

**VISVESVARAYA TECHNOLOGICAL
UNIVERSITY**

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT

on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

SHRIHARI VISWANATHAN (1BF24CS288)

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 **SHRIHARI VISWANATHAN (1BF24CS288)**, 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	QUADRATIC EQUATION	4
2	13/10/25	SGPA CALCULATION	6
3	20/10/25	BOOKS (TO STRING)	9
4	04/11/25	ABSTRACT CLASS DEMO	12
5	04/11/25	BANK	15
6	18/11/25	PACKAGES	20
7	25/11/25	EXCEPTION	25
8	09/12/25	THREADS	27

Github Link:

[https://github.com/ShrihariViswanathan/Semester-3/tree/main/Java\(OOJ\)](https://github.com/ShrihariViswanathan/Semester-3/tree/main/Java(OOJ))

Lab Program 1

QUADRATIC EQUATION

```
1  /*Develop a Java program that prints all real solutions to the quadratic
2   equation ax2+bx+c = 0. Read in a, b, c and use the quadratic formula. If
3   the discriminate b2-4ac is negative, display a message stating that there
4   are no real solutions.*/
5
6  import java.util.Scanner;
7
8  class QUADRATIC_EQUATION{
9    public static void main(String args[]){
10      double a;
11      double b;
12      double c;
13
14      Scanner scan = new Scanner(System.in);
15
16
17      System.out.print("a: ");
18      a = scan.nextDouble();
19
20      System.out.print("b: ");
21      b = scan.nextDouble();
22
23      System.out.print("c: ");
24      c = scan.nextDouble();
25
26      double disc = (b*b) - (4*a*c);
27
28      if (disc > 0){
29          System.out.println("The solutions are real");
30          double alpha = (-b + Math.pow(disc, 1/2))/ (2 * a);
31          double beta = (-b - Math.pow(disc, 1/2)) / (2 * a);
32          System.out.println("Alpha: " + alpha);
33          System.out.println("Beta: " + beta);
34      }
35
36      if (disc == 0){
37          System.out.println("The solutions are equal");
38          double sol = (-b) / (2*a);
39          System.out.println("Beta: " + sol);
40          System.out.println("Alpha: " + sol);
41      }
42
43      if (disc < 0 ){
44          System.out.println("The roots are imaginary");
45      }
46
47
48
49      scan.close();
50
51
52    }
53
54
55
56 }
```

OUTPUT:

```
PS C:\1bf24cs288> cd "c:\1bf24cs288\" ; if ($? -eq 0) {<br/>a: 1<br/>b: 1<br/>c: 1<br/>The roots are imaginary<br/>PS C:\1bf24cs288> cd "c:\1bf24cs288\" ; if ($? -eq 0) {<br/>a: 1<br/>b: -5<br/>c: 6<br/>The solutions are real<br/>Alpha: 3.0<br/>Beta: 2.0<br/>PS C:\1bf24cs288> cd "c:\1bf24cs288\" ; if ($? -eq 0) {<br/>a: 1<br/>b: -2<br/>c: 1<br/>The solutions are equal<br/>Beta: 1.0<br/>Alpha: 1.0<br/>PS C:\1bf24cs288> █
```

Lab Program 2

SGPA CALCULATION

```
1 import java.util.Scanner;
2
3 class Subject {
4     int subjectMarks;
5     int credits;
6     int grade;
7 }
8
9 class Student {
10    String name;
11    String usn;
12    double SGPA;
13    Scanner s;
14    Subject[] subject;
15
16    // Constructor
17    Student() {
18        s = new Scanner(System.in);
19        subject = new Subject[8]; // 8 subjects
20        for (int i = 0; i < 8; i++) {
21            subject[i] = new Subject();
22        }
23    }
24
25    // Method to get student's name and USN
26    void getStudentDetails() {
27        System.out.print(s: "Student Name: ");
28        name = s.nextLine();
29        System.out.print(s: "USN: ");
30        usn = s.nextLine();
31    }
32
33    // Method to read marks and credits and calculate grades
34    void getMarks() {
35        for (int i = 0; i < 8; i++) {
36            System.out.println("Enter Details for Subject " + (i + 1));
37
38            // Input marks
39            System.out.print(s: "Marks (0-100): ");
40            subject[i].subjectMarks = s.nextInt();
41            while(subject[i].subjectMarks < 0 || subject[i].subjectMarks > 100) {
42                System.out.println(x: "Invalid marks! Please enter marks between 0 and 100.");
43                System.out.print(s: "Marks (0-100): ");
44                subject[i].subjectMarks = s.nextInt();
45            }
46
47            // Input credits
48            System.out.print(s: "Credits: ");
49            subject[i].credits = s.nextInt();
50            while(subject[i].credits <= 0) {
51                System.out.println(x: "Invalid credits! Please enter positive credits.");
52                System.out.print(s: "Credits: ");
53                subject[i].credits = s.nextInt();
54            }
55
56            // Calculate grade based on marks
57            subject[i].grade = (subject[i].subjectMarks / 10) + 1;
58
59            // If marks > 100, grade capped at 10 (already checked but just in case)
60            if (subject[i].grade == 11)
61                subject[i].grade = 10;
62        }
```

```

63         // If marks < 40, grade set to 0 (fail)
64         if (subject[i].subjectMarks < 40)
65             subject[i].grade = 0;
66     }
67 }
68
69 // Calculate SGPA
70 void computeSGPA() {
71     int effectiveScore = 0;
72     int totalCredits = 0;
73
74     for (int i = 0; i < 8; i++) {
75         effectiveScore += (subject[i].grade * subject[i].credits);
76         totalCredits += subject[i].credits;
77     }
78
79     SGPA = (double) effectiveScore / (double) totalCredits;
80 }
81
82 // Display student details and SGPA
83 void display() {
84     System.out.println("Student Details:");
85     System.out.println("Name: " + name);
86     System.out.println("USN: " + usn);
87     System.out.printf("SGPA: %.2f\n", SGPA);
88
89
90     System.out.println();
91 }
92 }
93 }
94 }
95 }
96
97 class cgpa{
Run | Debug | Run main | Debug main
98     public static void main(String[] args) {
99
100         Student students[] = new Student[2];
101         for (int i = 0; i < 2; i++){
102             students[i] = new Student();
103             students[i].getStudentDetails();
104             students[i].getMarks();
105             students[i].computeSGPA();
106             students[i].display();
107         }
108     }
109 }
110 }
```

OUTPUT:

```
PS C:\1bf24cs288> cd "c:\1bf24cs288\" ; if ($?) { javac cgpa.java } ; if (?) { java cg
Student Name: shrihari
USN: 1bf24cs288
Enter Details for Subject 1
Marks (0-100): 100
Student Name: shrihari
USN: 1bf24cs288
Enter Details for Subject 1
Marks (0-100): 100
USN: 1bf24cs288
Enter Details for Subject 1
Marks (0-100): 100
Credits: 3
Marks (0-100): 100
Credits: 3
Enter Details for Subject 2
Credits: 3
Enter Details for Subject 2
Enter Details for Subject 2
Marks (0-100): 90
Credits: 4
Enter Details for Subject 3
Credits: 4
Enter Details for Subject 3
Marks (0-100): 95
Credits: 2
Enter Details for Subject 3
Marks (0-100): 95
Credits: 2
Marks (0-100): 95
Credits: 2
Credits: 2
Enter Details for Subject 4
Marks (0-100): 90
Credits: 1
Enter Details for Subject 5
Marks (0-100): 90
Credits: 3
Enter Details for Subject 6
Marks (0-100): 100
Credits: 3
Enter Details for Subject 7
Marks (0-100): 100
Credits: 4
Enter Details for Subject 8
Marks (0-100): 100
Credits: 2

Student Details:
Name: shrihari
USN: 1bf24cs288
SGPA: 10.00
```

Lab Program 3: TO STRING DEMO

```
J test.java > ...
1  import java.util.Scanner;
2
3  class Book{
4      String name;
5      String author;
6      double price;
7      int numPages;
8
9      Book(String name, String author, double price, int numPages){
10         this.name = name;
11         this.author = author;
12         this.price = price;
13         this.numPages = numPages;
14     }
15
16
17     public String toString(){
18         String name, author, price, numPages;
19         name = "Book name: " + this.name + "\n";
20         author = "Author name: " + this.author + "\n";
21         price = "Price: " + this.price + "\n";
22         numPages = "Number of pages: " + this.numPages + "\n";
23
24         return name + author + price + numPages;
25     }
26
27 } //end of class
28
29
30 class Books{
31     Run | Debug | Run main | Debug main
32     public static void main(String args[]){
```

```

33     Scanner s = new Scanner(System.in);
34
35     int n;
36     String name;
37     String author;
38     double price;
39     int numPages;
40
41     System.out.print(s: "No.of Books: ");
42     n = s.nextInt();
43     s.nextLine();
44
45     Book b[];
46     b = new Book[n];
47
48     for (int i = 0; i < n; i++){
49         //name
50
51         System.out.print("Name of Book " + (i+1));
52         name = s.nextLine();
53
54         System.out.print("Author of Book " + (i + 1));
55         author = s.nextLine();
56
57         System.out.print("Price of Book " + (i + 1));
58         price = s.nextDouble();
59
60         System.out.print("No.of Pages for Book " + (i + 1)) ;
61         numPages = s.nextInt();
62
63         b[i] = new Book(name, author, price, numPages);
64     }
65
66
67     for (int i = 0; i < b.length; i++){
68         System.out.println(b[i]);
69     }
70
71     }
72 }
```

OUTPUT:

```
D:\Educat\Java\Java_Ws\Java_Scm_C1230444\bmi---book
```

No.of Books: 1
Name of Book 1cruel prince
Author of Book 1holly jackson
Price of Book 1500
No.of Pages for Book 1358
Book name: cruel prince
Author name: holly jackson
Price: 500.0
Number of pages: 358

Lab Program 4:

ABSTRACT CLASS DEMO

```
1 import java.util.Scanner;
2
3 class Input {
4     Scanner sc = new Scanner(System.in);
5
6     int getInt(String message) {
7         System.out.print(message);
8         return sc.nextInt();
9     }
10 }
11
12 abstract class Shape extends Input {
13     int a, b;
14     abstract void printArea();
15 }
16
17 class Rectangle extends Shape {
18     void input() {
19         a = getInt(message: "Enter length: ");
20         b = getInt(message: "Enter breadth: ");
21     }
22
23     void printArea() {
24         System.out.println("Area of rectangle: " + (a * b));
25         System.out.println();
26     }
27 }
28
29 class Triangle extends Shape {
30     void input() {
31         a = getInt(message: "Enter base value: ");
32         b = getInt(message: "Enter height value: ");
```

```
33     }
34
35     void printArea() {
36         System.out.println("Area of Triangle: " + (0.5 * a * b));
37         System.out.println();
38     }
39 }
40
41 class Circle extends Shape {
42     void input() {
43         a = getInt(message: "Enter the radius: ");
44     }
45
46     void printArea() {
47         System.out.println("Area of Circle: " + (3.14159265 * a * a));
48         System.out.println();
49     }
50 }
51
52 public class abstractClass {
53     Run | Debug | Run main | Debug main
54     public static void main(String args[]) {
55         Rectangle r = new Rectangle();
56         r.input();
57         r.printArea();
58
59         Triangle t = new Triangle();
60         t.input();
61         t.printArea();
62         t.printArea();
63
64         Circle c = new Circle();
65         c.input();
66         c.printArea();
67     }
68 }
```

OUTPUT:

```
PS D:\1BF24CS288> cd "d:\1BF24CS288\"  
va abstractClass }  
Enter length: 12  
Enter breadth: 3  
Area of rectangle: 36  
  
Enter base value: 4  
Enter height value:5  
Area of Triangle: 10.0  
  
Enter the radius: 6  
Area of Circle: 113.09733540000002
```

Lab Program 5:

BANK

```
1 import java.util.Scanner;
2
3 class Account{
4     String customerName;
5     int accountNumber;
6     String accountType;
7     double balance;
8
9     Scanner sc = new Scanner(System.in);
10
11    void input(){
12        System.out.print(s: "Enter Customer Name: ");
13        customerName = sc.nextLine();
14
15        System.out.print(s: "Enter Account Number: ");
16        accountNumber = sc.nextInt();
17        sc.nextLine();
18    }
19
20
21    void deposit(){
22        System.out.print(s: "Enter amount to deposit: ");
23        double amount = sc.nextDouble();
24
25        balance += amount;
26        System.out.println(x: "Amount deposited Successfully");
27    }
28
29
30    void displayBalance(){
31        System.out.println("Current Balance: " + balance);
32    }
```

```

33
34     void withdraw(){
35         System.out.print(s: "Enter amount to withdraw: ");
36         double amount = sc.nextDouble();
37
38         if(amount <= balance){
39             balance -= amount;
40             System.out.println("You have " + amount + " rupees");
41             System.out.println(x: "Transaction Successful");
42             checkMinBalance();
43         }
44
45         else{
46             System.out.println(x: "Insufficient balance");
47         }
48     }
49
50     void checkMinBalance(){}
51
52 }
53
54 }
55
56
57
58 class Savings extends Account{
59     final double interestRate = 0.05;
60     Savings() {
61         accountType = "Savings";
62     }
63
64     void computeInterest(){
65         System.out.print(s: "Enter no.of Years: ");
66         double years = sc.nextDouble();
67
68         double interest = balance * Math.pow((1 + interestRate), years) - balance;
69         balance += interest;
70
71         System.out.println("Interest Added: " + interest);
72     }
73 }
74
75
76
77 class Current extends Account{
78     final double serviceCharge = 100;
79     final double minBalance = 1000;
80
81     Current() {
82         accountType = "Current";
83     }
84
85
86     @Override
87     void checkMinBalance(){
88         if (balance < minBalance){
89             balance -= serviceCharge;
90             System.out.println(x: "Balance < minBalance");
91             System.out.print("Penalty of " + serviceCharge + " is imposed");
92         }
93     }
94 }
```

```

95  }
96
97
98 public class bank {
Run | Debug | Run main | Debug main
99     public static void main(String[] args) {
100         Scanner sc = new Scanner(System.in);
101
102         System.out.print(s: "Enter account type (Savings/Current): ");
103         String accountType = sc.nextLine();
104
105         Account account = null;
106
107         if (accountType.trim().equalsIgnoreCase(anotherString: "Savings")) {
108             account = new Savings();
109         } else if (accountType.equalsIgnoreCase(anotherString: "Current")) {
110             account = new Current();
111         } else {
112             System.out.println(x: "Invalid account type.");
113             return;
114         }
115
116         account.input();
117
118         while (true) {
119             printMenu();
120             System.out.print(s: "Enter your choice: ");
121             int choice = sc.nextInt();
122
123             switch (choice) {
124                 case 1:
125                     account.deposit();
126                     break;
127                 case 2:
128                     account.withdraw();
129                     break;
130                 case 3:
131                     account.displayBalance();
132                     break;
133                 case 4:
134                     if (account instanceof Savings) {
135                         ((Savings) account).computeInterest();
136                     }
137
138                     else {
139                         System.out.println(x: "Compute Interest not available for Current accounts.");
140                     }
141                     break;
142                 case 5:
143                     System.out.println(x: "Exiting the bank system.");
144                     return;
145                 default:
146                     System.out.println(x: "Invalid choice. Please try again.");
147             }
148         }
149     }
150
151     public static void printMenu() {
152         System.out.println(x: "\n1. Deposit");
153         System.out.println(x: "2. Withdraw");
154         System.out.println(x: "3. Display Balance");
155         System.out.println(x: "4. Compute Interest (Only for Savings accounts)");
156         System.out.println(x: "5. Exit");
157     }
158
159 }

```

OUTPUT:

```
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Only for Savings accounts)
5. Exit
Enter your choice: 2
Enter amount to withdraw: 3
Insufficient balance

1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Only for Savings accounts)
5. Exit
Enter your choice: 2
Enter amount to withdraw: 150
You have 150.0 rupees
Transaction Successful
Balance < minBalance
Penalty of 100.0 is imposed

Enter account type (Savings/Current): current
Enter Customer Name: name12
Enter Account Number: 1234

1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Only for Savings accounts)
5. Exit
Enter your choice: 1
Enter amount to deposit: 250
Amount deposited Successfully
```

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

```
Enter Customer Name: name1
```

```
Enter Account Number: 1234
```

- 1. Deposit
- 2. Withdraw
- 3. Display Balance
- 4. Compute Interest (Only for Savings accounts)
- 5. Exit

```
Enter your choice: 1
```

```
Enter amount to deposit: 125
```

```
Amount deposited Successfully
```

- 1. Deposit
- 2. Withdraw
- 3. Display Balance
- 4. Compute Interest (Only for Savings accounts)
- 5. Exit

```
Enter your choice: 2
```

```
Enter amount to withdraw: 125
```

```
You have 125.0 rupees
```

```
Transaction Successful
```

Lab Program 6

PACKAGES

Folder Order :

Main.java
SEE/
 Externals.java
CIE/
 Internals.java
 Personal.java

1. Externals.java

```
1 package SEE;
2
3 import CIE.Internals;
4 import java.util.Scanner;
5
6 /**
7  * Derived from Internals, stores Semester End Examination (SEE) marks and calculates final marks.
8  */
9 public class Externals extends Internals {
10
11     // This array stores SEE marks (out of 100).
12     protected int marks[] = new int[5];
13     // Stores the final calculated marks (out of 100).
14     protected int finalMarks[] = new int[5];
15
16
17
18     /**
19      * Reads SEE marks for 5 courses using the provided Scanner object.
20      * @param s The shared Scanner instance.
21      */
22     public void inputSEEMarks(Scanner s) {
23         System.out.println("n--- Entering SEE Marks (out of 100) ---");
24         for (int i = 0; i < 5; i++) {
25             System.out.print("Enter SEE mark for Course " + (i + 1) + " (max 100): ");
26             while (!s.hasNextInt()) {
27                 System.out.println("Invalid input. Please enter a number: ");
28                 s.next();
29             }
30             // 'this.marks' refers to the SEE marks array
31             this.marks[i] = s.nextInt();
32             // Basic validation
```

```

33     |         if (this.marks[i] < 0 || this.marks[i] > 100) {
34     |             System.out.println(x: "Warning: Mark is outside the 0-100 range. Continuing...");
35     |
36     }
37     s.nextLine(); // Consume newline
38 }
39
40 /**
41 * Calculates final marks using the formula: Final = CIE Mark + (SEE Mark / 2).
42 */
43 public void calculateFinalMarks() {
44     System.out.println(x: "Calculating final marks...");
45     for (int i = 0; i < 5; i++) {
46         // super.marks refers to the CIE marks array from the Internals class.
47         int cie = super.marks[i];
48         // this.marks refers to the SEE marks array from the Externals class.
49         int see = this.marks[i];
50
51         // Final Mark (out of 100) = CIE (50) + (SEE/2) (50)
52         this.finalMarks[i] = cie + (see / 2);
53
54         // Cap the final mark at 100
55         if (this.finalMarks[i] > 100) {
56             this.finalMarks[i] = 100;
57         }
58     }
59 }
60
61 /**
62 * Displays all student details and the final calculated marks.
63 */
64 public void displayFinalMarks() {
65     System.out.println(x: "\n=====");
66     System.out.println(x: " Final Marks Report");
67     System.out.println(x: "=====");
68     // Explicitly call inherited method from Student class
69     super.displayStudentDetails();
70
71     System.out.println(x: "\nCourse | CIE (50) | SEE (100) | Final (100)");
72     System.out.println(x: "-----");
73     for (int i = 0; i < 5; i++) {
74         // Accessing CIE marks (super.marks) and SEE marks (this.marks)
75         System.out.printf(format: "%d | %d | %d | %d\n",
76                           (i + 1), super.marks[i], this.marks[i], this.finalMarks[i]);
77     }
78 }
79 }
80 }
```

2. Internals.java

```
1 package CIE;
2 import java.util.Scanner;
3
4
5 public class Internals extends Personal{
6
7
8
9
10    // Stores CIE marks for 5 courses (out of 50).
11    protected int marks[] = new int[5];
12
13
14    public void inputCIEmarks(Scanner s) {
15        // Using the passed shared Scanner instance.
16        System.out.println(x: "\n--- Entering CIE Marks (out of 50) ---");
17        for (int i = 0; i < 5; i++) {
18            System.out.print("Enter CIE mark for Course " + (i + 1) + " (max 50): ");
19            while (!s.hasNextInt()) {
20                System.out.println(x: "Invalid input. Please enter a number: ");
21                s.next();
22            }
23            marks[i] = s.nextInt();
24            // Basic validation
25            if (marks[i] < 0 || marks[i] > 50) {
26                System.out.println(x: "Warning: Mark is outside the 0-50 range. Continuing...");
27            }
28        }
29        s.nextLine(); // Consume newline
30    }
31 }
32 }
```

3. Main.java

```

1 import SEE.Externals;
2 import java.util.Scanner;
3
4 /**
5  * Main class to demonstrate the package structure, inheritance,
6  * and calculation of final marks for n students.
7  * FIX: The single Scanner 's' is now passed to all input methods
8  * to prevent input stream corruption.
9 */
10 class Main {
    Run | Debug | Run main | Debug main
11     public static void main(String args[]) {
12         // Only one Scanner object is created here to manage System.in
13         Scanner s = new Scanner(System.in);
14         int n = 0;
15
16         System.out.println(x: "--- Student Mark Calculation System ---");
17         System.out.print(s: "Enter the number of students (n): ");
18         if (s.hasNextInt()) {
19             n = s.nextInt();
20             if (n <= 0) {
21                 System.out.println(x: "Number of students must be positive. Setting n to 1.");
22                 n = 1;
23             }
24         } else {
25             System.out.println(x: "Invalid input. Setting number of students to 1.");
26             n = 1;
27         }
28
29         s.nextLine(); // Consume the rest of the line after reading n
30
31         // Array to hold n instances of the Externals class
32         Externals students[] = new Externals[n];
33
34         for (int i = 0; i < n; i++) {
35             System.out.println("\n--- Entering details for Student " + (i + 1) + " of " + n + " ---");
36
37             students[i] = new Externals();
38
39             // 2. Input student details - PASSING the Scanner
40             students[i].inputStudentDetails(s);
41
42             // 3. Input CIE marks - PASSING the Scanner
43             students[i].inputCIEmarks(s);
44
45             // 4. Input SEE marks - PASSING the Scanner
46             students[i].inputSEEmarks(s);
47         }
48
49         System.out.println(x: "\n*****");
50         System.out.println(x: "          FINAL RESULTS REPORT      ");
51         System.out.println(x: "*****");
52
53         for (int i = 0; i < n; i++) {
54             // 5. Calculate Final marks
55             students[i].calculateFinalMarks();
56
57             // 6. Display Final marks
58             students[i].displayFinalMarks();
59         }
60
61         s.close(); // Close the main scanner resource ONLY once
62     }
63 }

```

4. Personal.java

```
1 package CIE;
2 import java.util.Scanner;
3
4 public class Personal {
5
6
7     protected String usn = "";
8     protected String name = "";
9     protected int sem;
10
11    public void inputStudentDetails(Scanner s) {
12        // Using the passed shared Scanner instance.
13        System.out.print(s: "Enter Student USN: ");
14        this.usn = s.nextLine();
15        System.out.print(s: "Enter Student Name: ");
16        this.name = s.nextLine();
17        System.out.print(s: "Enter Semester: ");
18        // Ensure input is an integer
19        while (!s.hasNextInt()) {
20            System.out.println(x: "Invalid input. Please enter a number for the Semester: ");
21            s.next(); // Consume invalid input
22        }
23        this.sem = s.nextInt();
24        s.nextLine(); // Consume newline for safety
25    }
26
27
28    public void displayStudentDetails() {
29        System.out.println("Student USN: " + usn);
30        System.out.println("Student Name: " + name);
31        System.out.println("Semester: " + sem);
32    }
33
34
35 }
```

Lab program 7

EXCEPTION

```
1 import java.util.Scanner;
2
3 class WrongAge extends Exception{
4     WrongAge(String message){
5         super(message);
6     }
7 }
8
9 class Father{
10     float fatherAge;
11
12     Father(float fatherAge){
13         try{
14             if (fatherAge < 0)
15                 throw new WrongAge(message: "Father cant be less than zero years old");
16             this.fatherAge = fatherAge;
17
18         } catch (WrongAge e){
19             System.out.println("Error: " + e);
20         }
21     }
22 }
23
24 class Son extends Father{
25     float sonAge;
26
27     Son(float fatherAge, float sonAge){
28         super(fatherAge);
29         try{
30             if (sonAge >= fatherAge)
31                 throw new WrongAge(message: "Son's age cannot be greater than father's age");
32             this.sonAge = sonAge;
33
34         } catch(WrongAge e){
35             System.out.println("Error: " + e);
36         }
37     }
38 }
39
40     Son (float fatherAge){
41         super(fatherAge);
42     }
43 }
44
45
46 public class LAB_07_EXCEPTION{
Run | Debug | Run main | Debug main
47     public static void main(String[] args) {
48         Scanner sc = new Scanner(System.in);
49
50         Son s1;
51
52         System.out.print(s: "Enter father age: ");
53         float fatherAge = sc.nextFloat();
54         if (fatherAge < 0){
55             s1 = new Son(fatherAge);
56             System.exit(status: 0);
57         }
58
59         System.out.print(s: "Enter Son age: ");
60         float sonAge = sc.nextFloat();
61         s1 = new Son(fatherAge, sonAge);
62     }
63
64 }
```

Output

```
PS C:\1bf24cs288> cd "c:\1bf24cs288\" ; if ($?) { javac LAB_07_EXCEPTION.java } ; if ($?) { java LAB_07_EXCEPTION }
Enter father age: -78
Error: WrongAge: Father cant be less than zero years old
PS C:\1bf24cs288> cd "c:\1bf24cs288\" ; if ($?) { javac LAB_07_EXCEPTION.java } ; if ($?) { java LAB_07_EXCEPTION }
Enter father age: 55
Enter Son age: 30
PS C:\1bf24cs288> cd "c:\1bf24cs288\" ; if ($?) { javac LAB_07_EXCEPTION.java } ; if ($?) { java LAB_07_EXCEPTION }
Enter father age: 55
Enter Son age: 66
Error: WrongAge: Son's age cannot be greater than father's age
PS C:\1bf24cs288> █
```

Lab program 8

THREADS

```
1  class CollegeThread implements Runnable {
2      public void run() {
3          try {
4              while(true) {
5                  System.out.println(x: "BMS College of Engineering");
6                  Thread.sleep(millis: 10000); // 10 seconds
7              }
8          } catch(InterruptedException e) {
9              System.out.println(x: "CollegeThread interrupted");
10         }
11     }
12 }
13
14 class CSEThread implements Runnable {
15     public void run() {
16         try {
17             while(true) {
18                 System.out.println(x: "CSE");
19                 Thread.sleep(millis: 2000); // 2 seconds
20             }
21         } catch(InterruptedException e) {
22             System.out.println(x: "CSEThread interrupted");
23         }
24     }
25 }
26
27 public class TwoThreadsDemo {
Run | Debug | Run main | Debug main
28     public static void main(String[] args) {
29         Thread t1 = new Thread(new CollegeThread());
30         Thread t2 = new Thread(new CSEThread());
31
32         t1.start();
33         t2.start();
34     }
35 }
```

OUTPUT

```
PS C:\1bf24cs288> cd "c:\1bf24cs288\" ; if ($?) { javac Tw
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
CSE
```