

**VISVESVARAYA TECHNOLOGICAL
UNIVERSITY**

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT

on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Sreejeeta Roy (1BF24CS300)

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 **Sreejeeta Roy (1BF24CS300)**, 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	23.9.25	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 discriminant $b^2 - 4ac$ is negative, display a message stating that there are no real solutions.	5
2	14.10.25	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	7
3	28.10.25	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 <code>toString()</code> method that could display the complete details of the book. Develop a Java program to create n book objects	11
4	04.11.25	Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named <code>printArea()</code> . 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 <code>printArea()</code> that prints the area of the given shape.	13
5	4.11.25	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 d) Permit withdrawal and update the balance Check for the minimum balance, impose penalty if necessary and update the balance	15
6	18.11.25	Create a package CIE which has two classes - Student and Internals. The class Student has members like usn,	20

		<p>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.</p>	
7	25.11.25	<p>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.</p>	23
8	09.12.25	<p>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.</p>	25
9			
10			

GitHub Link:

<https://github.com/sreejeetaroybmsce/Java-.git>

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.*;
public class quadratic
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        int a,b,c;
        float r1=0;
        float r2=0;
        System.out.println("Sreejeeta Roy:1BF24CS300");
        System.out.println("Enter value of a");
        a=sc.nextInt();
        System.out.println("Enter the value of b");
        b=sc.nextInt();
        System.out.println("Enter the value of c");
        c=sc.nextInt();
        if(a==0)
            System.out.println("Not Quadratic Equation");
        else
        {
            float d=b*b-4*a*c;
            if(d==0)
            {
                r1=(-b)/(2*a);
                System.out.println("Roots are real and equal");
                System.out.println("Root 1 and Root 2: "+r1);
            }
            else if(d>0)
            {
                r1=(((-b)+((float)Math.sqrt(d)))/(2*a));
                r2=(((-b)-((float)Math.sqrt(d)))/(2*a));
                System.out.println("Roots are real and unequal");
                System.out.println("First root is: "+r1);
                System.out.println("Second root is: "+r2);
            }
        }
    }
}
```

```

        else
    {
        r1=(-b)/(2*a);
        r2=(float)Math.sqrt(-d)/(2*a);
        System.out.println("Roots are imaginary");
        System.out.println("The first root is: "+r1+"+"+r2+"i");
        System.out.println("The second root is: "+r1+"-"+r2+"i");
    }

}

}

```

Output:

```

PS C:\sreejeeta 1bf24cs300> javac quadratic.java
PS C:\sreejeeta 1bf24cs300> java quadratic
Sreejeeta Roy:1BF24CS300
Enter value of a
0
Enter the value of b
1
Enter the value od c
2
Not Quadratic Equation
PS C:\sreejeeta 1bf24cs300> java quadratic
Sreejeeta Roy:1BF24CS300
Enter value of a
1
Enter the value of b
4
Enter the value od c
5
Roots are imaginary
The first root is: -2.0+1.0i
The second root is: -2.0-1.0i
PS C:\sreejeeta 1bf24cs300> java quadratic
Sreejeeta Roy:1BF24CS300
Enter value of a
1
Enter the value of b
5
Enter the value od c
4
Roots are real and unequal
First root is: -1.0
Second root is: -4.0
PS C:\sreejeeta 1bf24cs300>

```

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.*;
public class student
{
    Scanner sc=new Scanner(System.in);
    String sub;
    int credits;
    String usn;
    int grade;
    String name;
    double sgpa;int count;
    void getStudentDetails()
    {
        System.out.println("Enter name");
        this.name=sc.nextLine();
        System.out.println("Enter usn");
        usn=sc.next();
        count++;
        getmarks();
    }
    void getmarks()
    {
        int arr[]=new int[8];
        int credit[]=new int[8];
        System.out.println("Enter marks and credit for 8 subjects");
        for(int i=0;i<8;i++)
        {
            System.out.println("Subject "+(i+1)+":");
            System.out.println("Enter marks");
            arr[i]=sc.nextInt();
            System.out.println("Enter credit");
            credit[i]=sc.nextInt();
        }
        int grade[]=new int[8];
        for(int i=0;i<8;i++)
        {
            grade[i]=(arr[i]/10)+1;
            if(grade[i]==11)
```

```

grade[i]=10;
if(grade[i]<=4)
grade[i]=0;
}
computesgpa(grade,credit);
}
void computesgpa(int[] grades,int[] credits)
{
int score=0;int total=0;
for(int i=0;i<8;i++)
{
score=score+(grades[i]*credits[i]);
total=total+credits[i];
}
double ans=(double)score/(double)total;
System.out.println("For student "+count+":");
System.out.println("Name is:"+name);
System.out.println("USN: "+usn);
System.out.println("sgpa: "+ans);

}
public static void main(String[] args)
{
student student1=new student();
student student2=new student();
System.out.println("Calculating for student 1");
student1.getStudentDetails();
System.out.println("Calculating for Student 2");
student2.getStudentDetails();

}

```

Output:

```
PS C:\sreejeeta 1bf24cs300> cd "c:\sreejeeta 1bf24cs300\" ; if ($?) { javac student.java } ; if ($?) { java student }
Calculating for student 1
Enter name
sreejeeta
Enter usn
1bf24cs300
Enter marks and credit for 8 subjects
Subject 1:
Enter marks
100
Enter credit
2
Subject 2:
Enter marks
89
Enter credit
3
Subject 3:
Enter marks
58
Enter credit
3
Subject 4:
Enter marks
100
Enter credit
4
Subject 5:
Enter marks
100
Enter credit
3
Subject 6:
Enter marks
90
Enter credit
2
Subject 7:
Enter marks
98
Enter credit
4
Subject 8:
Enter marks
91
Enter credit
5
For student 1:
Name is:sreejeeta
USN: 1bf24cs300
sgpa: 9.423076923076923
Calculating for Student 2
Enter name
rahul
Enter usn
```

PROBLEMS 13 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
sgpa: 9.423076923076923
```

```
Calculating for Student 2
```

```
Enter name
```

```
rahul
```

```
Enter usn
```

```
1bf24cs111
```

```
Enter marks and credit for 8 subjects
```

```
Subject 1:
```

```
Enter marks
```

```
100
```

```
Enter credit
```

```
3
```

```
Subject 2:
```

```
Enter marks
```

```
90
```

```
Enter credit
```

```
2
```

```
Subject 3:
```

```
Enter marks
```

```
89
```

```
Enter credit
```

```
4
```

```
Subject 4:
```

```
Enter marks
```

```
78
```

```
Enter credit
```

```
4
```

```
Subject 5:
```

```
Enter marks
```

```
67
```

```
Enter credit
```

```
3
```

```
Subject 6:
```

```
Enter marks
```

```
78
```

```
Enter credit
```

```
3
```

```
Subject 7:
```

```
Enter marks
```

```
89
```

```
Enter credit
```

```
4
```

```
Subject 8:
```

```
Enter marks
```

```
100
```

```
Enter credit
```

```
4
```

```
For student 1:
```

```
Name is:rahul
```

```
USN: 1bf24cs111
```

```
sgpa: 8.851851851851851
```

```
PS C:\sreejeeta 1bf24cs300> []
```

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.*;
public class book
{
    String name,author;
    int price,num_pages;
    book(String n,String a,int p,int page)
    {
        name=n;
        author=a;
        price=p;
        num_pages=page;
    }
    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.num_pages + "\n";

        return name + author + price + numPages;
    }
}
public class Lp3 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n, price, pages;
        String name, author;
        System.out.println("enter no of books");
        n = sc.nextInt();
        sc.nextLine();
        book b[] = new book[n];
        for(int i = 0 ;i<n;i++){
            System.out.println("enter name");
            name = sc.nextLine();
            System.out.println("enter author");
            author = sc.nextLine();
```

```

        System.out.println("enter price");
        price = sc.nextInt();
        System.out.println("enter pages");
        pages = sc.nextInt();
        b[i] = new book(name,author,price,pages);
        sc.nextLine();
    }
    for(int i = 0;i<n;i++)
        System.out.println(b[i]);
}

```

Output:

```

PS C:\Users\Admin\Documents\1bf24cs300> javac Lp3.java
PS C:\Users\Admin\Documents\1bf24cs300> java Lp3
enter no of books
2
Enter details for book1
enter name
hello
enter author
ravi
enter price
100
enter pages
20
Enter details for book2
enter name
air
enter author
riya
enter price
200
enter pages
100
Book name: hello
Author name: ravi
Price: 100
Number of pages: 20

Book name: air
Author name: riya
Price: 200
Number of pages: 100

```

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 a, b;
    abstract void printArea();
}

class Rectangle extends Shape {
    Rectangle()
    {
        System.out.println("Enter the dimensions of the rectangle (length and breadth):");
        Scanner s = new Scanner(System.in);
        a = s.nextInt();
        b = s.nextInt();
    }

    void printArea() {
        double area = a * b;
        System.out.println("Area of Rectangle = " + area);
    }
}

class Triangle extends Shape {
    Triangle()
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the dimensions of the triangle (base and height):");
        a = s.nextInt();
        b = s.nextInt();
    }

    void printArea() {
        double area = 0.5 * a * b;
        System.out.println("Area of Triangle = " + area);
    }
}
```

```

}

class Circle extends Shape {
    Circle()
    {   Scanner s = new Scanner(System.in);
        System.out.println("Enter radius of circle:");
        a = s.nextInt();
    }

    void printArea() {
        double area = 3.142 * a * a;
        System.out.println("Area of Circle = " + area);
    }
}

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

```

Output:

```

PS C:\Users\Admin\Documents\1bf24cs300> javac MainShape.java
PS C:\Users\Admin\Documents\1bf24cs300> java MainShape
Error: Could not find or load main class MainShape
Caused by: java.lang.ClassNotFoundException: MainShape
PS C:\Users\Admin\Documents\1bf24cs300> java MainShape
Enter the dimensions of the rectangle (length and breadth):
10
20
Enter the dimensions of the triangle (base and height):
10
20
Enter radius of circle:
10
Area of Rectangle = 200.0
Area of Triangle = 100.0
Area of Circle = 314.2
PS C:\Users\Admin\Documents\1bf24cs300>

```

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.nextLine();
        System.out.print("Enter account Number: ");
        accountNumber = s.nextInt();
        System.out.print("Enter type of account (saving/current): ");
        accountType = s.nextLine();
        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;
    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 bank {
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("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;
    }
}

```

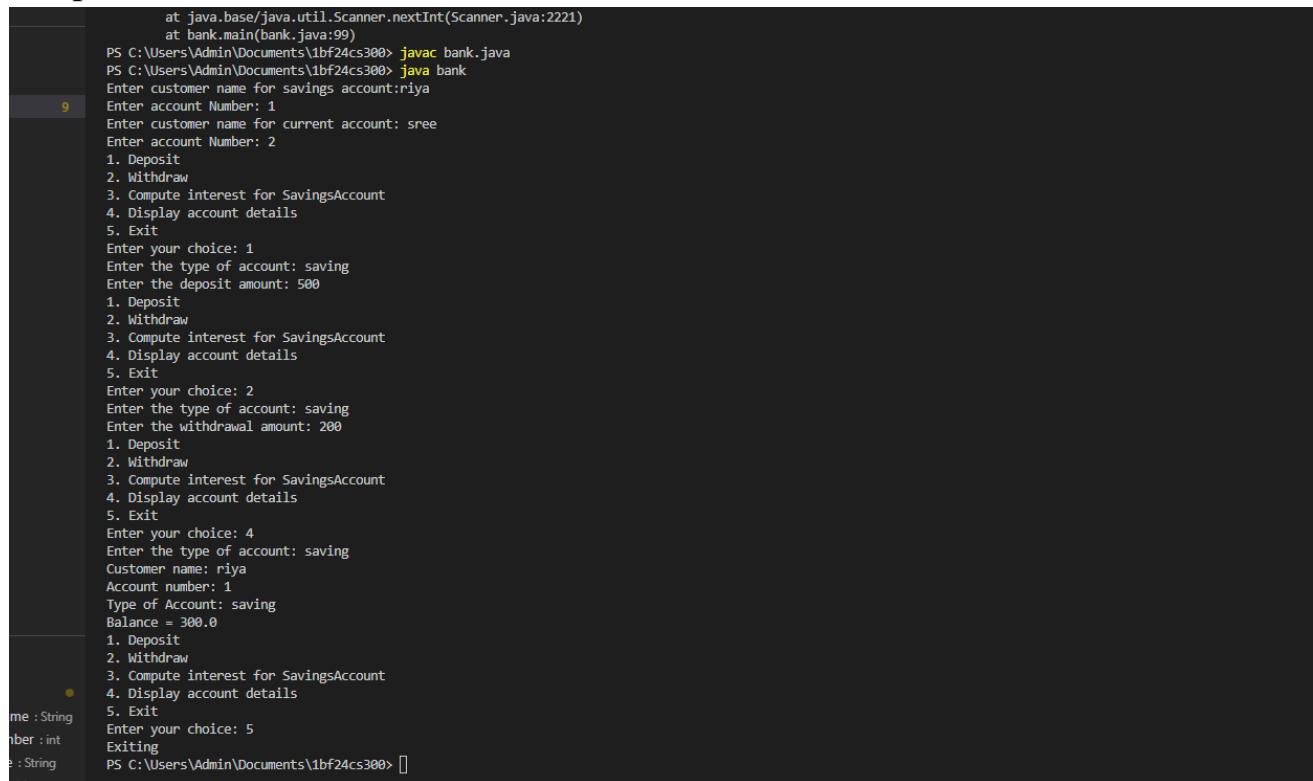
```

case 5:
    System.out.println("Exiting");
    break;

default:
    System.out.println("Invalid choice!");
}
} while (choice != 5);
}
}

```

Output:



```

at java.base/java.util.Scanner.nextInt(Scanner.java:2221)
at bank.main(bank.java:99)
PS C:\Users\Admin\Documents\1bf24cs300> javac bank.java
PS C:\Users\Admin\Documents\1bf24cs300> java bank
Enter customer name for savings account:riya
Enter account Number: 1
Enter customer name for current account: sree
Enter account Number: 2
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: 500
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
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: riya
Account number: 1
Type of Account: saving
Balance = 300.0
1. Deposit
2. Withdraw
3. Compute interest for SavingsAccount
4. Display account details
5. Exit
Enter your choice: 5
Exiting
PS C:\Users\Admin\Documents\1bf24cs300> []

```

```

public class Bank {
    String name;
    int number;
    String type;
    double balance;
    public void withdraw() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the withdrawal amount: ");
        int amount = sc.nextInt();
        if (amount > balance) {
            System.out.println("Insufficient balance");
        } else {
            balance -= amount;
            System.out.println("Withdrawal successful. New balance: " + balance);
        }
    }
    public void deposit() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the deposit amount: ");
        int amount = sc.nextInt();
        balance += amount;
        System.out.println("Deposit successful. New balance: " + balance);
    }
    public void computeInterest() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the interest rate: ");
        double rate = sc.nextDouble();
        double interest = balance * rate / 100;
        System.out.println("Interest calculated: " + interest);
    }
    public void displayDetails() {
        System.out.println("Customer name: " + name);
        System.out.println("Account number: " + number);
        System.out.println("Type of Account: " + type);
        System.out.println("Balance = " + balance);
    }
}

```

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 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:

```
package 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);
    }
}

package 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 for 5 subjects: ");
        for(int i = 0; i < 5; i++) {
            System.out.print("CIE Mark " + (i+1) + ": ");
            marks[i] = s.nextInt();
        }
    }
}
```

```

        }
    }
}

package SEE;
import CIE.Internals;
import java.util.Scanner;

public class Externals extends Internals {
    protected int seeMarks[];
    protected int finalMarks[];

    public Externals() {
        seeMarks = new int[5];
        finalMarks = new int[5];
    }

    public void inputSEEmarks() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter SEE marks for 5 subjects: ");
        for (int i = 0; i < 5; i++) {
            System.out.print("SEE Mark " + (i + 1) + ": ");
            seeMarks[i] = s.nextInt();
        }
    }

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

    public void displayFinalMarks() {
        displayStudentDetails();
        System.out.println("Final Marks (CIE + SEE/2): ");
        for (int i = 0; i < 5; i++) {
            System.out.println("Subject " + (i+1) + ": " + finalMarks[i]);
        }
    }
}

import SEE.Externals;

```

```

class Main {
    public static void main(String args[]) {
        java.util.Scanner s = new java.util.Scanner(System.in);

        System.out.print("Enter number of students: ");
        int n = s.nextInt();

        Externals[] st = new Externals[n];

        for (int i = 0; i < n; i++) {
            System.out.println("\n--- Student " + (i+1) + " ---");
            st[i] = new Externals();
            st[i].inputStudentDetails();
            st[i].inputCIEmarks();
            st[i].inputSEEmarks();
            st[i].calculateFinalMarks();
        }

        System.out.println("\n===== FINAL MARKS REPORT =====");
        for (int i = 0; i < n; i++) {
            st[i].displayFinalMarks();
        }
    }
}

```

Output:

```

PS C:\Users\SREEJEEATA\OneDrive\Desktop\PRGS\1BF24CS300> java .\LAB6.class
PS C:\Users\SREEJEEATA\OneDrive\Desktop\PRGS\1BF24CS300> java LAB6
Enter number of students: 2

--- Student 1 ---
Enter USN: 1bf24cs300
Enter Name: sreejeeta
Enter Semester: 2
Enter CIE marks for 5 subjects:
CIE Mark 1: 12
CIE Mark 2: 14
CIE Mark 3: 16
CIE Mark 4: 18
CIE Mark 5: 90
Enter SEE marks for 5 subjects:
SEE Mark 1: 34
SEE Mark 2: 23
SEE Mark 3: 12
SEE Mark 4: 67
SEE Mark 5: 56

--- Student 2 ---
Enter USN: 1bf24cs400
Enter Name: piya
Enter Semester: 2
Enter CIE marks for 5 subjects:
CIE Mark 1: 13
CIE Mark 2: 15
CIE Mark 3: 18
CIE Mark 4: 19
CIE Mark 5: 13
Enter SEE marks for 5 subjects:
SEE Mark 1: 34
SEE Mark 2: 56
SEE Mark 3: 67
SEE Mark 4: 78
SEE Mark 5: 79

===== FINAL MARKS REPORT =====
USN: 1bf24cs300
Name: sreejeeta

```

```
===== FINAL MARKS REPORT =====
```

```
USN: 1bf24cs300
```

```
Name: sreejeeta
```

```
Semester: 2
```

```
Final Marks (CIE + SEE/2):
```

```
Subject 1: 29
```

```
Subject 2: 25
```

```
Subject 3: 22
```

```
Subject 4: 51
```

```
Subject 5: 118
```

```
USN: 1bf24cs400
```

```
Name: piya
```

```
Semester: 2
```

```
Final Marks (CIE + SEE/2):
```

```
Subject 1: 30
```

```
Subject 2: 43
```

```
Subject 3: 51
```

```
Subject 4: 58
```

```
Subject 5: 52
```

```
PS C:\Users\SREEJEETA\OneDrive\Desktop\PRGS\1BF24CS300> █
```

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.Scanner;

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

class Father {
    int fatherAge;

    Father(int age) throws WrongAge {
        if (age < 0) {
            throw new WrongAge("Father's age cannot be negative");
        }
        fatherAge = age;
    }
}

class Son extends Father {
    int sonAge;

    Son(int fatherAge, int sonAge) throws WrongAge {
        super(fatherAge);

        if (sonAge < 0) {
            throw new WrongAge("Son's age cannot be negative");
        }

        if (sonAge >= fatherAge) {
            throw new WrongAge("Son's age must be less than father's age");
        }

        this.sonAge = sonAge;
    }
}

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

```

Scanner sc = new Scanner(System.in);

System.out.print("Enter father's age: ");
int fAge = sc.nextInt();

System.out.print("Enter son's age: ");
int sAge = sc.nextInt();

try {
    Son obj = new Son(fAge, sAge);
    System.out.println("Son's age is: " + obj.sonAge);
    System.out.println("Father's age is: " + obj.fatherAge);
} catch (WrongAge e) {
    System.out.println("Error: " + e.getMessage());
}

sc.close();
}
}

```

Output:

```

PS C:\1bf24cs300> cd "c:\1bf24cs300\" ; if ($?) { javac Son.java } ; if ($?) { java Son }
Enter father's age: 40
Enter son's age: 20
Son's age is: 20
Father's age is: 40
PS C:\1bf24cs300> cd "c:\1bf24cs300\" ; if ($?) { javac Son.java } ; if ($?) { java Son }
Enter father's age: 12
Enter son's age: 18
Error: Son's age must be less than father's age
PS C:\1bf24cs300> []

```

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 CollegeThread extends Thread {  
    public void run() {  
        try {  
            for(int i=0;i<4;i++) {  
                System.out.println("BMS College of Engineering");  
                Thread.sleep(10000); // 10 seconds  
            }  
        } catch (InterruptedException e) {  
            System.out.println("CollegeThread interrupted");  
        }  
    }  
}  
  
class CSEThread extends Thread {  
    public void run() {  
        try {  
            for(int i=0;i<4;i++){  
                System.out.println("CSE");  
                Thread.sleep(2000); // 2 seconds  
            }  
        } catch (InterruptedException e) {  
            System.out.println("CSEThread interrupted");  
        }  
    }  
}  
  
public class threaddemo {  
    public static void main(String[] args) {  
        CollegeThread t1 = new CollegeThread();  
        CSEThread t2 = new CSEThread();  
  
        t1.start();  
        t2.start();  
    }  
}
```

Output: