

**VISVESVARAYA TECHNOLOGICAL  
UNIVERSITY**

“JnanaSangama”, Belgaum -590014, Karnataka.



**LAB REPORT  
on**

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

*Submitted by*

**REKULAKUNTA SRAVANTHI (1BF24CS250)**

*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 **REKULAKUNTA SRAVANTHI (1BF24CS250)**, 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>Experiment Title</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	6-9
3	14/10/2025	Implement java program on Bank class	10-11
4	04/11/2025	Implement java program on Abstract class	12-13
5	04/11/2025	Implement java program on Bank management	14-17
6	18/11/2025	Implement java program on Packages	18-21
7	25/11/2025	Implement java program on Exceptions	21-23
8	09/12/2025	Implement java program on Multithreading	23-25

Github Link:

<https://github.com/sravanhirekulakunta/ooj1>

### **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 message stating that there are no real solutions.

Code:

```
import java.lang.Math;
import java.util.Scanner;
class quadratic{
    public static void main(String args[]){
        int a,b,c;
        float d;
        double r1,r2;
        Scanner in =new Scanner(System.in);
        System.out.println("Enter the value of a:");
        a=in.nextInt();
        System.out.println("Enter the value of b:");
        b=in.nextInt();
        System.out.println("Enter the value of c:");
        c=in.nextInt();
        if(a==0){
            System.out.println("Not a quadratic equation!");
        }
        else{
            d=(b*b-4*a*c);
            if(d==0){
                System.out.println("Roots are real and equal");
                r1=(-b)/(2*a));
                r2=r1;
                System.out.println("The roots are,"+r1);
            }else if(d>0){
                System.out.println("The roots are unequal");
                r1=(((-b)+(Math.sqrt(d)))/((2*a)));
                r2=(((-b)-(Math.sqrt(d)))/((2*a)));
                System.out.println("The Roots are:"+r1);
                System.out.println("and"+r2);
            }
        }
    }
}
```

```

        }else{
            System.out.println("Roots are imaginary");

        }

    in.close();

}

}

at quadratic.main(quadratic.java:10)
PS C:\1BF24CS250> cd "c:\1BF24CS250\" ; if ($?) { javac quadratic.java } ; if ($?) { java quadratic }
1
Enter the value of b:
2
Enter the value of c:
1
Roots are real and equal
The roots are,-1.0
PS C:\1BF24CS250> cd "c:\1BF24CS250\" ; if ($?) { javac quadratic.java } ; if ($?) { java quadratic }
Enter the value of a:
2
Enter the value of b:
3
Enter the value of c:
1
The roots are unequal
The Roots are:-0.5
and-1.0
PS C:\1BF24CS250> cd "c:\1BF24CS250\" ; if ($?) { javac quadratic.java } ; if ($?) { java quadratic }
Enter the value of a:
1
Enter the value of b:
2
Enter the value of c:
3
Roots are imaginary
PS C:\1BF24CS250> cd "c:\1BF24CS250\" ; if ($?) { javac quadratic.java } ; if ($?) { java quadratic }
Enter the value of a:
0
Enter the value of b:
1
Enter the value of c:
2
Not a quadratic equation!
PS C:\1BF24CS250>

```

## **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 Subject{
    int subjectMarks;
    int credits;
    int grade;
}
class Student{
    String name;
    String USN;
    double SGPA;
    Scanner s;
    Subject subject[];
    Student(){
        s=new Scanner(System.in);
        subject= new Subject[8];
        for(int i=0;i<8;i++){
            subject[i]=new Subject();
        }
    }
    void getstudentdetails(){
        System.out.print("Enter student name:");
        name=s.nextLine();
        System.out.print("Enter student USN:");
        USN=s.nextLine();

    }
    void getmarks(){
        for(int i=0;i<8;i++){
            System.out.println("Enter marks for subject"+(i+1)+":");
            subject[i].subjectMarks=s.nextInt();
            System.out.println("Enter credits for subject"+(i+1)+":");
            subject[i].credits=s.nextInt();
            subject[i].grade=(subject[i].subjectMarks/10)+1;
            if (subject[i].grade==11){
                subject[i].grade=10;
            }
        }
    }
}
```

```

        if (subject[i].grade<=4){
            subject[i].grade=0;
        }
    }
void computeSGPA(){
    int Score=0;
    int TotalCredits=0;
    for(int i=0;i<8;i++){
        Score+=(subject[i].grade*subject[i].credits);
        TotalCredits+=subject[i].credits;
    }
    SGPA=(double)Score/(double)TotalCredits;
}
void display(){
    System.out.println("Student details:");
    System.out.println("Name: "+name);
    System.out.println("USN: "+USN);
    System.out.println("SGPA: "+ SGPA);
}

public class studentSGPACal{
    public static void main(String args[]){
        Student student1=new Student();
        Student student2=new Student();
        student1.getstudentdetails();
        student1.getmarks();
        student1.computeSGPA();
        student1.display();
        student2.getstudentdetails();
        student2.getmarks();
        student2.computeSGPA();
        student2.display();
    }
}

```

```
PS C:\1BF24CS250> cd "c:\Users\Admin\Downloads\" ; if ($?) { javac studentSGPACal.java } ; if ($?) { java studentSGPACal

Enter student name:abcd
Enter student USN:1234
Enter marks for subject1:
78
Enter credits for subject1:
3
Enter marks for subject2:
81
Enter credits for subject2:
3
Enter marks for subject3:
82
Enter credits for subject3:
3
Enter marks for subject4:
87
Enter credits for subject4:
4
Enter marks for subject5:
83
Enter credits for subject5:
4
Enter marks for subject6:
90
Enter credits for subject6:
1
Enter marks for subject7:
92
Enter credits for subject7:
1
Enter marks for subject8:
95
Enter credits for subject8:
1
Student details:
Name: abcd
```

OUTPUT 5 DEBUG CONSOLE TERMINAL PORTS  Code +

```
USN: 1234
SGPA: 9.0
Enter student name:efgh
Enter student USN:1256
Enter marks for subject1:
89
Enter credits for subject1:
3
Enter marks for subject2:
87
Enter credits for subject2:
3
Enter marks for subject3:
90
Enter credits for subject3:
3
Enter marks for subject4:
92
Enter credits for subject4:
4
Enter marks for subject5:
87
Enter credits for subject5:
4
Enter marks for subject6:
90
Enter credits for subject6:
1
Enter marks for subject7:
91
Enter credits for subject7:
1
Enter marks for subject8:
89
Enter credits for subject8:
1
Student details:
Name: efgh
USN: 1256
SGPA: 9.45
```

### 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,author;
    int price,num_pages;
    Books(String name, String author, int price, int num_pages){

        this.name=name;
        this.author=author;
        this.price=price;
        this.num_pages=num_pages;
    }
    public String toString(){

        String name,author, price, num_pages;
        name="Book name:"+this.name+"\n";
        author="Author name: "+this.author+"\n";
        price="Price: "+this.price+"\n";
        num_pages="Number of Pages: "+this.num_pages+"\n";
        return name+author+price+num_pages;
    }
}
public class toStringDemo {
    public static void main(String args[]){
        Scanner s= new Scanner(System.in);
        int n, price, num_pages;
        String name, author;
        System.out.println("Enter the number of books: ");
        n=s.nextInt();
        Books b[];
        b=new Books[n];
        for(int i=0;i<n;i++){
            System.out.println("Enter the book "+(i+1)+" name: ");
            name=s.next();
        }
    }
}
```

```

        System.out.println("Enter the book "+(i+1)+" author: ");
        author=s.next();
        System.out.println("Enter the book "+(i+1)+" price: ");
        price=s.nextInt();
        System.out.println("Enter the number of pages in book "+(i+1)+" : ");
        num_pages=s.nextInt();
        b[i]=new Books(name,author,price,num_pages);
    }
    for(int i=0;i<n;i++){
        System.out.println(b[i]);
    }
}

```

OUTPUT 55 DEBUG CONSOLE TERMINAL PORTS

```

PS C:\1BF24CS250> cd "c:\Users\Admin\Desktop\OOJ lab 3\" ; if ($?) {
Enter the number of books:
2
Enter the book 1 name:
Ignite
Enter the book 1 author:
AbdulKalam
Enter the book 1 price:
250
Enter the number of pages in book 1 :
340
Enter the book 2 name:
Abcde
Enter the book 2 author:
xyz
Enter the book 2 price:
240
Enter the number of pages in book 2 :
330
Book name:Ignite
Author name: AbdulKalam
Price: 250
Number of Pages: 340

Book name:Abcde
Author name: xyz
Price: 240
Number of Pages: 330

PS C:\Users\Admin\Desktop\OOJ lab 3>

```

## 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;
class InputScanner{
    Scanner sc= new Scanner(System.in);
}
abstract class Shape extends InputScanner{
    int dim1, dim2;
    double Area;
    abstract void getinput();
    abstract void printArea();
}
class Rectangle extends Shape{
    void getinput(){
        System.out.println("Enter the dimensions of the Rectangle (length and breadth):");
        dim1=sc.nextInt();
        dim2=sc.nextInt();
    }
    void printArea(){
        Area=(double)dim1*dim2;
        System.out.println("Area of Rectangle: "+ Area);
    }
}
class Triangle extends Shape{
    void getinput(){
        System.out.println("Enter the dimensions of the Triangle (Base and Height):");
        dim1=sc.nextInt();
        dim2=sc.nextInt();
    }
    void printArea(){
        Area=(double)0.5*dim1*dim2;
        System.out.println("Area of Triangle: "+ Area);
    }
}
class Circle extends Shape{
    void getinput(){
        System.out.println("Enter the dimensions of the Circle (Radius):");
        dim1=sc.nextInt();
    }
    void printArea(){
```

```

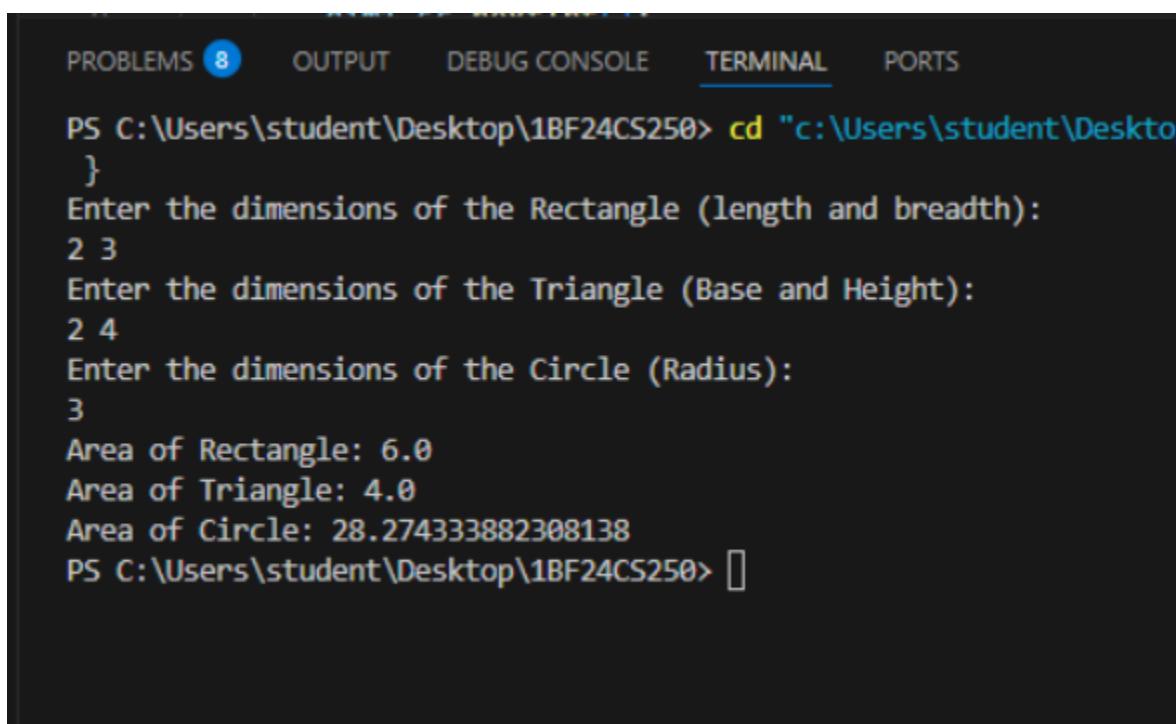
        Area=(double)Math.PI*dim1*dim1;
        System.out.println("Area of Circle: "+ Area);
    }
}

public class Abstractclassdemo {
    public static void main(String[] args) {
        Rectangle r= new Rectangle();
        r.getinput();
        Triangle t= new Triangle();
        t.getinput();
        Circle c= new Circle();
        c.getinput();

        r.printArea();
        t.printArea();
        c.printArea();

    }
}

```



The screenshot shows a terminal window with the following tabs at the top: PROBLEMS (8), OUTPUT, DEBUG CONSOLE, TERMINAL (selected), and PORTS.

The terminal output is as follows:

```

PS C:\Users\student\Desktop\1BF24CS250> cd "c:\Users\student\Desktop\1BF24CS250"
}
Enter the dimensions of the Rectangle (length and breadth):
2 3
Enter the dimensions of the Triangle (Base and Height):
2 4
Enter the dimensions of the Circle (Radius):
3
Area of Rectangle: 6.0
Area of Triangle: 4.0
Area of Circle: 28.274333882308138
PS C:\Users\student\Desktop\1BF24CS250> []

```

### **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 name,type;
    int accNo;
    double balance;

    void account(String n, int no, String t) {
        name = n;
        accNo = no;
        type = t;
        balance = 0.0;
    }

    void deposit(double amt) {
        balance += amt;
        System.out.println("Amount deposited.");
    }

    void display() {
        System.out.println("Customer: " + name);
        System.out.println("Account No: " + accNo);
        System.out.println("Type: " + type);
    }
}
```

```

        System.out.println("Balance: " + balance);
    }
}

class Savings extends Account {
    void interest() {
        double i = balance * 0.05;
        balance += i;
        System.out.println("Interest added: " + i);
    }

    void withdraw(double amt) {
        if (amt <= balance)
            balance -= amt;
        else
            System.out.println("Insufficient balance!");
    }
}

class Current extends Account {
    void withdraw(double amt) {
        balance -= amt;
        if (balance < 500) {
            balance -= 50;
            System.out.println("Service charge imposed.");
        }
    }
}

public class bankaccount{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Savings s = new Savings();
        Current c = new Current();

        System.out.print("Enter customer name for savings acc: ");
        String n1 = sc.next();
        System.out.print("Enter savings account number: ");
        int a1 = sc.nextInt();
        s.account(n1, a1, "Savings");

        System.out.print("Enter customer name for current account: ");
        String n2 = sc.next();
        System.out.print("Enter current account number: ");
        int a2 = sc.nextInt();
        c.account(n2, a2, "Current");
    }
}

```

```

int ch;
do {
    System.out.println("\n1.Deposit 2.Withdraw 3.Interest 4.Display 5.Exit");
    System.out.print("Enter choice: ");
    ch = sc.nextInt();

    switch (ch) {
        case 1 -> {
            System.out.print("Account type (saving/current): ");
            String t = sc.next();
            System.out.print("Amount: ");
            double d = sc.nextDouble();
            if (t.equalsIgnoreCase("saving"))
                s.deposit(d);
            else
                c.deposit(d);
        }
        case 2 -> {
            System.out.print("Account type (saving/current): ");
            String t = sc.next();
            System.out.print("Amount: ");
            double w = sc.nextDouble();
            if (t.equalsIgnoreCase("saving"))
                s.withdraw(w);
            else
                c.withdraw(w);
        }
        case 3 -> s.interest();
        case 4 -> {
            System.out.print("Account type (saving/current): ");
            String t = sc.next();
            if (t.equalsIgnoreCase("saving"))
                s.display();
            else
                c.display();
        }
        case 5 -> System.out.println("Thank you!");
        default -> System.out.println("Invalid choice!");
    }
} while (ch != 5);

sc.close();
}
}

```

OUTPUT 20 DEBUG CONSOLE TERMINAL PORTS

```
PS C:\1BF24CS250> cd "c:\1BF24CS250\" ; if ($?) { javac bankaccount.java
```

```
Enter customer name for savings acc: ram
```

```
Enter savings account number: 1234
```

```
Enter customer name for current account: ravi
```

```
Enter current account number: 1235
```

```
1.Deposit 2.Withdraw 3.Interest 4.Display 5.Exit
```

```
Enter choice: 1
```

```
Account type (saving/current): saving
```

```
Amount: 4500
```

```
Amount deposited.
```

```
1.Deposit 2.Withdraw 3.Interest 4.Display 5.Exit
```

```
Enter choice: 2
```

```
:1BF24CS250\first.class saving/current): saving
```

```
Amount: 1000
```

```
1.Deposit 2.Withdraw 3.Interest 4.Display 5.Exit
```

```
Enter choice: 3
```

```
Interest added: 175.0
```

```
1.Deposit 2.Withdraw 3.Interest 4.Display 5.Exit
```

```
Enter choice: 1
```

```
Account type (saving/current): current
```

```
Amount: 3000
```

```
Amount deposited.
```

```
1.Deposit 2.Withdraw 3.Interest 4.Display 5.Exit
```

```
Enter choice: 4
```

```
Account type (saving/current): saving
```

```
Customer: ram
```

```
Account No: 1234
```

```
Type: Savings
```

```
Balance: 3675.0
```

```
1.Deposit 2.Withdraw 3.Interest 4.Display 5.Exit
```

```
Enter choice: 5
```

```
Thank you!
```

```
PS C:\1BF24CS250>
```

### **Program 6**

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

Code:

CIE:

1)Student.java:

```
package project.CIE;
```

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

```
public class student {
```

```
    protected String usn;  
    protected String name;  
    protected int sem;
```

```
    public void inputStudentDetails() {  
        Scanner s = new Scanner(System.in);  
        System.out.print("Enter USN: ");  
        usn = s.nextLine();  
        System.out.print("Enter Name: ");  
        name = s.nextLine();  
        System.out.print("Enter Semester: ");  
        sem = s.nextInt();  
    }
```

```
    public void displayStudentDetails() {  
        System.out.println("USN: " + usn);  
        System.out.println("Name: " + name);  
        System.out.println("Semester: " + sem);  
    }  
}
```

## 2)internals.java:

```
package project.CIE;

import java.util.Scanner;

public class internals extends student {

    protected int marks[] = new int[5];

    public void inputCIEmarks() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter CIE marks of 5 subjects:");
        for (int i = 0; i < 5; i++) {
            System.out.print("Subject " + (i + 1) + ": ");
            marks[i] = s.nextInt();
        }
    }
}
```

SEE:

## externals.java:

```
package project.SEE;

import project.CIE.internals;
import java.util.Scanner;

public class externals extends internals {

    protected int smarks[]; // SEE marks
    protected int finalMarks[]; // Final CIE+SEE

    public externals() {
        smarks = new int[5];
        finalMarks = new int[5];
    }

    public void inputSEEmarks() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter SEE marks of 5 subjects:");
        for (int i = 0; i < 5; i++) {
            System.out.print("Subject " + (i + 1) + ": ");




```

```

        smarks[i] = s.nextInt();
    }

public void calculateFinalMarks() {
    for (int i = 0; i < 5; i++) {
        finalMarks[i] = (int) (smarks[i]/2)+ marks[i]; // SEE/2 + CIE
    }
}

public void displayFinalMarks() {
    displayStudentDetails();
    System.out.println("\nFinal Marks in 5 Subjects:");
    for (int i = 0; i < 5; i++) {
        System.out.println("Subject " + (i + 1) + ": " + finalMarks[i]);
    }
}

```

main.java:

```

package project;

import project.SEE.externals;

public class main {

    public static void main (String args[]) {

        externals s = new externals();

        System.out.println("Enter Student Details:");
        s.inputStudentDetails();

        System.out.println("\nEnter Internal Marks:");
        s.inputCIEmarks();

        System.out.println("\nEnter SEE Marks:");
        s.inputSEEmarks();

        s.calculateFinalMarks();

        System.out.println("\n---- FINAL RESULT ----");
        s.displayFinalMarks();
    }
}

```

The screenshot shows a terminal window from an IDE (likely Eclipse) displaying the output of a Java program. The terminal tab is selected at the top. The output shows the program's logic for calculating final marks based on internal and external marks.

```
a\AppData\Roaming\Code\User\workspaceStorage\4e18dfeb16fc5deaff8fb28ed3eec479\redhat.java\jdt_ws\JAVA PROGR  
AMS_49984ef5\bin" project.main "  
Enter Student Details:  
Enter USN: 1BF24CS250  
Enter Name: REKULAKUNTA SRAVANTHI  
Enter Semester: 03  
  
Enter Internal Marks:  
Enter CIE marks of 5 subjects:  
Subject 1: 45  
Subject 2: 46  
Subject 3: 47  
Subject 4: 48  
Subject 5: 49  
  
Enter SEE Marks:  
Enter SEE marks of 5 subjects:  
Subject 1: 90  
Subject 2: 92  
Subject 3: 94  
Subject 4: 96  
Subject 5: 98  
  
---- FINAL RESULT ----  
USN: 1BF24CS250  
Name: REKULAKUNTA SRAVANTHI  
Semester: 3  
  
Final Marks in 5 Subjects:  
Subject 1: 90  
Subject 2: 92  
Subject 3: 94  
Subject 4: 96  
Subject 5: 98
```

## 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 uses 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(){  
        super("Age Error!");
```

```

        }
    public WrongAge(String message){
        super(message);
    }

}

class InputScanner{
    Scanner s = new Scanner(System.in);
}

class Father extends InputScanner{
    protected int FatherAge;
    public Father() throws WrongAge{
        System.out.println("Enter father's Age: ");
        FatherAge=s.nextInt();
        if(FatherAge<0){
            throw new WrongAge("Age cannot be negative!");
        }
    }

    public void display(){
        System.out.println("Father's Age: "+ FatherAge);
    }
}

class Son extends Father{
    protected int SonAge;
    public Son() throws WrongAge{
        super();
        System.out.println("Enter son's age: ");
        SonAge= s.nextInt();
        if(SonAge<0){
            throw new WrongAge("Age cannot be negative!");
        }
        if(SonAge>=FatherAge){
            throw new WrongAge("Son's age cannot be greater than or equal to father's age!!!");
        }
    }

    public void display(){
        super.display();
        System.out.println("Son's Age: "+ SonAge);
    }
}

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

```

```

try {
    Son son= new Son();
    son.display();
} catch (WrongAge e) {
    System.out.println("Error: "+ e.getMessage());
} catch(Exception e){
    System.out.println("Unexpected Error: "+ e.getMessage());
}
}
}

```

The screenshot shows a terminal window with the following tabs: OUTPUT (29), DEBUG CONSOLE, TERMINAL (selected), and PORTS. The terminal output is as follows:

```

PS C:\1BF24CS250> cd "c:\1BF24CS250\" ; if ($?) { javac ageexception.java } ; if ($?) { java ageexception }
Enter father's Age:
32
Enter son's age:
-2
Error: Age cannot be negative!
PS C:\1BF24CS250> cd "c:\1BF24CS250\" ; if ($?) { javac ageexception.java } ; if ($?) { java ageexception }
Enter father's Age:
-1
Error: Age cannot be negative!
PS C:\1BF24CS250> cd "c:\1BF24CS250\" ; if ($?) { javac ageexception.java } ; if ($?) { java ageexception }
Enter father's Age:
28
Enter son's age:
31
Error: Son's age cannot be greater than or equal to father's age!!
PS C:\1BF24CS250> cd "c:\1BF24CS250\" ; if ($?) { javac ageexception.java } ; if ($?) { java ageexception }
Enter father's Age:
32
Enter son's age:
8
Father's Age: 32
Son's Age: 8
PS C:\1BF24CS250>

```

## 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 NewThread1 extends Thread{

    public void run(){

        for(int i=5; i>0; i--){
            System.out.println("BMS College of Engineering");
        }
    }
}

```

```
try{  
    Thread.sleep(10000);  
}catch (InterruptedException e) {  
    System.out.println("Interrupt!!!");  
}  
}  
}  
}  
}  
}  
class NewThread2 extends Thread{  
  
    public void run(){  
        for(int i=20; i>0; i--){  
            System.out.println("CSE");  
            try{  
                Thread.sleep(2000);  
            } catch (InterruptedException e) {  
                System.out.println("Interrupt!!!");  
            }  
        }  
    }  
}
```

```
public class Threadprogram1 {  
    public static void main (String args[]){  
        NewThread1 n1= new NewThread1();  
        NewThread2 n2= new NewThread2();  
        n1.start();  
        n2.start();  
    }  
}
```

OUTPUT 35

DEBUG CONSOLE

TERMINAL

PORTS

```
PS C:\1BF24CS250> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-4c4a7492ea2bda3b272100c1627\redhat.java\jdt_ws\1BF24CS250_9556bdbe'
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
PS C:\1BF24CS250>
```