

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT
on

Object Oriented Java Programming (23CS3PCOOJ)

Submitted by

Simrik Sharma (1BF24CS293)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in

B.M.S. COLLEGE OF ENGINEERING

(Autonomous Institution under VTU)

BENGALURU-560019

Aug-2025 to Jan-2026

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 **SIMRIK SHARMA (1BF24CS293)**, who is bonafide student of **B.M.S. College of Engineering**. It is in partial fulfilment 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.

Dr. Seema Patil Associate Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE
--	--

Index

Sl. No.	Date	Experiment Title	Page No.
1	23/09/25	Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c = 0$. Read in a, b, c and use the quadratic formula. If the discriminate $b^2 - 4ac$ is negative, display a message stating that there are no real solutions.	5
2	13/10/25	Develop a Java program to create a class Student with members USN, name, an array credits and an array mark. Include methods to accept and display details and a method to calculate SGPA of a student.	7
3	14/10/25	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.	10
4	04/11/25	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.	12
5	04/11/25	Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks: a) Accept deposit from customer and update the balance. b) Display the balance. c) Compute and deposit interest	14

		d) Permit withdrawal and update the balance Check for the minimum balance, impose penalty if necessary and update the balance.	
6	18/11/25	Create a package CIE which has two classes - Personal 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 Personal. 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.	19
7	25/11/25	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=father's age.	21
8	09/12/25	Write a program which creates two threads, one thread displaying "BMS College of Engineering" once every ten seconds	23

Github Link:

<https://github.com/Simrik-Sharma/Java-1BF24CS293>

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.Scanner;
import java.lang.Math;
class Quadratic
{
    public static void main (String []args)
    {
        double a, b, c, d, r1, r2;
        Scanner in = new Scanner (System.in);
        System.out.println ("Enter the first number: ");
        a = in.nextDouble();
        System.out.println ("Enter the second number: ");
        b = in.nextDouble();
        System.out.println ("Enter the third number: ");
        c = in.nextDouble();
        if (a==0)
        {
            System.out.println ("the quadratic equation does not exist");
            System.out.println ("Enter the a non zero number: ");
            a = in.nextDouble();
        }
        d = b*b - 4*a*c;
        if (d==0)
        {
            r1 = (-b)/(2*a);
            System.out.println ("The roots are real and equal ");
            System.out.println (r1);
        }
        else if (d>0)
        {
            r1 = ((-b) + (Math.sqrt(d)))/(double)(2*a);
            r2 = ((-b) - (Math.sqrt(d)))/(double)(2*a);
            System.out.println ("The roots are real and unequal ");
            System.out.println (r1);
            System.out.println (r2);
        }
        else if (d<0)
        {
            r1 = (-b)/(2*a);
            r2 = Math.sqrt(-d)/(2*a);
            System.out.println ("The roots are imaginary");
            System.out.println (r1);
            System.out.println (r2);
        }
    }
}
```

```

}
PS C:\Users\Admin\Desktop\Watali (1WN24CS169)> cd "c:\Users\Admin\Desktop\1BF24CS293\" ; if ($?) { javac Quadratic.java } ; if ($?) { java Quadratic }
Enter the first number:
3
Enter the second number:
7
Enter the third number:
3
The roots are real and unequal
-0.5657414540893352
-1.7675918792439982
PS C:\Users\Admin\Desktop\1BF24CS293> cd "c:\Users\Admin\Desktop\1BF24CS293\" ; if ($?) { javac Quadratic.java } ; if ($?) { java Quadratic }
Enter the first number:
0
Enter the second number:
5
Enter the third number:
2
the quadratic equation does not exist
Enter the a non zero number:
4
The roots are imaginary
-0.625
0.33071891388307384
PS C:\Users\Admin\Desktop\1BF24CS293>

```

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
{
    String USN;
    String name;
    int[] marks = new int[8];
    int[] credits = new int[8];
    void read(Scanner sc)
    {
        System.out.println("Enter USN:");
        USN = sc.nextLine();
        System.out.println("Enter Name:");
        name = sc.nextLine();
        System.out.println("Enter marks and credits for 8 subjects:");
        for (int i = 0; i < 8; i++)
        {
            System.out.println("Subject " + (i + 1) + " marks:");
            marks[i] = sc.nextInt();
            System.out.println("Subject " + (i + 1) + " credits:");
            credits[i] = sc.nextInt();
            sc.nextLine();
        }
    }
    double gradePoint(int mark)
    {
        if (mark >= 90) return 10;
        else if (mark >= 80) return 9;
        else if (mark >= 70) return 8;
        else if (mark >= 60) return 7;
        else if (mark >= 50) return 6;
        else if (mark >= 40) return 5;
        else return 0;
    }
    double sgpa()
    {
```

```

        int totalCredits = 0;
        double totalPoints = 0;
        for (int i = 0; i < 8; i++)
        {
            totalPoints += gradePoint(marks[i]) * credits[i];
            totalCredits += credits[i];
        }
        return totalPoints / totalCredits;
    }
    void display()
    {
        System.out.println("USN: " + USN);
        System.out.println("Name: " + name);
        System.out.printf("SGPA: %.2f\n", sgpa());
    }
}
class gpaa
{
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        Student s1 = new Student();
        System.out.println("Enter details for Student 1:");
        s1.read(sc);
        System.out.println("Details for Student 1:");
        s1.display();
        System.out.println();
        Student s2 = new Student();
        System.out.println("Enter details for Student 2:");
        s2.read(sc);
        System.out.println("Details for Student 2:");
        s2.display();
        sc.close();
    }
}

```



```
cd "c:\1BF24CS293\" ; if ($?) { javac gpaa.java } ; if ($?) {
Enter details for Student 1:
Enter USN:
1BF24CS293
Enter Name:
SIMRIK SHARMA
Enter marks and credits for 8 subjects:
Subject 1 marks:
88
Subject 1 credits:
4
Subject 2 marks:
90
Subject 2 credits:
4
Subject 3 marks:
89
Subject 3 credits:
4
Subject 4 marks:
88
Subject 4 credits:
3
Subject 5 marks:
79
Subject 5 credits:
3
Subject 6 marks:
82
Subject 6 credits:
3
Subject 7 marks:
86
Subject 7 credits:
1
Subject 8 marks:
90
Subject 8 credits:
1
Details for Student 1:
USN: 1BF24CS293
Name: SIMRIK SHARMA
SGPA: 9.09
```

```

Enter details for Student 2:
Enter USN:
1BF24CS284
Enter Name:
SHRAVANI APARAJ
Enter marks and credits for 8 subjects:
Subject 1 marks:
89
Subject 1 credits:
4
Subject 2 marks:
86
Subject 2 credits:
4
Subject 3 marks:
95
Subject 3 credits:
4
Subject 4 marks:
90
Subject 4 credits:
3
Subject 5 marks:
83
Subject 5 credits:
3
Subject 6 marks:
71
Subject 6 credits:
3
Subject 7 marks:
82
Subject 7 credits:
1
Subject 8 marks:
88
Subject 8 credits:
1
Details for Student 2:
USN: 1BF24CS284
Name: SHRAVANI APARAJ
SGPA: 9.17
PS C:\1BF24CS293> 
```

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.

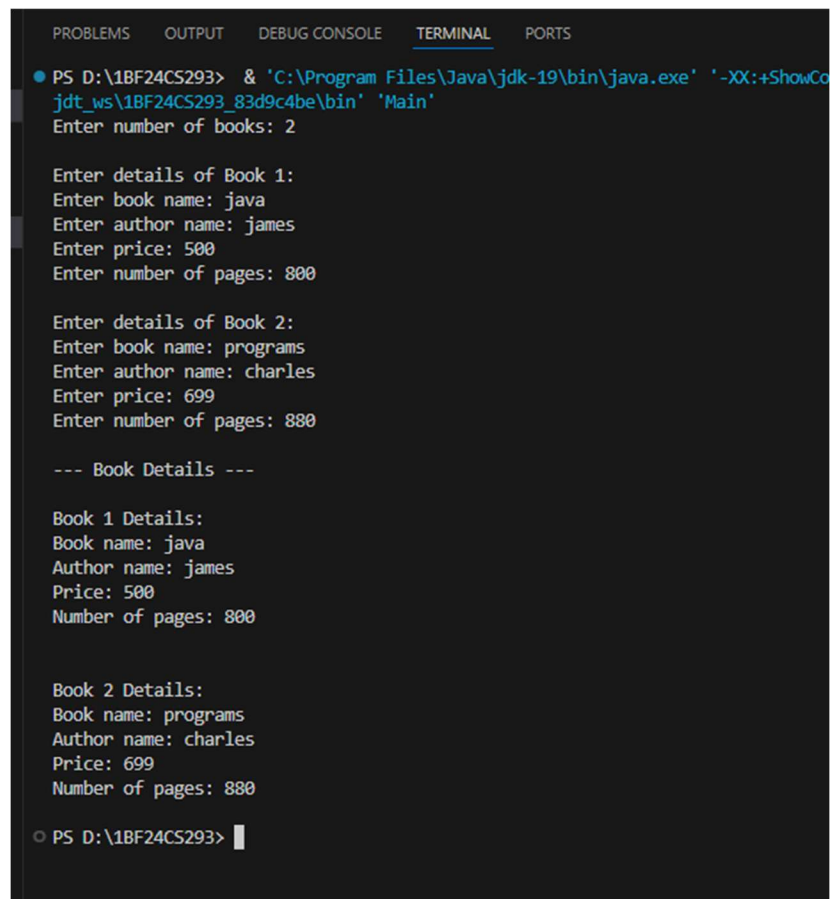
Code:

```
import java.util.Scanner;
class Books {
    String name;
    String author;
    int price;
    int 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 class Main {
    public static void main(String args[]) {
        Scanner s = new Scanner(System.in);
        int n, price, numPages;
        String name, author;
        System.out.print("Enter number of books: ");
        n = s.nextInt();
        Books[] b = new Books[n];
        for (int i = 0; i < n; i++) {
            System.out.println("\nEnter details of Book " + (i + 1) + ":");
            System.out.print("Enter book name: ");
            name = s.next();
            System.out.print("Enter author name: ");
```

```

        author = s.next();
        System.out.print("Enter price: ");
        price = s.nextInt();
        System.out.print("Enter number of pages: ");
        numPages = s.nextInt();
        b[i] = new Books(name, author, price, numPages);
    }
    System.out.println("\n--- Book Details ---");
    for (int i = 0; i < n; i++) {
        System.out.println("\nBook " + (i + 1) + " Details:");
        System.out.println(b[i].toString());
    }
    s.close();
}
}

```



```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
● PS D:\1BF24CS293> & 'C:\Program Files\Java\jdk-19\bin\java.exe' '-XX:+ShowCo
jdt_ws\1BF24CS293_83d9c4be\bin' 'Main'
Enter number of books: 2

Enter details of Book 1:
Enter book name: java
Enter author name: james
Enter price: 500
Enter number of pages: 800

Enter details of Book 2:
Enter book name: programs
Enter author name: charles
Enter price: 699
Enter number of pages: 880

--- Book Details ---

Book 1 Details:
Book name: java
Author name: james
Price: 500
Number of pages: 800

Book 2 Details:
Book name: programs
Author name: charles
Price: 699
Number of pages: 880

○ PS D:\1BF24CS293> 

```

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.

Code:

```
import java.util.Scanner;
abstract class Shape {
    int value1;
    int value2;
    public abstract void printArea();
}
class Rectangle extends Shape {
    public Rectangle(int length, int width) {
        this.value1 = length;
        this.value2 = width;
    }
    public void printArea() {
        // Area = Length * Width
        double area = (double)value1 * value2;
        System.out.printf("Area of Rectangle (L=%d, W=%d): %.2f\n", value1, value2, area);
    }
}
class Triangle extends Shape {
    public Triangle(int base, int height) {
        this.value1 = base;
        this.value2 = height;
    }
    public void printArea() {
        // Area = 0.5 * Base * Height
        double area = 0.5 * value1 * value2;
        System.out.printf("Area of Triangle (B=%d, H=%d): %.2f\n", value1, value2, area);
    }
}
class Circle extends Shape {
    private static final double PI = Math.PI;
    public Circle(int radius) {
        this.value1 = radius;
        this.value2 = 0;
    }
}
```

```

    public void printArea() {
        double area = PI * value1 * value1;
        System.out.printf("Area of Circle (Radius=%d): %.2f\n", value1, area);
    }
}

public class shapearea {
    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);
        System.out.println("--- Shape Area Calculation Program ---");
        System.out.print("Enter Rectangle Length: ");
        int rLength = scanner.nextInt();
        System.out.print("Enter Rectangle Width: ");
        int rWidth = scanner.nextInt();
        Rectangle rect = new Rectangle(rLength, rWidth);
        rect.printArea();
        System.out.print("\nEnter Triangle Base: ");
        int tBase = scanner.nextInt();
        System.out.print("Enter Triangle Height: ");
        int tHeight = scanner.nextInt();
        Triangle tri = new Triangle(tBase, tHeight);
        tri.printArea();
        System.out.print("\nEnter Circle Radius: ");
        int cRadius = scanner.nextInt();
        Circle circ = new Circle(cRadius);
        circ.printArea();

        scanner.close();

    }
}

```

```

PS D:\1BF24C5293> & 'C:\Program Files\Java\jdk-11
jdt_ws\1BF24C5293_83d9c4be\bin' 'EasyShapeArea'
--- Shape Area Calculation Program ---
Enter Rectangle Length: 2
Enter Rectangle Width: 3
Area of Rectangle (L=2, W=3): 6.00

Enter Triangle Base: 2
Enter Triangle Height: 3
Area of Triangle (B=2, H=3): 3.00

Enter Circle Radius: 7
Area of Circle (Radius=7): 153.94
PS D:\1BF24C5293>

```

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

Code:

```
import java.util.Scanner;
class Account {
    String customerName;
    int accountNumber;
    String accountType;
    double balance;
    void getAccountDetails()
    {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter customer name: ");
        customerName = s.next();
        System.out.print("Enter account Number: ");
        accountNumber = s.nextInt();
        System.out.print("Enter type of account (saving/current): ");
        accountType = s.next();
        balance = 0;
    }
    void display() {
        System.out.println("Customer name: " + customerName);
        System.out.println("Account number: " + accountNumber);
        System.out.println("Type of Account: " + accountType);
        System.out.println("Balance = " + balance);
    }
}
class Sav_acct extends Account {
```

```

void deposit()
{
    Scanner s = new Scanner(System.in);
    System.out.print("Enter the deposit amount: ");
    double amount = s.nextDouble();
    balance += amount;
}

void withdraw()
{
    Scanner s = new Scanner(System.in);
    System.out.print("Enter the withdrawal amount: ");
    double amount = s.nextDouble();
    if (amount > balance) {
        System.out.println("Insufficient balance!");
    } else {
        balance -= amount;
    }
}

void computeInterest()
{
    Scanner s = new Scanner(System.in);
    System.out.print("Enter the rate of interest: ");
    double rate = s.nextDouble();
    System.out.print("Enter the time period (years): ");
    int time = s.nextInt();
    double interest = balance * Math.pow((1 + rate / 100), time) - balance;
    balance += interest;
    System.out.println("Interest added = " + interest);
}
}

class Cur_acct extends Account {
    final double minBalance = 500; // minimum required balance
    final double serviceCharge = 100;
    void deposit()
    {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter the deposit amount: ");
        double amount = s.nextDouble();
        balance += amount;
    }
    void withdraw()

```

```

{
    Scanner s = new Scanner(System.in);
    System.out.print("Enter the withdrawal amount: ");
    double amount = s.nextDouble();
    if (amount > balance) {
        System.out.println("Insufficient balance!");
    } else {
        balance -= amount;
        checkMinBalance();
    }
}
}
void checkMinBalance()
{
    if (balance < minBalance) {
        balance -= serviceCharge;
        System.out.println("Balance below minimum! Service charge of Rs." + serviceCharge + "
imposed.");
    }
}
}
}
public class maibank {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        Sav_acct sav = new Sav_acct();
        Cur_acct cur = new Cur_acct();
        System.out.print("Enter customer name for savings account: ");
        sav.customerName = s.next();
        System.out.print("Enter account Number: ");
        sav.accountNumber = s.nextInt();
        sav.accountType = "saving";
        System.out.print("Enter customer name for current account: ");
        cur.customerName = s.next();
        System.out.print("Enter account Number: ");
        cur.accountNumber = s.nextInt();
        cur.accountType = "current";
        int choice;
        do {
            System.out.println("\n-----MENU-----");
            System.out.println("1. Deposit");
            System.out.println("2. Withdraw");
            System.out.println("3. Compute interest for SavingsAccount");

```



```

System.out.println("4. Display account details");
System.out.println("5. Exit");
System.out.print("Enter your choice: ");
choice = s.nextInt();
switch (choice) {
    case 1:
        System.out.print("Enter the type of account: ");
        String type = s.next();
        if (type.equalsIgnoreCase("saving"))
            sav.deposit();
        else
            cur.deposit();
        break;
    case 2:
        System.out.print("Enter the type of account: ");
        type = s.next();
        if (type.equalsIgnoreCase("saving"))
            sav.withdraw();
        else
            cur.withdraw();
        break;
    case 3:
        sav.computeInterest();
        break;
    case 4:
        System.out.print("Enter the type of account: ");
        type = s.next();
        if (type.equalsIgnoreCase("saving"))
            sav.display();
        else
            cur.display();
        break;
    case 5:
        System.out.println("Exiting...");
        break;
    default:
        System.out.println("Invalid choice!");
}
} while (choice != 5);
}
}

```

```

PS D:\1BF24CS293> & "C:\Program Files\Java\jdk-19\bin\java.exe" "-XX:+ShowCodeDetails"
.\java\jdt_ws\1BF24CS293_83d9c4be\bin\ 'maibank'
Enter customer name for savings account: john
Enter account Number: 1
Enter customer name for current account: smith
Enter account Number: 2

----MENU-----
1. Deposit
2. Withdraw
3. Compute interest for SavingsAccount
4. Display account details
5. Exit
Enter your choice: 1
Enter the type of account: saving
Enter the deposit amount: 10000

----MENU-----
1. Deposit
2. Withdraw
3. Compute interest for SavingsAccount
4. Display account details
5. Exit
Enter your choice: 2
Enter the type of account: saving
Enter the withdrawal amount: 200

----MENU-----
1. Deposit
2. Withdraw
3. Compute interest for SavingsAccount
4. Display account details
5. Exit
Enter your choice: 4
Enter the type of account: saving
Customer name: john
Account number: 1
Type of Account: saving
Balance = 10000.0

----MENU-----
1. Deposit
2. Withdraw
3. Compute interest for SavingsAccount
4. Display account details
5. Exit
Enter your choice: 1
Enter the type of account: current
Enter the deposit amount: 2305

----MENU-----
1. Deposit
2. Withdraw
3. Compute interest for SavingsAccount
4. Display account details
5. Exit
Enter your choice: 2
Enter the type of account: current
Enter the withdrawal amount: 300

----MENU-----
1. Deposit
2. Withdraw
3. Compute interest for SavingsAccount
4. Display account details
5. Exit
Enter your choice: 4
Enter the type of account: current
Customer name: smith
Account number: 2
Type of Account: current
Balance = 2005.0

----MENU-----
1. Deposit
2. Withdraw
3. Compute interest for SavingsAccount
4. Display account details
5. Exit
Enter your choice: 5
Exiting...
PS D:\1BF24CS293>

```

Program 6

Create a package CIE which has two classes - Personal 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 Personal. 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:

```
// cie/Personal.java
package cie;
public class Personal {
    public String usn, name;
    public int sem;
    public Personal(String usn, String name, int sem) {
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }
}

// cie/Internals.java
package cie;
public class Internals {
    public int[] internalMarks = new int[5];
}

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

// FinalMarks.java
import cie.*;
import see.*;
import java.util.*;
public class FinalMarks {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of students: ");
        int n = sc.nextInt();
        External[] students = new External[n];
        Internals[] internals = new Internals[n];
        for (int i = 0; i < n; i++) {
            System.out.println("\nEnter details for student " + (i + 1));
            System.out.print("USN: ");
```


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=father’s age.

Code:

```
import java.util.*;
class WrongAge extends Exception {
    WrongAge() {
        super("Age Error!");
    }
    WrongAge(String msg) {
        super(msg);
    }
}
class InputScanner {
    Scanner s = new Scanner(System.in);
}
class Father extends InputScanner {
    int fatherAge;
    Father() throws WrongAge {
        System.out.print("Enter father's age: ");
        fatherAge = s.nextInt();
        if (fatherAge < 0)
            throw new WrongAge("Age cannot be negative");
    }
    void display() {
        System.out.println("Father's age: " + fatherAge);
    }
}
class Son extends Father {
    int sonAge;
    Son() throws WrongAge {
        super();
        System.out.print("Enter son's age: ");
        sonAge = s.nextInt();
        if (sonAge >= fatherAge)
            throw new WrongAge("Son's age cannot be greater than or equal to father's age");
        else if (sonAge < 0 && fatherAge < 0)
            throw new WrongAge("Age cannot be negative");
    }
    void display() {
        System.out.println("Son's age: " + sonAge);
    }
}
public class Main {
```

```

public static void main(String[] args) {
    try {
        Son obj = new Son();
        System.out.println("\n--- Age Details ---");
        obj.display();
    } catch (WrongAge e) {
        System.out.println("Exception caught: " + e.getMessage());
    }
}
}

```

```

PS C:\Simrik-1BF24CS293> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Admin\AppData\Roaming\Code\User\workspaceStorage\67e9b8863ba13b6d
Enter father's age: 48
Enter son's age: 50
Exception caught: Son's age cannot be greater than or equal to Father's age
PS C:\Simrik-1BF24CS293> ^C
PS C:\Simrik-1BF24CS293>
PS C:\Simrik-1BF24CS293> c;; cd 'c:\Simrik-1BF24CS293'; & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Admin\AppData\Roaming\Code\User\wo
rkspaceStorage\67e9b8863ba13b6d1b5b91f64e292462\redhat_java\jdt_ws\Simrik-1BF24CS293_1dd5bc9d\bin' 'Main'
Enter father's age: 48
Enter son's age: 19

--- Age Details ---
Son's age: 19
Son's age: 19
PS C:\Simrik-1BF24CS293> ^C
PS C:\Simrik-1BF24CS293>
PS C:\Simrik-1BF24CS293> c;; cd 'c:\Simrik-1BF24CS293'; & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Admin\AppData\Roaming\Code\User\wo
rkspaceStorage\67e9b8863ba13b6d1b5b91f64e292462\redhat_java\jdt_ws\Simrik-1BF24CS293_1dd5bc9d\bin' 'Main'
Enter father's age: 48
Enter son's age: -19
Exception caught: Age cannot be negative
PS C:\Simrik-1BF24CS293>

```

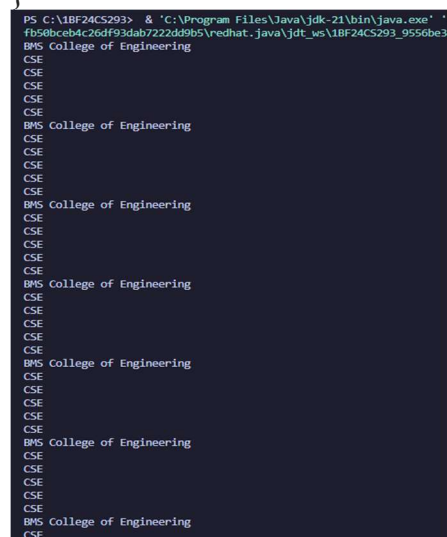
Program 8

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 sample extends Thread {
    private String message;
    private int interval;
    public sample(String message, int interval) {
        this.message = message;
        this.interval = interval;
    }
    public void run() {
        while (true) {
            System.out.println(message);
            try {
                Thread.sleep(interval);
            } catch (InterruptedException e) {
                System.out.println("Thread interrupted");
            }
        }
    }
}

public class Main {
    public static void main(String[] args) {
        sample t1 = new sample("BMS College of Engineering", 10000);
        sample t2 = new sample("CSE", 2000);
        t1.start();
        t2.start();
    }
}
```



The screenshot shows a terminal window with the following output:

```
P5 C:\1BF24C5293> & "C:\Program Files\Java\jdk-21\bin\java.exe" ^
fb50bceb4c2edf93dab722dd9b5\redhat_java\jdt_ws\1BF24C5293_9556be3
BMS College of Engineering
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
```

