

# **VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

“JnanaSangama”, Belgaum -590014, Karnataka.



## **LAB REPORT on**

### **Object Oriented Java Programming (23CS3PCOOJ)**

*Submitted by*

**Siddu Lokeshwar Reddy (1BF24CS291)**

*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 **Siddu Lokeshwar Reddy (1BF24CS291)**, 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**

<b>Sl. No.</b>	<b>Date</b>	<b>ExperimentTitle</b>	<b>Page No.</b>
1	23/09/2025	Implement java program on Quadratic equation	4-5
2	14/10/2025	Implement java program to calculate SGPA	6-7
3	14/10/2025	Implement java program on Book class	8-9
4	04/11/2025	Implement java program on Abstract class	10-11
5	04/11/2025	Implement java program on Bank management	11-13
6	18/11/2025	Implement java program on Packages	14-15
7	25/11/2025	Implement java program on Exceptions	16-17
8	09/12/2025	Implement java program on Multithreading	18-
9			
10			

Github Link:

<https://github.com/siddulokeshwar/JAVAPROGRAMS-OOJ>

### 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:

Code:

```
1 import java.util.Scanner;
2 import java.lang.Math;
3 public class quadratic {
4     Run | Debug
5     public static void main(String args[])
6     {
7         Scanner s = new Scanner(System.in);
8         int a,b,c,d;
9         System.out.println("Enter coefficient of X^2:");
10        a=s.nextInt();
11        System.out.println("Enter coefficient of X:");
12        b=s.nextInt();
13        System.out.println("Enter a constant :");
14        c=s.nextInt();
15        if(a==0){
16            System.out.println("Not a quadratic equation");
17            System.out.println("Enter coefficient of X^2 other than zero(0):");
18            a=s.nextInt();
19        }
20        d=b*b-4*a*c;
21        if(d==0){
22            System.out.print(s: "Roots are real and equal:");
23            double r1 =((-b)/2*a);
24            System.out.println(r1);
25        }
26        else if(d>0)
27        {
28            System.out.print(s: "Roots are real and distinct:");
29            double r1 = ((-b) + (Math.sqrt(d)))/(double)(2*a);
30            double r2 = ((-b) - (Math.sqrt(d)))/(double)(2*a);
31            System.out.println(r1+" ,"+r2);
32        }
33        else if(d<0)
34        {
35            System.out.print(s: "Roots are not real :");
36            double r1 = (-b)/(2*a);
37            double r2 = Math.sqrt(-d)/(2*a);
38            System.out.println(r1+" ,"+r2);
39        }
40    }
41 }
```

output:

```
PS C:\18F24CS291> cd "c:\18F24CS291\" ; if ($?) { javac quadratic.java } ; if ($?) { java quadratic }
Enter coeffecient of X^2:
8
Enter coeffecient of X:
4
Enter a constant :
4
Not a quadratic equation
Enter coeffecient of X^2 other than zero(0):
1
Roots are real and equal and value is:-2.0
PS C:\18F24CS291> cd "c:\18F24CS291\" ; if ($?) { javac quadratic.java } ; if ($?) { java quadratic }
● Enter coeffecient of X^2:
3
Enter coeffecient of X:
5
Enter a constant :
1
Roots are real and distinct:-0.2324881207560018 ,-1.434258545910665
● PS C:\18F24CS291> cd "c:\18F24CS291\" ; if ($?) { javac quadratic.java } ; if ($?) { java quadratic }
Enter coeffecient of X^2:
-1
Enter coeffecient of X:
5
Enter a constant :
-12
Roots are not real :2.0 ,-2.3979157616563596
```

## 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:

```
1 import java.util.Scanner;
2 class Student {
3     Scanner sc = new Scanner(System.in);
4     class Subject {
5         int grade, marks, credits;
6     }
7
8     Subject s[] = new Subject[8];
9     String name, usn;
10    double SGPA;
11
12    void getStudentDetails() {
13        System.out.print("Enter student name: ");
14        name = sc.nextLine();
15        System.out.print("Enter USN: ");
16        usn = sc.nextLine();
17    }
18    int N;
19    void getMarks() {
20        System.out.print("Enter number of subjects: ");
21        N = sc.nextInt();
22        for (int i = 0; i < N; i++) {
23            s[i] = new Subject();
24        }
25        for (int i = 0; i < N; i++) {
26            System.out.print("Enter marks for Subject " + (i + 1) + ": ");
27            s[i].marks = sc.nextInt();
28            System.out.print("Enter credits for Subject " + (i + 1) + ": ");
29            s[i].credits = sc.nextInt();
30        }
31        for (int i = 0; i < N; i++) {
32            s[i].grade = (s[i].marks / 10) + 1;
33            if (s[i].grade == 11) {s[i].grade = 10;}
34            if (s[i].grade <= 4) {s[i].grade = 0;}
35        }
36    void computeSGPA() {
37        int effective_score = 0;
38        int total_credits = 0;
39        for (int i = 0; i < N; i++) {
40            effective_score += s[i].grade * s[i].credits;
41            total_credits += s[i].credits;
42        }
43        SGPA = (double) effective_score / total_credits;
44    }
45
46 public class cgpa{
47     Run|Debug
48     public static void main(String[] args) {
49         Scanner sc = new Scanner(System.in);
50         System.out.print("\n" + "Enter number of students: ");
51         int n = sc.nextInt();
52         sc.nextLine();
53         Student students[] = new Student[n];
54         for (int i = 0; i < n; i++) {
55             System.out.println("\n--- Student " + (i + 1) + " ---");
56             students[i] = new Student();
57             students[i].getStudentDetails();
58             students[i].getMarks();
59             students[i].computeSGPA();
60         }
61         System.out.println("===== SGPA Report =====");
62         for (int i = 0; i < n; i++) {
63             System.out.println("SGPA of " + students[i].name + " (" + students[i].usn + ") is: " + students[i].SGPA);
64         }
65 }
```

## output:

```
PS C:\IBF24CS291> & 'C:\Program Files\Java\jdk-21\bin\java.exe' ^>>>showCodeDetailsInExceptionMessages^>>>cp^> 'C:\Users\Adein\AppData\Roaming\Code\User\workspaceStorage\f0905537e094d15801a03918cc6de2\src\main\java\jdt\ui\IBF24CS291_9556be3b\bin' ^>>>cpd^>
Enter number of students: 2
    ... Student 1 ...
Enter student name: Lokesh
Enter USN: IBF24CS291
Enter number of subjects: 8
Enter marks for Subject 1: 77
Enter credits for Subject 1: 4
Enter marks for Subject 2: 64
Enter credits for Subject 2: 1
Enter marks for Subject 3: 70
Enter credits for Subject 3: 1
Enter marks for Subject 4: 81
Enter credits for Subject 4: 3
Enter marks for Subject 5: 79
Enter credits for Subject 5: 1
Enter marks for Subject 6: 81
Enter credits for Subject 6: 3
Enter marks for Subject 7: 88
Enter credits for Subject 7: 3
Enter marks for Subject 8: 94
Enter credits for Subject 8: 4

    ... Student 2 ...
Enter student name: Hrudhi
Enter USN: IMA24CS232
Enter number of subjects: 8
Enter marks for Subject 1: 76
Enter credits for Subject 1: 4
Enter marks for Subject 2: 89
Enter credits for Subject 2: 1
Enter marks for Subject 3: 67
Enter credits for Subject 3: 1
Enter marks for Subject 4: 97
Enter credits for Subject 4: 3
Enter marks for Subject 5: 56
Enter credits for Subject 5: 1
Enter marks for Subject 6: 88
Enter credits for Subject 6: 3
Enter marks for Subject 7: 92
Enter credits for Subject 7: 3
Enter marks for Subject 8: 92
Enter credits for Subject 8: 4

===== SGPA Report =====
SGPA of Lokesh (IBF24CS291) is: 8.8
SGPA of Hrudhi (IMA24CS232) is: 8.9
○ PS C:\IBF24CS291>
```

Activate Windows

### 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:

```
1 import java.util.Scanner;
2 class Books{
3     String name, author;
4     int price;
5     int numPages;
6     Books(String name, String author, int price, int numPages){
7         this.name = name;
8         this.author = author;
9         this.price = price;
10        this.numPages = numPages;
11    }
12
13    public String toString (){
14        String name, author, price, numPages;
15        name = "Book Name: " + this.name + "\n";
16        author = "Author Name: " + this.author + "\n";
17        price = "Price: " + this.price + "\n";
18        numPages = "Number of pages: " + this.numPages + "\n";
19        return name + author + price + numPages;
20    }
21 }
22
23 public class books {
24     Run|Debug
25     public static void main (String args[]){
26         Scanner s = new Scanner(System.in);
27         int n, price, numPages;
28         String name, author;
29         System.out.println("Enter the no. of books:");
30         n = s.nextInt();
31         s.nextLine();
32         Books b[];
33         b = new Books[n];
34         for (int i=0; i<n; i++){
35             System.out.println("Enter book name:");
36             name = s.nextLine();
37             System.out.println("Enter author's name:");
38             author = s.nextLine();
39             System.out.println("Enter price:");
40             price = s.nextInt();
41             System.out.println("Enter no. of pages in book:");
42             numPages = s.nextInt();
43             s.nextLine();
44             b[i] = new Books(name, author, price, numPages);
45             System.out.println(b[i].toString());
46         }
47         s.close();
48     }
49 }
```

## output:

```
PS C:\Users\DELL\OneDrive\Desktop\003> & 'C:\Program Files\Java\jdk-22\bin\java -jar' "C:\Users\DELL\Downloads\BookManager-0.0.1-SNAPSHOT.jar" >> "C:\Users\DELL\OneDrive\Desktop\003_23795900.txt" & BookManager
Enter the no. of books:
2
Enter book name:
It Ends with Us
Enter author's name:
Colleen Hoover
Enter price:
499
Enter no. of pages in book:
384
Book Details:
Book Name: It Ends with Us
Author Name: Colleen Hoover
Price: 499
Number of pages: 384

Enter book name:
Can We Be Strangers Again
Enter author's name:
Shrijeet Shandilya
Enter price:
499
Enter no. of pages in book:
232
Book Details:
Book Name: Can We Be Strangers Again
Author Name: Shrijeet Shandilya
Price: 499
Number of pages: 232
```

CHAT

- powershell
- Run books
- Run Studen...
- Run books
- Run books
- Run books
- Run BookM...

Ask about your code

AI responses may be inaccurate.

Generate Agent Instructions to onboard AI onto your codebase.

BookManager.java

Add context (F), extensions (E), commands (A), GPT-AI (V)

## 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 :

```
1 import java.util.Scanner;
2 abstract class shape{
3     int a,b;
4     shape(int c){
5         a=c;
6     }
7     shape(int c,int d){
8         a=c;b=d;
9     }
10    void printArea(){}
11 }
12 class rectangle extends shape{
13     rectangle(int c,int d){
14         super(c,d);
15     }
16     void printArea(){
17         System.out.println("Area of rectangle is: "+(double)a*b);
18     }
19 }
20 class triangle extends shape{
21     triangle(int c,int d){
22         super(c,d);
23     }
24     void printArea(){
25         System.out.println("Area of triangle is: "+(0.5f)*(a*b));
26     }
27 }
28 class circle extends shape{
29     circle(int c){
30         super(c);
31     }
32     void printArea(){
33         System.out.println("Area of circle is: "+(3.14f*a*a));
34     }
35 }
36 public class lab4 {
37     public static void main(String args[]){
38         Scanner sc = new Scanner(System.in);
39         int c,d;
40         System.out.println(x: "Enter Dimensions of rectangle(length and width):");
41         c=sc.nextInt();
42         d=sc.nextInt();
43         rectangle r = new rectangle(c,d);
44         r.printArea();
45         System.out.println(x: "Enter Dimensions of triangle(base and height):");
46         c=sc.nextInt();
47         d=sc.nextInt();
48         triangle t = new triangle(c,d);
49         t.printArea();
50         System.out.println(x: "Enter radius of circle:");
51         c=sc.nextInt();
52         circle cr =new circle(c);
53         cr.printArea();
54     }
55 }
```

## Output

## 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. Code:

code:

```
1 package Lab5;
2 import java.util.*;
3
4 class Account {
5     String name;
6     String accNo;
7     String accType;
8     double balance;
9     Account(String name, String accNo, String accType, double balance) {
10         this.name = name;
11         this.accNo = accNo;
12         this.accType = accType;
13         this.balance = balance;
14     }
15     void deposit(double amt) {
16         balance += amt;
17     }
18     void displayBalance() {
19         System.out.println("Balance: " + balance);
20     }
21 }
22
23 class SavAcct extends Account {
24     SavAcct(String name, String accNo, double balance) {
25         super(name, accNo, accType: "Savings", balance);
26     }
27     void computeInterest(double rate, int times) {
28         balance = balance * Math.pow(1 + rate / 100, times);
29     }
30     void withdraw(double amt) {
31         if (amt <= balance) balance -= amt;
32         else System.out.println("Insufficient balance");
33     }
34 }
```

```

36  class CurAcct extends Account {
37      double minBalance = 500;
38      double penalty = 50;
39      CurAcct(String name, String accNo, double balance) {
40          super(name, accNo, accType: "Current", balance);
41      }
42      void withdraw(double amt) {
43          if (amt <= balance) {
44              balance -= amt;
45              checkMinBalance();
46          } else System.out.println(x: "Insufficient balance");
47      }
48      void checkMinBalance() {
49          if (balance < minBalance) balance -= penalty;
50      }
51  }
52
53  public class Bank {
54      Run|Debug
55      public static void main(String[] args) {
56          Scanner sc = new Scanner(System.in);
57          System.out.print(s: "Enter number of customers: ");
58          int n = sc.nextInt();
59          sc.nextLine();
60          for (int i = 0; i < n; i++) {
61              System.out.println("\nCustomer " + (i + 1));
62              System.out.print(s: "Enter name: ");
63              String name = sc.nextLine();
64              System.out.print(s: "Enter account number: ");
65              String accNo = sc.nextLine();
66              System.out.print(s: "Enter account type (savings/current): ");
67              String type = sc.nextLine().toLowerCase();
68              System.out.print(s: "Enter initial balance: ");
69
70              double bal = sc.nextDouble();
71              if (type.equals(anObject: "savings")) {
72                  SavAcct s = new SavAcct(name, accNo, bal);
73                  int choice;
74                  do {
75                      System.out.println(x: "\n1. Deposit");
76                      System.out.println(x: "2. Withdraw");
77                      System.out.println(x: "3. Compute Interest");
78                      System.out.println(x: "4. Display Balance");
79                      System.out.println(x: "5. Exit");
80                      System.out.print(s: "Enter choice: ");
81                      choice = sc.nextInt();
82                      switch (choice) {
83                          case 1:
84                              System.out.print(s: "Enter deposit amount: ");
85                              s.deposit(sc.nextDouble());break;
86                          case 2:
87                              System.out.print(s: "Enter withdrawal amount: ");
88                              s.withdraw(sc.nextDouble());break;
89                          case 3:
90                              System.out.print(s: "Enter interest rate: ");
91                              double rate = sc.nextDouble();
92                              System.out.print(s: "Enter number of times interest applied: ");
93                              int times = sc.nextInt();
94                              s.computeInterest(rate, times);break;
95                          case 4:
96                              s.displayBalance();break;
97                      }
98                  } while (choice != 5);
99              }
100             else if (type.equals(anObject: "current")) {

```

```

98     else if (type.equals(anObject: "current")) {
99         CurAcct c = new CurAcct(name, accNo, bal);
100        int choice;
101        do {
102            System.out.println(x: "\n1. Deposit");
103            System.out.println(x: "2. Withdraw");
104            System.out.println(x: "3. Display Balance");
105            System.out.println(x: "4. Exit");
106            System.out.print(s: "Enter choice: ");
107            choice = sc.nextInt();
108            switch (choice) {
109                case 1:
110                    System.out.print(s: "Enter deposit amount: ");
111                    c.deposit(sc.nextDouble());break;
112                case 2:
113                    System.out.print(s: "Enter withdrawal amount: ");
114                    c.withdraw(sc.nextDouble());break;
115                case 3:
116                    c.displayBalance(); break;
117            }
118        } while (choice != 4);
119    }
120    sc.nextLine();
121}
122}
123

```

## Output:

```

PS C:\1BF24CS291> cd "c:\1BF24CS291\" ; if ($?) { javac Bank.java } ; if ($?) { java Bank }
● Enter number of customers: 1

Customer 1
Enter name: loki
Enter account number: 123
Enter account type (savings/current): savings
Enter initial balance: 123

1. Deposit
2. Withdraw
3. Compute Interest
4. Display Balance
5. Exit
Enter choice: 1
Enter deposit amount: 123

1. Deposit
2. Withdraw
3. Compute Interest
4. Display Balance
5. Exit
Enter choice: 3
Enter interest rate: 12
Enter number of times interest applied: 1

1. Deposit
2. Withdraw
3. Compute Interest
4. Display Balance
5. Exit
Enter choice: 4
Balance: 275.5200000000004

1. Deposit
2. Withdraw
3. Compute Interest
4. Display Balance
5. Exit
Enter choice: 5
○ PS C:\1BF24CS291>

```

## 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.

Code:

- Project
  - CIE
    - internals.java
    - student.java
  - SEE   externals.java
    -
  - main.java

```
1  package CIE;
2
3  public class internal extends student {
4      public int ciemarks[] = new int[5];
5  public  internal(String n,String u,int s,int cie[]){
6      {
7          super(n,u,s);
8      for(int i=0;i<5;i++){
9          ciemarks[i]=cie[i];
10     }
11   }
12 }
```

```
1  package CIE;
2  public class student{
3      String usn,name;int sem;
4      public student(String n,String u,int s){
5          name=n;usn=u;sem=s;
6      }
7  }
```

```

1 import SEE.externals;
2 import CIE.student;
3 import CIE.internal;
4 import java.util.Scanner;
5
6 public class final_marks {
7     Run | Debug
8     public static void main(String[] args) {
9         Scanner sc = new Scanner(System.in);
10        System.out.println("Enter no of students:");
11        int n = sc.nextInt();
12        sc.nextLine();
13        internal[] i = new internal[n];
14        externals[] e = new externals[n];
15
16        for (int j = 0; j < n; j++) {
17            System.out.println("Enter name of student " + (j + 1) + " :");
18            String Sname = sc.nextLine();
19            System.out.println("Enter USN of student:");
20            String USN = sc.nextLine();
21            System.out.println("Enter current semester of " + Sname + " :");
22            int semester = sc.nextInt();
23            int[] cieMarks = new int[5];
24            for (int k = 0; k < 5; k++){
25                System.out.println("Enter marks of sub " +(k+1)+ " :");
26                cieMarks[k] = sc.nextInt();
27            }
28            int[] seeMarks = new int[5];
29            for (int k = 0; k < 5; k++){
30                System.out.println("Enter marks of sub " +(k+1)+ " :");
31                seeMarks[k] = sc.nextInt();
32            }
33            sc.nextLine();
34            i[j] = new internal(Sname, USN, semester, cieMarks);
35            e[j] = new externals(Sname, USN, semester, seeMarks);
36        }
37        System.out.println("\n----- FINAL MARKS OF STUDENTS -----");
38        for (int f = 0; f < n; f++) {
39            for (int k = 0; k < 5; k++){
40                int finalMarks = (i[f].cieMarks[k]) + (e[f].seeMarks[k] / 2);
41                System.out.println("Student " + (f + 1) + " Subject " + (k + 1) + ": " + finalMarks);
42            }
43        }
44    }
45 }

```

## Output:

```

----- FINAL MARKS OF STUDENTS -----
Student 1 Subject 1: 11
Student 1 Subject 2: 30
Student 1 Subject 3: 77
Student 1 Subject 4: 75
Student 1 Subject 5: 70
PS C:\1BF24CS291> cd "c:\1BF24CS291\" ; if ($?) { javac final_marks.java } ; if ($?) { java final_marks }
Enter no of students:
1
Enter name of student 1 :
Lokesh
Enter USN of student:
1BF24CS291
Enter current semester of Lokesh :
3
Enter marks of sub 1 :
46
Enter marks of sub 2 :
43
Enter marks of sub 3 :
39
Enter marks of sub 4 :
35
Enter marks of sub 5 :
47
Enter marks of sub 1 :
95
Enter marks of sub 2 :
89
Enter marks of sub 3 :
68
Enter marks of sub 4 :
72
Enter marks of sub 5 :
90

----- FINAL MARKS OF STUDENTS -----
Student 1 Subject 1: 93
Student 1 Subject 2: 87
Student 1 Subject 3: 73
Student 1 Subject 4: 71
Student 1 Subject 5: 92

```

## 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 cases both father and son’s age and throws an exception if son’s age is >=father’s age.

Code:

```
1 package Lab7;
2
3 public class father{
4     public int father_age;
5     public father(int a) throws WrongAgeException{
6         if(a<0){
7             throw new WrongAgeException("Age cannot be negative");
8         }
9         father_age=a;
10    }
11    public void display(){
12        System.out.println("Father's age is :" +father_age);
13    }
14 }
```

```
1 import java.util.Scanner;
2 class WrongAgeException extends Exception {
3     public WrongAgeException(String message) {
4         super(message);
5     }
6 }
7
8 class Father {
9     protected int father_age;
10    public Father(int age) throws WrongAgeException {
11        if (age < 0) {
12            throw new WrongAgeException(message: "Father's age cannot be negative.");
13        }
14        father_age = age;
15    }
16    public void display() {
17        System.out.println("Father's age: " + father_age);
18    }
19 }
20
21 class Son extends Father {
22     private int son_age;
23     public Son(int fAge, int sAge) throws WrongAgeException {
24         super(fAge);
25         if (sAge < 0) {
26             throw new WrongAgeException(message: "Son's age cannot be negative.");
27         }
28         if (sAge >= father_age) {
29             throw new WrongAgeException(message: "Son's age cannot be greater than or equal to father's age.");
30         }
31         son_age = sAge;
32     }
33 }
```

```
33     }
34     public void display() {
35         System.out.println("Son's age: " + son_age);
36     }
37
38 public class lab7 {
39     Run | Debug
40     public static void main(String[] args) {
41         Scanner sc = new Scanner(System.in);
42         try {
43             System.out.print(s: "Enter Father's age: ");
44             int f_age = sc.nextInt();
45             Father f = new Father(f_age);
46             f.display();
47             System.out.print(s: "Enter Son's age: ");
48             int s_age = sc.nextInt();
49             Son s = new Son(f_age, s_age);
50             s.display();
51         } catch (WrongAgeException e) {
52             System.out.println("Error: " + e.getMessage());
53         }
54     }
55 }
```

## Output:

```
● PS C:\1BF24CS291\Lab7> java lab7
Enter Father's age: 10
Father's age: 10
Enter Son's age: 20
Error: Son's age cannot be greater than or equal to father's age.
● PS C:\1BF24CS291\Lab7> javac lab7.java
● PS C:\1BF24CS291\Lab7> java lab7
Enter Father's age: -1
Error: Father's age cannot be negative.
○ PS C:\1BF24CS291\Lab7> █
```

## 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:

```
1  class BMS implements Runnable{
2      public void run(){
3          try{
4              for(int i=0;i<5;i++){
5                  System.out.println(x: "BMS college of Engineering");
6                  Thread.sleep(millis: 10000);
7              }
8          }
9          catch(InterruptedException e){
10              System.out.println(x: "Interrupted Exception in BMS");
11          }
12      }
13  }
14  class CSE implements Runnable{
15      public void run(){
16          try{
17              for(int i=0;i<5;i++){
18                  System.out.println(x: "CSE");
19                  Thread.sleep(millis: 2000);
20              }
21          }
22          catch(InterruptedException e){
23              System.out.println(x: "Interrupted Exception in CSE");
24          }
25      }
26  }
27  class threads{
28      Run | Debug
29      public static void main(String[] args) {
30          Thread b = new Thread(new BMS());
31          Thread c = new Thread(new CSE());
32          b.start();
33          c.start();
34      }
35  }
```

## Output:

```
PS C:\1BF24CS291\LAB-8> cd "c:\1BF24CS291\LAB-8\" ; if ($?) { javac threads.java } ; if ($?) { java threads }
● BMS college of Engineering
CSE
CSE
CSE
CSE
CSE
BMS college of Engineering
○ PS C:\1BF24CS291\LAB-8>
```

