8

Intimate Producer
Consumed: 8
Put: 9

Intimate Consumer

Producer waiting
Got: 9

Intimate Producer
```

```
Got: 9
Intimate Producer
Consumed: 9
Put: 10

Intimate Consumer

Producer waiting
Got: 10
Intimate Producer
Consumed: 10
Put: 11

Intimate Consumer

Producer waiting
Got: 11
Intimate Producer
Consumed: 11
Put: 12

Intimate Consumer

Producer waiting
Got: 12
Intimate Producer
Consumed: 12
Put: 13

Intimate Consumer

Producer waiting
Got: 13
Intimate Producer
Consumed: 13
Put: 14

Intimate Consumer

Got: 14
Intimate Producer
Consumed: 14
```

B) Dead lock

not. 14

1) Deadlock a thread enters the monitor on object X and another thread enters the monitor on object Y. If the thread in X tries to call any synchronized method on X, it will be blocked as expected.

```
class A {
    synchronized void foo() {
        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("trying to call B.last()");
        b.last();
    }
}
```

DATE: PAGE:

```
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 A.bar");
        try {
            Thread.sleep(1000);
        } catch (Exception e) {
            System.out.println("B interrupted");
        }
        System.out.println(name + " trying to call A.last()");
        a.last();
    }
}

class DeadLock implements Runnable {
    A a = new A();
    B b = new B();
    DeadLock() {
        Thread.currentThread();
        setName("Main Thread");
        Thread t = new Thread();
        t.setPriority(Thread.MAX_PRIORITY);
        t.start();
        a.foo();
        System.out.println("Back in main thread");
    }
}
```

```

public void run() {
    b.bar();
    System.out.println("Back in other thread");
}

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

OUTPUT:
LAKSHITH P
IBM23CS16A
Main Thread entered A.foo
Racing Thread entered B.bar
Racing Thread trying to call A.last()
Inside A.last
Back in other thread
Main Thread trying to call B.last()
Inside B.last
Back in main thread.

Jyoti

```

CODE:

```

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

        try {
            Thread.sleep(1000);
        } catch(Exception e) {
            System.out.println("A Interrupted");
        }

        System.out.println(name + " trying to call B.last()");
        b.last();
    }

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

class B {

```

```

synchronized void bar(A a) {
    String name = Thread.currentThread().getName();
    System.out.println(name + " entered B.bar");

    try {
        Thread.sleep(1000);
    } catch(Exception e) {
        System.out.println("B Interrupted");
    }

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

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

class Deadlock implements Runnable {
    A a = new A();
    B b = new B();
    Deadlock() {
        System.out.println("LAKSHITH P \n 1BM23CS164");
        Thread.currentThread().setName("MainThread");
        Thread t = new Thread(this, "RacingThread");
        t.start();

        a.foo(b);
        System.out.println("Back in main thread");
    }

    public void run() {
        b.bar(a);
        System.out.println("Back in other thread");
    }
}

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

```

Out put:

```
D:\1BM23CS164\LAB-PROGRAM 10 B>javac Deadlock.java
```

```
D:\1BM23CS164\LAB-PROGRAM 10 B>java Deadlock.java
```

```
LAKSHITH P
```

```
1BM23CS164
```

```
MainThread entered A.foo
```

```
RacingThread entered B.bar
```

```
RacingThread trying to call A.last()
```

```
Inside A.last
```

```
Back in other thread
```

```
MainThread trying to call B.last()
```

```
Inside B.last
```

```
Back in main thread
```



IBM 23CS 164

Name LAKSHITH.P

Std 3rd em Sec 3-4

Roll No. _____ Subject OJ / observation book Std 3rd sem Sec 3-C
School/College Tel. No. _____ School/College B.M.S.C.E

School/College Tel. No. _____ B.M.S.C.E

Sl. No.	Date	Title	Page No.	Teacher Sign / Remarks
1.	23/09/24	LAB-1 program quadratic		S Jay
2.	07/10/24	LAB-2 program SAPA calculator		Prabhakar
3	14/10/24	LAB-3 Program Book (to string)		Chaitanya
4.	21/10/24	LAB-4 Program Area		Varshini
5	28/10/24	LAB-5 Program Bank	10	A. S. S. I.
6.	11/11/24	LAB 6 Final Mark program		
7	28/11/24	LAB 7 Exceptions		
8	28/11/24	LAB 8 Multithreading		
9	28/11/24	LAB 9 GUI interface division		
10	28/11/24	LAB 10 A & B PRized and Deadlock		