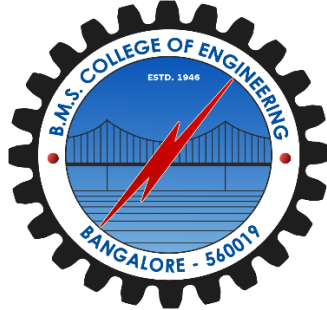


B.M.S. College of Engineering

(Autonomous College Affiliated to Visvesvaraya Technological University, Belgaum)

Bull Temple Road, Basavanagudi, Bengaluru – 560019



**Department of
Computer Science & Engineering (CSE)**

Lab Programs Report

Course Title:

Object Oriented Java Programming

Course Code: 23CS3PCOOJ

By

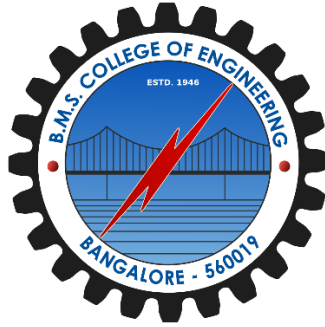
AVANI.A (1BM22CS059)

B.M.S. College of Engineering

(Autonomous College Affiliated to

Visvesvaraya Technological University, Belgaum)

Bull Temple Road, Basavanagudi, Bengaluru – 560019



Department of Computer Science & Engineering (CSE)

LABORATORY CERTIFICATE

This is to certify that “**AVANI.A**” has satisfactorily completed the course of Experiments in Practical **OBJECT ORIENTED JAVA PROGRAMMING** prescribed by the Department during the odd semester 2023-24.

Name of the Candidate: AVANI.A

USN:**1BM22CS059** SEM:3 SEC: B

AVANI.A

1BM22CS059

SHRAVYA.AR

Assistant professor

Dept of cse

BMSCE

TABLE OF CONTENTS:

S.NO	TITLE	PAGE NUMBER
1.	LAB PROGRAM -1	1-3
2.	LAB PROGRAM -2	4-6
3.	LAB PROGRAM -3	7-9
4.	LAB PROGRAM -4	10-12
5.	LAB PROGRAM -5	13-21
6.	LAB PROGRAM -6	22-26
7.	LAB PROGRAM -7	27-29
8.	LAB PROGRAM -8	30-31

LAB PROGRAM- 1

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 percentage of a student.

CODE:

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

```
class Student {
```

```
    String USN;
```

```
    String name;
```

```
    int marks[] = new int[6];
```

```
    float totalMarks = 0;
```

```
    void getData(int i) {
```

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

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

```
        USN = s.next();
```

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

```
        name = s.next();
```

```
        System.out.println("Enter Student " + (i + 1) + " Marks");
```

```
        for (int j = 0; j < 6; j++) {
```

```
            System.out.println("Enter Marks of Subject " + (j + 1) + ":");
```

```
            marks[j] = s.nextInt();
```

```
            totalMarks += marks[j];
```

```
        }
```

```
    }
```

```
    void calculatePercentage(int i) {
```

```
        float percentage = (totalMarks / 6);
```

```
        System.out.println("Percentage of student " + (i + 1) + " = " + percentage + "%");
```

```

    }
}

class StudentMain {
    public static void main(String[] args) {
        System.out.println("AVANI ");
        System.out.println("1BM22CS059");
        System.out.println("Enter the number of Students");
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        Student[] students = new Student[n];

        for (int i = 0; i < n; i++) {
            students[i] = new Student();
            students[i].getData(i);
            students[i].calculatePercentage(i);
        }
    }
}

```

OUTPUT:

```
C:\Users\avani\OneDrive\Desktop\JAVA PGMS (LAB)>javac StudentMain.java

C:\Users\avani\OneDrive\Desktop\JAVA PGMS (LAB)>java StudentMain
AVANI
1BM22CS059
Enter the number of Students
1
Enter USN:
1bm22cs059
Enter Name:
avani.a
Enter Student1 Marks
Enter Marks of Subject 1:
56
Enter Marks of Subject 2:
66
Enter Marks of Subject 3:
57
Enter Marks of Subject 4:
84
Enter Marks of Subject 5:
91
Enter Marks of Subject 6:
81
Percentage of student 1 = 72.5%

C:\Users\avani\OneDrive\Desktop\JAVA PGMS (LAB)>
```

LAB PROGRAM-2

Create a class Book that contains four members: name, author, price, and 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 Books {
    String Name;
    String Author;
    double price;
    int num_pages;

    Books(String Name, String Author, double price, int num_pages) {
        this.Name = Name;
        this.Author = Author;
        this.num_pages = num_pages;
        this.price = price;
    }

    void setDetails() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter Name:");
        Name = s.nextLine();
        System.out.println("Enter Author Name:");
        Author = s.nextLine();
        System.out.println("Enter price:");
        price = s.nextDouble();
        System.out.println("Enter Number of pages:");
        num_pages = s.nextInt();
        System.out.println("-----");
    }
}
```

```

    }

    void getDetails() {
        System.out.println("Book Name: " + Name);
        System.out.println("Author Name: " + Author);
        System.out.println("Price: " + price);
        System.out.println("Number of pages: " + num_pages);
    }

    public String toString() {
        return "Book Name: " + Name + "\nAuthor Name: " + Author + "\nPrice: " + price + "\nNumber
of pages: " + num_pages;
    }
}

class BookMain {
    public static void main(String args[]) {
        Scanner s = new Scanner(System.in);
        int n, i;
        System.out.println("Enter number of books:");
        n = s.nextInt();
        Books[] books = new Books[n];

        for (i = 0; i < n; i++) {
            System.out.println("Enter details of book " + (i + 1));
            books[i] = new Books("", "", 0.0, 0);
            books[i].setDetails();
        }

        for (i = 0; i < n; i++) {
            System.out.println("Details of book " + (i + 1));
            books[i].getDetails();
        }
    }
}

```


OUTPUT:

```
C:\Users\avani\OneDrive\Desktop\avani javalab>javac BookMain.java

C:\Users\avani\OneDrive\Desktop\avani javalab>java BookMain
Enter number of books:
2
Enter details of book 1
Enter Name:
twisted love
Enter Author Name:
ana huang
Enter price:
400
Enter Number of pages:
259
-----
Enter details of book 2
Enter Name:
colleen hoover
Enter Author Name:
it ends w us
Enter price:
479
Enter Number of pages:
300
-----
Details of book 1
Book Name: twisted love
Author Name: ana huang
Price: 400.0
Number of pages: 259
Details of book 2
Book Name: colleen hoover
Author Name: it ends w us
Price: 479.0
Number of pages: 300

C:\Users\avani\OneDrive\Desktop\avani javalab>
```

LAB PROGRAM-3

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:

```
/*java program o write quadartic equation */
import java.util.Scanner;

class Quad {
    int a, b, c;
    double root1, root2, d;
    Scanner s = new Scanner(System.in);

    void input()
    {

        System.out.println("Quadratic equation is in the form :  $ax^2 + bx + c$ ");

        System.out.print("Enter a:");
        a = s.nextInt();
        System.out.print("Enter b:");
        b = s.nextInt();
        System.out.print("Enter c:");
        c = s.nextInt();
    }

    void discriminant() {

        d= (b*b)-(4*a*c);
    }
}
```

```

void calculateRoots() {
    if(d>0)
    {
        System.out.println("Roots are real and unequal");
        root1 = ( - b + Math.sqrt(d))/(2*a);
        root2 = (-b - Math.sqrt(d))/(2*a);
        System.out.println("First root is:"+root1);
        System.out.println("Second root is:"+root2);
    }

    else if(d == 0)

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

        root1 = (-b+Math.sqrt(d))/(2*a);
        System.out.println("Root:"+root1);
    }

    else

    {
        System.out.println("No real solutions. Roots are imaginary");
        double real = -b / (2 * a);
        double imaginary = Math.sqrt(-d) / (2 * a);
        System.out.println("The equation has two complex roots: " + real + " + " + imaginary + "i
        and "
        + real + " - " + imaginary + "i");
    }

}

```

```

}

class Main {

    public static void main(String[] args) {
        Quad q= new Quad();
        q.input();
        q.discriminant();
        q.calculateRoots();

    }

}

```

OUTPUT:

```

C:\Users\avani\OneDrive\Desktop\avani javalab>javac QMain.java

C:\Users\avani\OneDrive\Desktop\avani javalab>java QMain
Quadratic equation is in the form : ax^2 + bx + c
Enter a:1
Enter b:1
Enter c:1
No real solutions. Roots are imaginary
The equation has two complex roots: 0.0 + 0.8660254037844386i and 0.0 - 0.8660254037844386i

C:\Users\avani\OneDrive\Desktop\avani javalab>java QMain
Quadratic equation is in the form : ax^2 + bx + c
Enter a:1
Enter b:-2
Enter c:1
Roots are real and equal
Root:1.0

C:\Users\avani\OneDrive\Desktop\avani javalab>java QMain
Quadratic equation is in the form : ax^2 + bx + c
Enter a:1
Enter b:5
Enter c:2
Roots are real and unequal
First root is:-0.4384471871911697
Second root is:-4.561552812808831

C:\Users\avani\OneDrive\Desktop\avani javalab>java QMain
Quadratic equation is in the form : ax^2 + bx + c
Enter a:-1
Enter b:-2
Enter c:-5
No real solutions. Roots are imaginary
The equation has two complex roots: -1.0 + -2.0i and -1.0 - -2.0i

```

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

CODE:

```
import java.util.Scanner;

abstract class Shape {
    int dim1;
    int dim2;

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

    abstract void printArea();
}

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

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

```

class Triangle extends Shape {
    Triangle(int height, int base) {
        super(height, base);
    }

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

```

```

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

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

```

```

public class AbstractMain {
    public static void main(String[] args) {
        System.out.println("avani a");
        System.out.println("1BM22CS059");
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the length and breadth of the rectangle");
        int l = s.nextInt();
        int b = s.nextInt();
    }
}

```

```

        System.out.println("Enter base and height of the triangle");
        int ba = s.nextInt();
        int h = s.nextInt();
        System.out.println("Enter the radius of the circle");
        int r = s.nextInt();
        Rectangle re = new Rectangle(l, b);
        Triangle t = new Triangle(h, ba);
        Circle c = new Circle(r);
        re.printArea();
        t.printArea();
        c.printArea();
    }
}

```

OUTPUT:

```

C:\Users\avani\OneDrive\Desktop\avani javalab>javac AbstractMain.java
C:\Users\avani\OneDrive\Desktop\avani javalab>java AbstractMain
avani a
1BM22CS059
Enter the length and breadth of the rectangle
2 4
Enter base and height of the triangle
4 5
Enter the radius of the circle
67
Area of rectangle = 8.0
Area of triangle = 10.0
Area of circle = 14102.60942196458
C:\Users\avani\OneDrive\Desktop\avani javalab>

```

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:

- 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.

CODE:

```
import java.util.Scanner;

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

    Account(String name, int number, String type, double
initialBalance) {
        customerName = name;
        accountNumber = number;
        accountType = type;
        balance = initialBalance;
    }

    void deposit(double amount) {
        if (amount > 0) {
            balance += amount;

            System.out.println("Deposit of INR " + amount + "
successful");
        } else {
            System.out.println("Invalid deposit amount. Please enter a
positive value.");
        }
    }
}
```



```

    }
}

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

void withdraw(double amount) {
    if (balance >= amount) {
        balance -= amount;
        System.out.println("Withdrawal of INR " + amount + "
successful");
    } else {
        System.out.println("Insufficient funds");
    }
}

void computeInterest() {
}

void checkMinimumBalance(double minBalance, double
serviceCharge) {
}
}

class SavAcct extends Account {
    double interestRate = 0.05;

    SavAcct(String name, int number, String type, double
initialBalance) {
        super(name, number, type, initialBalance);
    }

    void computeInterest() {

```

```

        double interest = balance * interestRate;

        balance += interest;

        System.out.println("Interest of INR " + interest + " added to
the account");
    }
}

class CurAcct extends Account {
    double minBalance = 1000;
    double serviceCharge = 50;

    CurAcct(String name, int number, String type, double
initialBalance) {
        super(name, number, type, initialBalance);
    }

    void checkMinimumBalance(double minBalance, double
serviceCharge) {
        if (balance < minBalance) {
            System.out.println("Service charge of INR " +
serviceCharge + " imposed");
            balance -= serviceCharge;
        }
    }
}

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

        System.out.print("Enter the number of users: ");

        int numUsers = scanner.nextInt();

        Account[] accounts = new Account[numUsers];

        for (int i = 0; i < numUsers; i++) {
            System.out.println("\nUser " + (i + 1));

```

```

        System.out.print("Enter customer name: ");
        scanner.nextLine();
        String name = scanner.nextLine();
        System.out.print("Enter account number: ");
        int
accNumber = scanner.nextInt();
        System.out.print("Enter initial deposit amount: INR ");
        double initialDeposit = scanner.nextDouble();
        System.out.print("Enter account type (Savings/Current):
");
        scanner.nextLine();
        String accType = scanner.nextLine();
        if (accType.equalsIgnoreCase("Savings")) {
            accounts[i] = new SavAcct(name, accNumber, accType,
initialDeposit);
        } else if (accType.equalsIgnoreCase("Current")) {
            accounts[i] = new CurAcct(name, accNumber, accType,
initialDeposit);
        } else {
            System.out.println("Invalid account type entered.
Defaulting to Account.");
            accounts[i] = new Account(name, accNumber,
"Account", initialDeposit);
        }
    }
}
boolean exit = false;
while (!exit) {
    System.out.println("\nChoose an option:");
    System.out.println("1. Deposit");
    System.out.println("2. Withdraw");

```

```

        System.out.println("3. Display Balance");
        System.out.println("4. Compute Interest (Savings only)");
        System.out.println("5. Exit");
        System.out.print("Enter your choice: ");
        while (!scanner.hasNextInt()) {
            System.out.println("Invalid input. Please enter a
number.");
            scanner.next();
        }
        int
choice = scanner.nextInt();
        switch (choice) {
            case 1:
                System.out.print("Enter account number: ");
                int
accNum = scanner.nextInt();
                System.out.print("Enter deposit amount: INR ");
                double depositAmount = scanner.nextDouble();
                for (Account acc : accounts) {
                    if (acc.accountNumber == accNum) {
                        acc.deposit(depositAmount);
                    }
                }
                break;
            case 2:
                System.out.print("Enter account number: ");
                accNum = scanner.nextInt();
                System.out.print("Enter withdrawal amount: INR ");
                double withdrawAmount = scanner.nextDouble();
                for (Account acc : accounts) {

```

```

        if (acc.accountNumber == accNum) {
            acc.withdraw(withdrawAmount);
        }
    }
    break;
case 3:
    System.out.print("Enter account number: ");
    accNum = scanner.nextInt();
    for (Account acc : accounts) {
        if (acc.accountNumber == accNum) {
            acc.displayBalance();
        }
    }
    break;
case 4:
    System.out.print("Enter account number (for Savings
account): ");
    accNum = scanner.nextInt();
    for (Account acc : accounts) {
        if (acc.accountNumber == accNum && acc
instanceof SavAcct) {
            ((SavAcct) acc).computeInterest();
        }
    }
    break;
case 5:
    exit = true;
    break;
default:
    System.out.println("Invalid choice. Please enter a

```

```
valid option.");
```

```
}
```

```
}
```

```
}
```

```
}
```

OUTPUT:

```
C:\Users\avani\OneDrive\Desktop\avani javalab>javac Bank.java
```

```
C:\Users\avani\OneDrive\Desktop\avani javalab>java Bank
```

```
Enter the number of users: 1
```

```
User 1
```

```
Enter customer name: avani
```

```
Enter account number: 12345
```

```
Enter initial deposit amount: INR 200
```

```
Enter account type (Savings/Current): savings
```

```
Choose an option:
```

```
1. Deposit
```

```
2. Withdraw
```

```
3. Display Balance
```

```
4. Compute Interest (Savings only)
```

```
5. Exit
```

```
Enter your choice: 1
```

```
Enter account number: 12345
```

```
Enter deposit amount: INR 100
```

```
Deposit of INR 100.0 successful
```

```
Choose an option:
```

```
1. Deposit
```

```
2. Withdraw
```

```
3. Display Balance
```

```
4. Compute Interest (Savings only)
```

```
5. Exit
```

```
Enter your choice: 3
```

```
Enter account number: 12345
```

```
Account Number: 12345
```

```
Customer Name: avani
```

```
Account Type: savings
```

```
Balance: INR 300.0
```

```
Choose an option:
```

```
1. Deposit
```

```
2. Withdraw
```

```
3. Display Balance
```

```
4. Compute Interest (Savings only)
```

```
5. Exit
```

```
Enter your choice: 2
```

```
Enter account number: 12345
```

```

Enter withdrawal amount: INR 50
Withdrawal of INR 50.0 successful

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 3
Enter account number: 12345
Account Number: 12345
Customer Name: avani
Account Type: savings
Balance: INR 250.0

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 4
Enter account number (for Savings account): 12345
Interest of INR 12.5 added to the account

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 5

C:\Users\avani\OneDrive\Desktop\avani javalab>

C:\Users\avani\OneDrive\Desktop\avani javalab>javac Bank.java

C:\Users\avani\OneDrive\Desktop\avani javalab>java Bank
Enter the number of users: 1

User 1
Enter customer name: avani
Enter account number: 555
Enter initial deposit amount: INR 200
Enter account type (Savings/Current): current

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 1
Enter account number: 555
Enter deposit amount: INR 200
Deposit of INR 200.0 successful

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 3
Enter account number: 555
Account Number: 555
Customer Name: avani
Account Type: current
Balance: INR 400.0

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 2
Enter account number: 555
Enter withdrawal amount: INR 350

```

```
Withdrawal of INR 350.0 successful

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 3
Enter account number: 555
Account Number: 555
Customer Name: avani
Account Type: current
Balance: INR 50.0

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 5

C:\Users\avani\OneDrive\Desktop\avani javalab>
```


LAB PROGRAM-6

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

- Create a folder CIE and save the programs Student.java and Internals.java within it.
- Create a folder SEE and save the program External.java within it.
- Save the Main program outside these two folders.
- Compile Main.java and Execute the Main.class

CODE:

```
package CIE;
```

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

```

        this.sem = sem;
    }
}

package CIE;

public class internals extends student {
    public int[] internalMarks;

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

package SEE;

import CIE.student;

public class externals extends student {
    public int[] seeMarks;

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

import CIE.internals;
import SEE.externals;
import java.util.Scanner;

public class main1 {

```

```

public static void main(String[] args) {
    System.out.println("Bhuvana M");
    System.out.println("1BM22CS071");
    Scanner scanner = new Scanner(System.in);
    System.out.print("Enter the number of students: ");
    int n = scanner.nextInt();
    internals[] cieStudents = new internals[n];
    externals[] seeStudents = new externals[n];

    // Input CIE marks
    for (int i = 0; i < n; i++) {
        System.out.println("Enter details for CIE of 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[] cieMarks = new int[5];
        System.out.print("Enter CIE marks for 5 courses: ");
        for (int j = 0; j < 5; j++) {
            cieMarks[j] = scanner.nextInt();
        }
        cieStudents[i] = new internals(usn, name, sem, cieMarks);
    }

    // Input SEE marks
    for (int i = 0; i < n; i++) {
        System.out.println("Enter details for SEE of 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[] seeMarks = new int[5];
System.out.print("Enter SEE marks for 5 courses: ");
for (int j = 0; j < 5; j++) {
    seeMarks[j] = scanner.nextInt();
}
seeStudents[i] = new externals(usn, name, sem, seeMarks);
}

// Displaying final marks
System.out.println("\nFinal Marks of Students:");
for (int i = 0; i < n; i++) {
    System.out.println("\nDetails of Student " + (i + 1));
    System.out.println("USN: " + cieStudents[i].usn);
    System.out.println("Name: " + cieStudents[i].name);
    System.out.println("Semester: " + cieStudents[i].sem);
    System.out.println("CIE Marks: ");
    for (int j = 0; j < 5; j++) {
        System.out.print(cieStudents[i].internalMarks[j] + " ");
    }
    System.out.println("\nSEE Marks: ");
    for (int j = 0; j < 5; j++) {
        System.out.print(seeStudents[i].seeMarks[j] + " ");
    }
}
}

```

}

OUTPUT:

```
C:\Users\avani\OneDrive\Desktop\avani_java>cd C:\Users\avani\OneDrive\Desktop\avani_java
C:\Users\avani\OneDrive\Desktop\avani_java>javac CIE\student.java CIE\internals.java SEE\externals.java main1.java
C:\Users\avani\OneDrive\Desktop\avani_java>java main1
AVANI.A
1BM22CS059
Enter the number of students: 1
Enter details for CIE of student 1
USN: 1bm22cs059
Name: avani.a
Semester: 3
Enter CIE marks for 5 courses: 20 34 22 20 24
Enter details for SEE of student 1
USN: 1bm22cs059
Name: avani.a
Semester: 3
Enter SEE marks for 5 courses: 98 80 76 64 88

Final Marks of Students:

Details of Student 1
USN: 1bm22cs059
Name: avani.a
Semester: 3
CIE Marks:
20 34 22 20 24
SEE Marks:
98 80 76 64 88
C:\Users\avani\OneDrive\Desktop\avani_java>
```

LAB PROGRAM-7

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

CODE :

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

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

```
class Father {  
    int fatherAge;  
  
    Father() throws WrongAge {  
        Scanner s = new Scanner(System.in);  
        System.out.println("Enter Father's Age:");  
        fatherAge = s.nextInt();  
        if (fatherAge < 0) {  
            throw new WrongAge("Age cannot be negative"); // Fixed: 'Throw' to 'throw'  
        }  
    }  
}
```

```

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

class Son extends Father {
    int sonAge;

    Son() throws WrongAge {
        super();
        Scanner s = new Scanner(System.in);
        System.out.println("Enter Son's Age:");
        sonAge = s.nextInt();
        if (sonAge > fatherAge) {
            throw new WrongAge("Son's age cannot be greater than Father's age");
        } else if (sonAge == fatherAge) {
            throw new WrongAge("Son's age cannot be equal to Father's Age"); // Fixed: 'Throw'
to 'throw'
        } else if (sonAge < 0) {
            throw new WrongAge("Age cannot be negative"); // Fixed: 'Throw' to 'throw'
        }
    }

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

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

```

```

    try {
        Son s = new Son();
        s.display();
    } catch (WrongAge e) {
        System.out.println(e.getMessage());
    }
}
}

```

OUTPUT:

```

C:\Users\avani\OneDrive\Desktop\avani javalab>javac Main.java

C:\Users\avani\OneDrive\Desktop\avani javalab>java Main
Enter Father?s Age:
52
Enter Son?s Age:
25
Father Age is: 52
Son?s age is: 25

C:\Users\avani\OneDrive\Desktop\avani javalab>java Main
Enter Father?s Age:
20
Enter Son?s Age:
20
Son?s age cannot be equal to Father?s Age

C:\Users\avani\OneDrive\Desktop\avani javalab>java Main
Enter Father?s Age:
-20
Age cannot be negative

C:\Users\avani\OneDrive\Desktop\avani javalab>

```


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 BMS implements Runnable {  
    public void run() {  
        while (true) {  
            try {  
                System.out.println("BMS college of engineering"); // Fixed: System with uppercase  
'S'  
                Thread.sleep(10000);  
            } catch (InterruptedException e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}
```

```
class CSE implements Runnable {  
    public void run() {  
        while (true) {  
            try {  
                System.out.println("CSE"); // Fixed: System with uppercase 'S'  
                Thread.sleep(2000);  
            } catch (InterruptedException e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}
```

```
}
```

```
public class Main1 {  
    public static void main(String[] args) {  
        Thread t1 = new Thread(new BMS());  
        Thread t2 = new Thread(new CSE());  
        t1.start(); // Fixed: 'start' with lowercase 's'  
        t2.start(); // Fixed: 'start' with lowercase 's'  
    }  
}
```

Output:

```
C:\Users\avani\OneDrive\Desktop\avani javalab>javac Main1.java  
C:\Users\avani\OneDrive\Desktop\avani javalab>java Main1  
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
```

THANK YOU.