

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT

On

OBJECT ORIENTED JAVA PROGRAMMING (23CS3PCOOJ)

Submitted by

BRIJESH S G (1BM22SC073)

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

December 2023 - March 2024

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



This is to certify that the Lab work entitled “**OBJECT ORIENTED JAVA PROGRAMMING**” carried out by **BRIJESH S G (1BM22SC073)**, who is a 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 during the year 2023-24. The Lab report has been approved as it satisfies the academic requirements in respect of Object Oriented Java Programming Lab - **(23CS3PCOOJ)** work prescribed for the said degree.

Prof. Shravya A R
Assistant Professor
Department of CSE
BMSCE, Bengaluru

Dr. Jyothi S Nayak
Professor and Head
Department of CSE
BMSCE, Bengaluru

Index

| Sl. No. | Experiment Title | Page No. |
|---------|------------------|----------|
| 1 | Lab Program 1 | 1 |
| 2 | Lab Program 2 | 3 |
| 3 | Lab Program 3 | 6 |
| 4 | Lab Program 4 | 9 |
| 5 | Lab Program 5 | 11 |
| 6 | Lab Program 6 | 18 |
| 7 | Lab Program 7 | 22 |
| 8 | Lab Program 8 | 24 |

Course outcomes:

| | |
|-----|--|
| CO1 | Apply the knowledge of Java concepts to find the solution for a given problem. |
| CO2 | Analyze the given Java application for correctness/functionalities. |
| CO3 | Develop Java programs / applications for a given requirement. |
| CO4 | Conduct practical experiments for demonstrating features of Java. |

Lab Program 1

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

Code:

```
import java.util.*;
class Quad{
    int a;
    int b;
    int c;
    void input(){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the numbers:");
        System.out.print("Enter a: ");
        a = sc.nextInt();
        System.out.print("Enter b: ");
        b = sc.nextInt();
        System.out.print("Enter c: ");
        c = sc.nextInt();
    }
    void calc(){
        float d,x1,x2;
        d = (b*b-(4*a*c));
        if(d>0){
            x1 = (float)(-b-Math.sqrt(b*b-4*a*c))/(2*a);
            x2 = (float)(-b+Math.sqrt(b*b-4*a*c))/(2*a);
            System.out.println("the roots are real and distinct: "+x1+" and "+x2);
        }
        else if(d<0){
            x1 = -b/(2*a);
            x2 = (float)Math.sqrt(-(b*b-4*a*c))/2*a;
            System.out.println("The root are imaginary and distinct");
            System.out.println("root1: "+x1+"+i "+x2);
            System.out.println("root2: "+x1+"-i "+x2);
        }
        else if(d==0){
            x1=x2=-b/(2*a);
            System.out.println("The roots are real and equal: "+x1+" and "+x2);
        }
    }
}
```

```

}

class QuadraticRun
{
    public static void main(String[] args){
        System.out.println("\n\n");
        System.out.println("Name: Brijesh S G \nUSN: 1BM22CS073\n");
        Quad q = new Quad();
        q.input();
        q.calc();
        System.out.println("\n\n");
    }
}

```

Output:

```

Name: Brijesh S G
USN: 1BM22CS073

Enter the numbers:
Enter a: 1
Enter b: -5
Enter c: 6
the roots are real and distinct: 2.0 and 3.0

```

```

Name: Brijesh S G
USN: 1BM22CS073

Enter the numbers:
Enter a: 1
Enter b: 2
Enter c: 5
The root are imaginary and distinct
root1:-1.0+i 2.0
root2:-1.0-i 2.0

```

```

Name: Brijesh S G
USN: 1BM22CS073

Enter the numbers:
Enter a: 1
Enter b: -4
Enter c: 4
The roots are real and equal: 2.0 and 2.0

```

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.

Code:

```
import java.util.Scanner;

class Student {
    private String usn;
    private String name;
    private int[] credits;
    private int[] marks;

    public Student(String usn, String name, int[] credits, int[] marks) {
        this.usn = usn;
        this.name = name;
        this.credits = credits;
        this.marks = marks;
    }

    public void acceptDetails() {
        Scanner studentInput = new Scanner(System.in);
        System.out.print("Enter USN: ");
        this.usn = studentInput.nextLine();
        System.out.print("Enter Name: ");
        this.name = studentInput.nextLine();
        System.out.print("Enter the number of subjects: ");
        int subjects = studentInput.nextInt();

        this.credits = new int[subjects];
        this.marks = new int[subjects];

        System.out.println("\nEnter details for each subject:");

        for (int i = 0; i < subjects; i++) {
            System.out.print("Enter credits of subject " + (i + 1) + ": ");
            this.credits[i] = studentInput.nextInt();
            System.out.print("Enter marks of subject " + (i + 1) + ": ");
            this.marks[i] = studentInput.nextInt();
            System.out.println();
        }

        studentInput.close();
    }
}
```

```

    }

    public void displayDetails() {
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Credits ");
        for (int i = 0; i < credits.length; i++) {
            System.out.println("Subject " + (i + 1) + ": " + credits[i] + "
credit");
        }
        System.out.println("Marks: ");
        for (int i = 0; i < marks.length; i++) {
            System.out.println("Subject " + (i + 1) + ": " + marks[i] + "
marks");
        }
    }

    public double calculateSGPA() {
        double totalCredits = 0;
        double totalGradePoints = 0;

        for (int i = 0; i < credits.length; i++) {
            totalCredits += credits[i];
            totalGradePoints += (calculateGradePoints(marks[i]) *
credits[i]);
        }
        return totalGradePoints / totalCredits;
    }

    private int calculateGradePoints(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;
        }
    }
}

```

```

class SGPA {
    public static void main(String[] args) {
        System.out.println("\n\n");
        Student student = new Student("USN", "Name", null, null);
        student.acceptDetails();
        System.out.println("Student Details: \n");
        student.displayDetails();
        double sgpa = student.calculateSGPA();
        System.out.println("\nSGPA: " + sgpa);
        System.out.println("\n\n");
    }
}

```

Output:

```

Enter USN: 1BM22CS073
Enter Name: Brijesh S G
Enter the number of subjects: 4

Enter details for each subject:
Enter credits of subject 1: 3
Enter marks of subject 1: 89

Enter credits of subject 2: 4
Enter marks of subject 2: 96

Enter credits of subject 3: 4
Enter marks of subject 3: 88

Enter credits of subject 4: 4
Enter marks of subject 4: 99

```

```

Student Details:

USN: 1BM22CS073
Name: Brijesh S G
Credits
Subject 1: 3 credit
Subject 2: 4 credit
Subject 3: 4 credit
Subject 4: 4 credit
Marks:
Subject 1: 89 marks
Subject 2: 96 marks
Subject 3: 88 marks
Subject 4: 99 marks

SGPA: 9.533333333333333

```


Lab Program 3

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

Code:

```
import java.util.Scanner;

class Book {
    private String name;
    private String author;
    private double price;
    private int numPages;

    public Book(String name, String author, double price, int numPages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getAuthor() {
        return author;
    }

    public void setAuthor(String author) {
        this.author = author;
    }

    public double getPrice() {
        return price;
    }

    public void setPrice(double price) {
        this.price = price;
    }
}
```

```

    }

    public int getNumPages() {
        return numPages;
    }

    public void setNumPages(int numPages) {
        this.numPages = numPages;
    }

    public String toString() {
        return "Book Name: " + name + "\nAuthor: " + author + "\nPrice: ₹" +
price + "\nPages: " + numPages;
    }
}

public class Main {
    public static void main(String[] args) {
        System.out.print("\n\n");
        System.out.println("Name: Brijesh S G \nUSN: 1BM22CS073\n");
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter Number of Books: ");
        int n = scanner.nextInt();
        scanner.nextLine();

        Book[] books = new Book[n];

        for (int i = 0; i < n; i++) {
            System.out.println("Enter details for Book " + (i + 1));
            System.out.print("Name: ");
            String name = scanner.nextLine();
            System.out.print("Author: ");
            String author = scanner.nextLine();
            System.out.print("Price: ");
            double price = scanner.nextDouble();
            System.out.print("Number of pages: ");
            int numPages = scanner.nextInt();
            scanner.nextLine();

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

        for (Book book : books) {
            System.out.println(book.toString());
        }
        scanner.close();
    }
}

```

```
}  
}
```

Output:

```
Name: Brijesh S G  
USN: 1BM22CS073  
  
Enter Number of Books:  
2  
Enter details for Book 1  
Name: Percy Jackson and the Lightning Thief  
Author: Rick Riordan  
Price: 293  
Number of pages: 400  
Enter details for Book 2  
Name: Mistborn: The Final Empire  
Author: Brandon Sanderson  
Price: 490  
Number of pages: 672
```

```
Book Name: Percy Jackson and the Lightning Thief  
Author: Rick Riordan  
Price: ₹293.0  
Pages: 400  
Book Name: Mistborn: The Final Empire  
Author: Brandon Sanderson  
Price: ₹490.0  
Pages: 672
```

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 contains only the method printArea() that prints the area of the given shape.

Code:

```
import java.util.Scanner;

abstract class Shape {
    protected int width, height;

    public abstract void printArea();
}

class Rectangle extends Shape {
    public Rectangle(int width, int height) {
        this.width = width;
        this.height = height;
    }

    public void printArea() {
        System.out.println("Rectangle Area: " + (width * height));
    }
}

class Triangle extends Shape {
    public Triangle(int base, int height) {
        this.width = base;
        this.height = height;
    }

    public void printArea() {
        System.out.println("Triangle Area: " + (0.5 * width * height));
    }
}

class Circle extends Shape {
    public Circle(int radius) {
        this.width = radius;
        this.height = radius;
    }
}
```

```

        public void printArea() {
            System.out.println("Circle Area: " + (Math.PI * width * height));
        }
    }

    public class lab {
        public static void main(String[] args) {
            System.out.print("\n\n");
            System.out.println("Name: Brijesh S G \nUSN: 1BM22CS073\n");

            Scanner scanner = new Scanner(System.in);

            System.out.println("Enter dimensions for the Rectangle:");
            System.out.print("Width: ");
            int rectangleWidth = scanner.nextInt();
            System.out.print("Height: ");
            int rectangleHeight = scanner.nextInt();

            System.out.println("Enter dimensions for the Triangle:");
            System.out.print("Base: ");
            int triangleBase = scanner.nextInt();
            System.out.print("Height: ");
            int triangleHeight = scanner.nextInt();

            System.out.println("Enter dimensions for the Circle:");
            System.out.print("Radius: ");
            int circleRadius = scanner.nextInt();

            Shape rectangle = new Rectangle(rectangleWidth, rectangleHeight);
            Shape triangle = new Triangle(triangleBase, triangleHeight);
            Shape circle = new Circle(circleRadius);

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

            scanner.close();
            System.out.print("\n\n");
        }
    }
}

```

Output:

```
Name: Brijesh S G
USN: 1BM22CS073

Enter dimensions for the Rectangle:
Width: 5
Height: 6
Enter dimensions for the Triangle:
Base: 4
Height: 6
Enter dimensions for the Circle:
Radius: 2
Rectangle Area: 30
Triangle Area: 12.0
Circle Area: 12.566370614359172
```

Lab Program 5

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

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

- a) Accept deposits from customers and update the balance.
- b) Display the balance.
- c) Compute and deposit interest

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

Code:

```
import java.util.Scanner;

abstract class Account {
    protected String customerName;
    protected int accountNumber;
    protected double balance;
    protected String accountType;

    public Account(String customerName, int accountNumber, String
accountType) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
    }

    public abstract void deposit(double amount);

    public abstract int withdraw(double amount);

    public abstract void displayBalance();
}

class SavAcct extends Account {
    private double interestRate;

    public SavAcct(String customerName, int accountNumber, double
interestRate) {
        super(customerName, accountNumber, "Savings");
        this.interestRate = interestRate;
    }

    public void deposit(double amount) {
        balance += amount;
        computeInterest();
        System.out.println("Deposit successful!");
    }
}
```

```

public int withdraw(double amount) {
    if (balance >= amount) {
        balance -= amount;
        System.out.println("Withdrawal successful!");
        return 1;
    } else {
        System.out.println("Insufficient funds!");
        return 0;
    }
}

public void displayBalance() {
    System.out.println("Account Type: " + accountType);
    System.out.println("Customer: " + customerName);
    System.out.println("Account Number: " + accountNumber);
    System.out.println("Balance: " + balance);
}

private void computeInterest() {
    double interest = balance * (interestRate / 100);
    balance += interest;
    System.out.println("Interest deposited: " + interest);
}
}

class CurAcct extends Account {
    private double minimumBalance;
    private double serviceCharge;

    public CurAcct(String customerName, int accountNumber, double
minimumBalance, double serviceCharge) {
        super(customerName, accountNumber, "Current");
        this.minimumBalance = minimumBalance;
        this.serviceCharge = serviceCharge;
    }

    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposit successful!");
    }
}

```



```

    public int withdraw(double amount) {
        if (balance - amount >= minimumBalance) {
            balance -= amount;
            System.out.println("Withdrawal successful!");
            return 1;
        } else {
            System.out.println("Insufficient funds! Service charge of " +
serviceCharge + " imposed.");
            balance -= serviceCharge;
            return 0;
        }
    }

    public void displayBalance() {
        System.out.println("Account Type: " + accountType);
        System.out.println("Customer: " + customerName);
        System.out.println("Account Number: " + accountNumber);
        System.out.println("Balance: " + balance);
    }
}

public class lab {
    public static void main(String[] args) {
        System.out.println("\n\n");
        System.out.println("Name: Brijesh S G \nUSN: 1BM22CS073\n");

        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter customer name: ");
        String customerName = scanner.nextLine();
        System.out.println("Enter account number: ");
        int accountNumber = scanner.nextInt();

        System.out.println("Choose account type:");
        System.out.println("1. Savings");
        System.out.println("2. Current");
        int accountTypeChoice = scanner.nextInt();

        if (accountTypeChoice == 1) {
            System.out.println("Enter interest rate for Savings account:");
            double interestRate = scanner.nextDouble();
            SavAcct savingsAccount = new SavAcct(customerName, accountNumber,
interestRate);

```

```

        operateAccount(savingsAccount);
    } else if (accountTypeChoice == 2) {
        System.out.println("Enter minimum balance for Current account:");
        double minBalance = scanner.nextDouble();
        System.out.println("Enter service charge for Current account:");
        double serviceCharge = scanner.nextDouble();
        CurAcct currentAccount = new CurAcct(customerName, accountNumber,
minBalance, serviceCharge);
        operateAccount(currentAccount);
    } else {
        System.out.println("Invalid account type choice!");
    }

    scanner.close();
    System.out.println("\n\n");
}

private static void operateAccount(Account account) {
    Scanner scanner = new Scanner(System.in);

    int choice;
    do {
        System.out.println("\nMenu:");
        System.out.println("1. Deposit");
        System.out.println("2. Withdraw");
        System.out.println("3. Display Balance");
        System.out.println("4. Exit");
        System.out.print("Enter your choice: ");
        choice = scanner.nextInt();

        switch (choice) {
            case 1:
                System.out.print("Enter amount to deposit: ");
                double depositAmount = scanner.nextDouble();
                account.deposit(depositAmount);
                break;
            case 2:
                System.out.print("Enter amount to withdraw: ");
                double withdrawAmount = scanner.nextDouble();
                account.withdraw(withdrawAmount);
                break;
            case 3:
                account.displayBalance();
                break;
            case 4:

```

```

        System.out.println("Exiting...");
        break;
    default:
        System.out.println("Invalid choice!");
    }
} while (choice != 4);

scanner.close();
}
}

```

Output:

```

Name: Brijesh S G
USN: 1BM22CS073

Enter customer name:
Brijesh
Enter account number:
123
Choose account type:
1. Savings
2. Current
1
Enter interest rate for Savings account:
4

Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
Enter your choice: 1
Enter amount to deposit: 100
Interest deposited: 4.0
Deposit successful!

Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
Enter your choice: 1
Enter amount to deposit: 100
Interest deposited: 8.16
Deposit successful!

```

```

4. Exit
Enter your choice: 3
Account Type: Savings
Customer: Brijesh
Account Number: 123
Balance: 212.16

Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
Enter your choice: 2
Enter amount to withdraw: 10
Withdrawal successful!

Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
Enter your choice: 3
Account Type: Savings
Customer: Brijesh
Account Number: 123
Balance: 202.16

Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
Enter your choice: 4
Exiting...

```

```
Name: Brijesh S G
USN: 1BM22CS073

Enter customer name:
Brijesh S G
Enter account number:
345
Choose account type:
1. Savings
2. Current
2
Enter minimum balance for Current account:
100
Enter service charge for Current account:
30

Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
Enter your choice: 3
Account Type: Current
Customer: Brijesh S G
Account Number: 345
Balance: 0.0

Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
Enter your choice: 1
Enter amount to deposit: 1000
Deposit successful!
```

```
Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
Enter your choice: 2
Enter amount to withdraw: 990
Insufficient funds! Service charge of 30.0 imposed.

Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
Enter your choice: 3
Account Type: Current
Customer: Brijesh S G
Account Number: 345
Balance: 970.0

Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
Enter your choice: 4
Exiting...
```

Lab Program 6

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

Code:

```
package CIE;

public class Internals extends CIE.Student
{
    public int [] InternalMarks;

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

package CIE;
public class Student
{
    String name;
    String usn;
    int sem;

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

```

package SEE;

public class Externals extends CIE.Student
{
    public int[] SeeMarks;

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

//Main File

import CIE.Internals;
import SEE.Externals;
import java.util.Scanner;

public class FinalMarks {
    public static void main(String[] args) {
        System.out.println("\n\n");
        System.out.println("Name: Brijesh S G \nUSN: 1BM22CS073\n");
        Scanner s1 = new Scanner(System.in);

        System.out.println("\nEnter the number of students");
        int n = s1.nextInt();

        String[] name = new String[n];
        String[] usn = new String[n];
        int[] sem = new int[n];
        int[][] InternalMarks = new int[n][5];
        int[][] SeeMarks = new int[n][5];

        for (int i = 0; i < n; i++) {
            System.out.println("Enter details of Student" + (i + 1) + ":");
            System.out.print("\nName:");
            name[i] = s1.next();
            s1.nextLine();
            System.out.print("\nUSN:");
            usn[i] = s1.next();
            s1.nextLine();
            System.out.print("\nSem:");
            sem[i] = s1.nextInt();

            System.out.println("Enter internal marks for 5 courses:");

```

```

        for (int j = 0; j < 5; j++) {
            System.out.println("\nCourse" + (j + 1) + ":");
            InternalMarks[i][j] = s1.nextInt();
        }

        System.out.println("Enter See marks for 5 courses:");
        for (int j = 0; j < 5; j++) {
            System.out.println("\nCourse" + (j + 1) + ":");
            SeeMarks[i][j] = s1.nextInt();
        }
    }

    int[][] FinalMarks = new int[n][5];
    for (int i = 0; i < n; i++) {
        Internals I1 = new Internals(name[i], usn[i], sem[i],
InternalMarks[i]);
        Externals E1 = new Externals(name[i], usn[i], sem[i],
SeeMarks[i]);

        for (int j = 0; j < 5; j++) {
            FinalMarks[i][j] = I1.InternalMarks[i] + E1.SeeMarks[j];
        }
        System.out.println("\n\nFinal Marks for " + n + " Student in 5
courses:");

        for (int k = 0; k < n; k++) {
            System.out.println(name[k] + ":");
            System.out.println("CIE marks:" + ": ");
            for (int j = 0; j < 5; j++) {
                System.out.println("Course" + (j + 1) + ":
"+InternalMarks[k][j]);
            }
            System.out.println("SEE marks:" + ": ");
            for (int j = 0; j < 5; j++) {
                System.out.println("Course" + (j + 1) + ": "+SeeMarks[k][j]);
            }
            System.out.println("\n\n");
        }
        s1.close();
    }
}
}

```

Output:

```
Name: Brijesh S G
USN: 1BM22CS073
```

```
Enter the number of students
1
```

```
Enter details of Student1:
```

```
Name:Student 1
```

```
USN:1BM22CS0X1
```

```
Sem:3
```

```
Enter internal marks for 5 courses:
```

```
Course1:
40
```

```
Course2:
40
```

```
Course3:
40
```

```
Course4:
40
```

```
Course5:
39
```

```
Enter See marks for 5 courses:
```

```
Course1:
99
```

```
Course2:
98
```

```
Course3:
97
```

```
Course4:
99
```

```
Course5:
100
```

```
Final Marks for 1 Student in 5 courses:
```

```
Student:
```

```
CIE marks::
```

```
Course1: 40
```

```
Course2: 40
```

```
Course3: 40
```

```
Course4: 40
```

```
Course5: 39
```

```
SEE marks::
```

```
Course1: 99
```

```
Course2: 98
```

```
Course3: 97
```

```
Course4: 99
```

```
Course5: 100
```


Lab Program 7

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and a 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 causes both father and son’s age and throws an exception if son’s age is >=father’s age.

Code:

```
import java.util.Scanner;

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

class Father {
    private int age;

    public Father(int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException("Age cannot be negative");
        }
        this.age = age;
    }

    public int getAge() {
        return age;
    }
}

class Son extends Father {
    private int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAgeException {
        super(fatherAge);

        if (sonAge >= fatherAge) {
```

```

        throw new WrongAgeException("Son's age should be less than
Father's age");
    }

    this.sonAge = sonAge;
}

public int getSonAge() {
    return sonAge;
}
}

class ExceptionHandlingDemo {
    public static void main(String[] args) {
        System.out.println("\n\n");
        System.out.println("Name: Brijesh S G \nUSN: 1BM22CS073\n");

        Scanner scanner = new Scanner(System.in);

        try {
            // Taking user input for Father's age
            System.out.print("Enter Father's Age: ");
            int fatherAge = scanner.nextInt();
            Father father = new Father(fatherAge);
            System.out.println("Father's Age: " + father.getAge());

            // Taking user input for Son's age
            System.out.print("Enter Son's Age: ");
            int sonAge = scanner.nextInt();
            Son son = new Son(father.getAge(), sonAge);
            System.out.println("Son's Age: " + son.getSonAge());

        } catch (WrongAgeException e) {
            System.out.println("Exception caught: " + e.getMessage());
        } finally {
            scanner.close();
        }
        System.out.println("\n\n");
    }
}

```

Output:

```
Name: Brijesh S G
USN: 1BM22CS073
```

```
Enter Father's Age: 50
Father's Age: 50
Enter Son's Age: 14
Son's Age: 14
```

```
Name: Brijesh S G
USN: 1BM22CS073
```

```
Enter Father's Age: -50
Exception caught: Age cannot be negative
```

```
Name: Brijesh S G
USN: 1BM22CS073
```

```
Enter Father's Age: 12
Father's Age: 12
Enter Son's Age: 50
Exception caught: Son's age should be less than Father's age
```

Lab Program 8

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

Code:

```
class DisplayThread extends Thread {
    private String message;
    private int interval;

    public DisplayThread(String message, int interval) {
        this.message = message;
        this.interval = interval;
    }

    public void run() {
        try {
            while (true) {
```

```

        System.out.println(message);
        Thread.sleep(interval);
    }
} catch (InterruptedException e) {
    e.printStackTrace();
}
}
}

public class DisplayMessages {
    public static void main(String[] args) {
        // Create a thread for "BMS College of Engineering" every ten seconds
        DisplayThread thread1 = new DisplayThread("BMS College of
Engineering", 10000);

        // Create a thread for "CSE" every two seconds
        DisplayThread thread2 = new DisplayThread("CSE", 2000);

        // Start both threads
        thread1.start();
        thread2.start();
    }
}

```

Output:

```

Name: Brijesh S G
USN: 1BM22CS073

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

```