

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT On

Object Oriented Java Programming (23CS3PCOOJ)

Submitted by

Sagarmatha Khatri (1BF24CS261)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING



B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
August-December 2025

B.M.S. COLLEGE OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING



DECLARATION

This is to certify that the Lab work entitled “Object Oriented Java Programming (23CS3PCOOJ)” carried out by **Sagarmatha Khatri (1BF24CS261)**, 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	Title	Page no.
1	23/09/2025	Program 1: Implementation of Quadratic Equation	4-5
2	23/09/2025	Program 2: Implement SGPA Calculator	6-8
3	23/09/2025	Program 3: Implement toString() Function	9-11
4	23/09/2025	Program 4: Implement Abstract Class	12-14
5	23/09/2025	Program 5: Implement Bank Account Operations	15-22
6	23/09/2025	Program 6: Implement Packages	23-28
7	23/09/2025	Program 7: Implement Exception Handling	29-31
8	23/09/2025	Program 8: Implement Threads	31-33

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 discriminate $b^2 - 4ac$ is negative, display a message stating that there are no real solutions.

//Code:

```
import java.util.Scanner;
import java.lang.Math;
class lab1 {
    public static void main(String args[]){
        Scanner sc=new Scanner (System.in);
        System.out.println ("enter the coefficients of a b and c:");
        int a= sc.nextInt ();
        int b= sc.nextInt ();
        int c= sc.nextInt ();
        if (a==0){
            System.out.println("not a quadratic equation");
        }
        else{
            double r1, r2;
            int d= b*b-4*a*c;
            if (d==0){
                System.out.println("Roots are real and equal");
                r1 = (-b)/(2*a);
                System.out.println("r1=r2="+r1);
            }else if(d>0){
                r1 = ((-b) + (Math.sqrt(d)))/(double)(2*a);
                r2 = ((-b) - (Math.sqrt(d)))/(double)(2*a);
                System.out.println("Roots are distinct");
                System.out.println("r1="+r1);
                System.out.println("r2="+r2);
            }else if (d<0){
                System.out.println("roots are imaginary");
                r1 = (-b)/(2*a);
                r2 = Math.sqrt(-d)/(2*a);
                System.out.println("r1="+r1);
                System.out.println("r2="+r2);
            }
        }
    }
}
```

```

    }
  }
}

```

//Output

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
r1=r2=0.0
PS C:\Users\Admin\Desktop\1bf24cs261> cd "c:\Users\Admin\Desktop\1bf24cs261\" ; if ($?) { javac lab1.java } ; if ($?) { java lab1 }
enter the coffecients of a b and c:
0
11
1
not a quadratic equation
PS C:\Users\Admin\Desktop\1bf24cs261> cd "c:\Users\Admin\Desktop\1bf24cs261\" ; if ($?) { javac lab1.java } ; if ($?) { java lab1 }
enter the coffecients of a b and c:
cd "c:\Users\Admin\Desktop\1bf24cs261\" ; if ($?) { javac lab1.java } ; if ($?) { java lab1 }
Exception in thread "main" java.util.InputMismatchException
    at java.base/java.util.Scanner.next(Scanner.java:1602)
    at java.base/java.util.Scanner.nextInt(Scanner.java:2267)
    at java.base/java.util.Scanner.nextInt(Scanner.java:2221)
    at lab1.main(lab1.java:7)
PS C:\Users\Admin\Desktop\1bf24cs261> cd "c:\Users\Admin\Desktop\1bf24cs261\" ; if ($?) { javac lab1.java } ; if ($?) { java lab1 }
0
2
3
not a quadratic equation
PS C:\Users\Admin\Desktop\1bf24cs261> cd "c:\Users\Admin\Desktop\1bf24cs261\" ; if ($?) { javac lab1.java } ; if ($?) { java lab1 }
enter the coffecients of a b and c:
0
0
not a quadratic equation
PS C:\Users\Admin\Desktop\1bf24cs261> cd "c:\Users\Admin\Desktop\1bf24cs261\" ; if ($?) { javac lab1.java } ; if ($?) { java lab1 }
enter the coffecients of a b and c:
0
0
Roots are real and equal
r1=r2=0.0
PS C:\Users\Admin\Desktop\1bf24cs261> cd "c:\Users\Admin\Desktop\1bf24cs261\" ; if ($?) { javac tempCodeRunnerFile.java } ; if ($?) { java tempCodeRunnerFile }
tempCodeRunnerFile.java:2: error: unnamed classes are a preview feature and are disabled by default.
    int a= sc.nextInt ();
    (use --enable-preview to enable unnamed classes)
1 error
PS C:\Users\Admin\Desktop\1bf24cs261> cd "c:\Users\Admin\Desktop\1bf24cs261\" ; if ($?) { javac lab1.java } ; if ($?) { java lab1 }
enter the coffecients of a b and c:
1
4
2
Roots are distinct
r1=-0.5857864376269049
r2=-3.414213562373095
PS C:\Users\Admin\Desktop\1bf24cs261> cd "c:\Users\Admin\Desktop\1bf24cs261\" ; if ($?) { javac lab1.java } ; if ($?) { java lab1 }
enter the coffecients of a b and c:
1
2
3
roots are imaginary
r1=-1.0
r2=1.4142135623730951
PS C:\Users\Admin\Desktop\1bf24cs261>

```

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, credits, grade;
}

class lab2 {
    String name, usn;
    double SGPA;
    Scanner sc = new Scanner(System.in);
    Subject[] subject;

    lab2() {
        subject = new Subject[8];
        for (int i = 0; i < 8; i++) {
            subject[i] = new Subject();
        }
    }

    void getStudentDetails() {
        System.out.print("Enter name: ");
        name = sc.nextLine();
        System.out.print("Enter USN: ");
        usn = sc.nextLine();
    }

    void getMarks() {
        for (int i = 0; i < 8; i++) {
            System.out.print("Enter Marks for Subject " + (i + 1) + ": ");
```

```

        subject[i].subjectMarks = sc.nextInt();
        System.out.print("Enter Credits for Subject " + (i + 1) + ": ");
        subject[i].credits = sc.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;
    }
    sc.nextLine(); // consume leftover newline
}

void calculateSGPA() {
    int effectiveScore = 0, totalCredits = 0;
    for (int i = 0; i < 8; i++) {
        effectiveScore += (subject[i].grade * subject[i].credits);
        totalCredits += subject[i].credits;
    }
    SGPA = (double) effectiveScore / (double) totalCredits;
}

void displayStudentInfo() {
    System.out.println("\nStudent Details:");
    System.out.println("Name: " + name);
    System.out.println("USN: " + usn);
    System.out.println("SGPA: " + SGPA);
}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    lab2[] students = new lab2[2]; // for 2 students

    for (int i = 0; i < 2; i++) {
        System.out.println("\nEnter details for Student " + (i + 1) + ": ");
        students[i] = new lab2();
        students[i].getStudentDetails();
        students[i].getMarks();
        students[i].calculateSGPA();
        System.out.println("Results: ");
    }
}

```

```
        students[i].displayStudentInfo();
    }

}
```

//Output:

```
Enter name: Sagarmatha Khatri
Enter name: Sagarmatha Khatri
Enter name: Sagarmatha Khatri
Enter USN: 1BF24CS261
Enter USN: 1BF24CS261
Enter Marks for Subject 1: 90
Enter Credits for Subject 1: 4
Enter Marks for Subject 2: 83
Enter Credits for Subject 2: 4
Enter Marks for Subject 3: 77
Enter Credits for Subject 3: 3
Enter Marks for Subject 2: 83
Enter Credits for Subject 2: 4
Enter Marks for Subject 3: 77
Enter Credits for Subject 3: 3
Enter Credits for Subject 2: 4
Enter Marks for Subject 3: 77
Enter Credits for Subject 3: 3
Enter Marks for Subject 4: 96
Enter Marks for Subject 3: 77
Enter Credits for Subject 3: 3
Enter Marks for Subject 4: 96
Enter Credits for Subject 3: 3
Enter Marks for Subject 4: 96
Enter Credits for Subject 4: 3
Enter Marks for Subject 5: 91
Enter Credits for Subject 5: 3
Enter Marks for Subject 6: 99
Enter Credits for Subject 6: 1
Enter Marks for Subject 7: 98
Enter Credits for Subject 7: 2
Enter Marks for Subject 8: 97
Enter Credits for Subject 8: 1
SGPA: 9.523809523809524
PS C:\Users\Admin\Desktop\1bf24cs261>
```


Program 3:

Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString() method that could display the complete details of the book. Develop a Java program to create n book objects.

//Code:

```
import java.util.Scanner;

class Books {
    String name;
    String author;
    int price;
    int 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 = "Book name: " + this.name + "\n";
        String author = "Author name: " + this.author + "\n";
        String price = "Price: " + this.price + "\n";
        String num_pages = "Number of pages: " + this.num_pages + "\n";
        return name + author + price + num_pages;
    }
}

public class lab3
{
    public static void main(String args[])
```

```

{
    Scanner in = new Scanner(System.in);
    String name,author;
    int price, num_pages;

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

    Books[] b = new Books[n];

    for (int i = 0; i < n; i++)
    {
        System.out.println("\n\nEnter details for book " + (i + 1) + ":");

        System.out.print("Name: ");
        name = in.next();

        System.out.print("Author: ");
        author = in.next();

        System.out.print("Price: ");
        price = in.nextInt();

        System.out.print("Number of pages: ");
        num_pages= in.nextInt();

        b[i] = new Books(name, author, price, num_pages);
    }

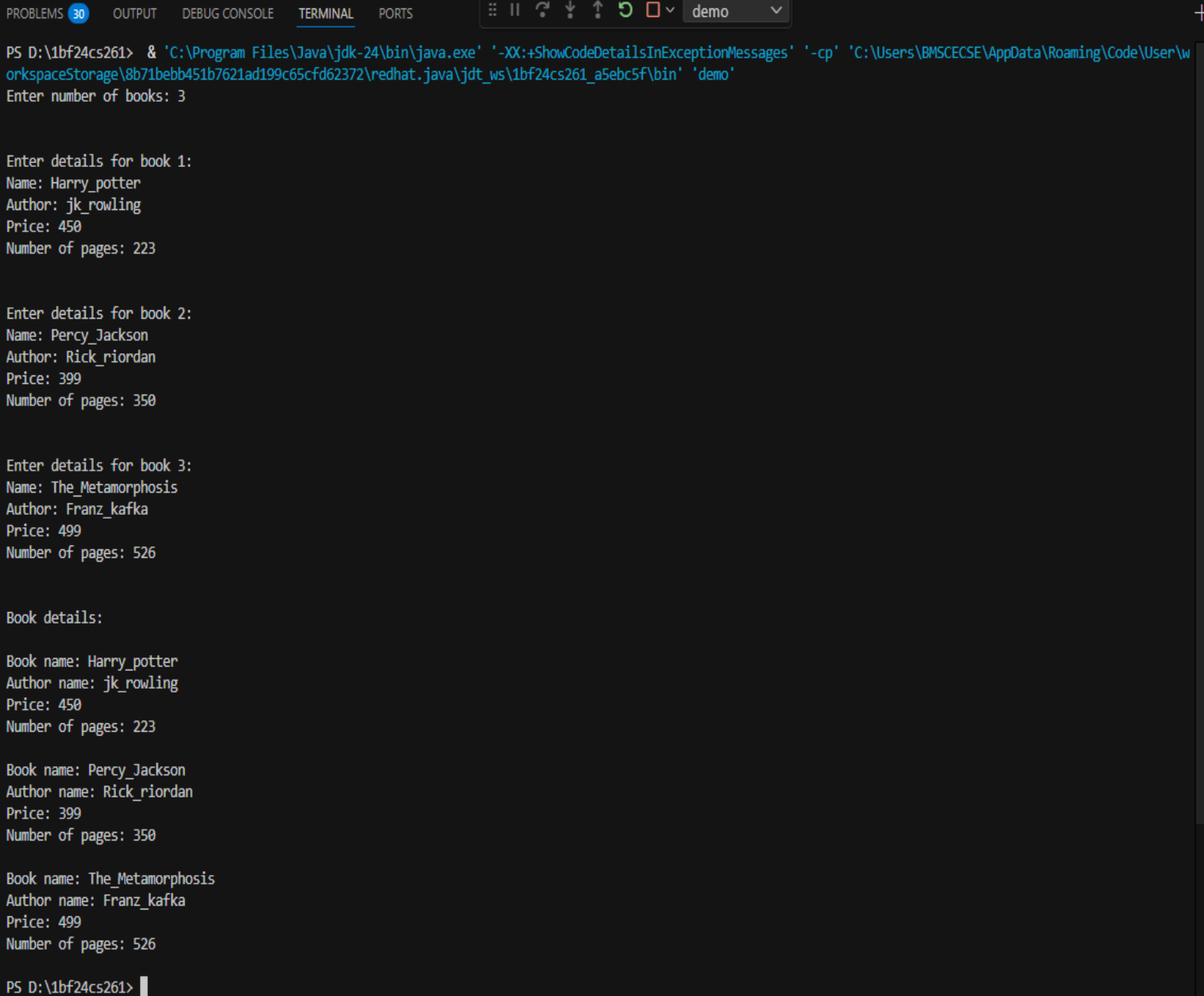
    System.out.println("\n\nBook details:\n");

    for (int i = 0; i < n; i++)
    {
        System.out.println(b[i].toString());
    }

    in.close();
}
}

```

//Output:



```
PS D:\1bf24cs261> & 'C:\Program Files\Java\jdk-24\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\BMSCECSE\AppData\Roaming\Code\User\workspaceStorage\8b71bebb451b7621ad199c65cfd62372\redhat.java\jdt_ws\1bf24cs261_a5ebc5f\bin' 'demo'
Enter number of books: 3

Enter details for book 1:
Name: Harry_potter
Author: jk_rowling
Price: 450
Number of pages: 223

Enter details for book 2:
Name: Percy_Jackson
Author: Rick_riordan
Price: 399
Number of pages: 350

Enter details for book 3:
Name: The_Metamorphosis
Author: Franz_kafka
Price: 499
Number of pages: 526

Book details:

Book name: Harry_potter
Author name: jk_rowling
Price: 450
Number of pages: 223

Book name: Percy_Jackson
Author name: Rick_riordan
Price: 399
Number of pages: 350

Book name: The_Metamorphosis
Author name: Franz_kafka
Price: 499
Number of pages: 526

PS D:\1bf24cs261>
```

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);

    public int getIntInput(String prompt) {
        System.out.print(prompt);
        return sc.nextInt();
    }
}

abstract class Shape extends InputScanner {
    int dim1, dim2;

    public void inputDimensions(String label1, String label2) {
        dim1 = getIntInput("Enter " + label1 + ": ");
        dim2 = getIntInput("Enter " + label2 + ": ");
    }

    abstract void printArea();
}

class Rectangle extends Shape {
    void input() {
        inputDimensions("length", "breadth");
    }

    void printArea() {
```

```

        int area = dim1 * dim2;
        System.out.println("Area of Rectangle: " + area);
    }
}

```

```

class Triangle extends Shape {
    void input() {
        inputDimensions("base", "height");
    }

    void printArea() {
        double area = 0.5 * dim1 * dim2;
        System.out.println("Area of Triangle: " + area);
    }
}

```

```

class Circle extends Shape {
    void input() {
        dim1 = getIntInput("Enter radius: ");
        dim2 = 0;
    }

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

```

```

public class lab4 {
    public static void main(String[] args) {
        Rectangle rect = new Rectangle();
        Triangle tri = new Triangle();
        Circle circ = new Circle();

        System.out.println("\n--- Rectangle ---");
        rect.input();
    }
}

```

```
        rect.printArea();

        System.out.println("\n--- Triangle ---");
        tri.input();
        tri.printArea();

        System.out.println("\n--- Circle ---");
        circ.input();
        circ.printArea();
    }
}
```

//Output:

```
--- Rectangle ---
Enter length: 12
Enter breadth: 123
Area of Rectangle: 1476

--- Triangle ---
Enter base: 1232
Enter height: 12323
Area of Triangle: 7590968.0

--- Circle ---
Enter radius: 12
Area of Circle: 452.3893421169302
```

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;
    String accountNumber;
    String accountType;
    double balance;

    Account(String c, String aN, String aT, double b) {
        this.customerName = c;
        this.accountNumber = aN;
        this.accountType = aT;
        this.balance = b;
    }

    void deposit(double amount)
    {
        if (amount > 0)
```

```

        {
            balance += amount;
            System.out.println("Deposited: " + amount);
        }
        else
        {
            System.out.println("Invalid deposit amount.");
        }
    }
}

void displayBalance() {
    System.out.println("Current Balance: Rs. " + balance);
}
}

class Saving extends Account
{
    double INTEREST_RATE = 0.05;

    Saving(String customerName, String accountNumber, double initialBalance)
    {
        super(customerName, accountNumber, "Savings", initialBalance);
    }

    void computeAndDepositInterest(int years) {
        double interest = balance * Math.pow((1 + INTEREST_RATE), years) - balance;
        balance += interest;
        System.out.println("Interest of Rs. " + String.format("%.2f", interest) + " added to your
account.");
    }
    void withdraw(double amount)
    {
        if (amount <= balance)
        {
            balance -= amount;
            System.out.println("Withdrawn:" + amount);
        } else
        {

```



```

        System.out.println("Insufficient balance!");
    }
}

class Current extends Account {
    double MIN_BALANCE = 1000.0;
    double SERVICE_CHARGE = 50.0;

    public Current(String customerName, String accountNumber, double initialBalance)
    {
        super(customerName, accountNumber, "Current", initialBalance);
    }

    void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn: Rs. " + amount);
            checkMinimumBalance();
        } else {
            System.out.println("Insufficient balance!");
        }
    }

    void checkMinimumBalance() {
        if (balance < MIN_BALANCE) {
            balance -= SERVICE_CHARGE;
            System.out.println("Balance below minimum! Service charge of Rs." +
SERVICE_CHARGE + " imposed.");
        }
    }
}

public class bank1 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);

        System.out.println("Enter Customer Name:");
        String name = in.nextLine();
    }
}

```

```
System.out.println("Enter Account Number:");
String accNo = in.nextLine();
```

```
System.out.println("Enter Account Type (savings/current):");
String type = in.nextLine().toLowerCase();
```

```
System.out.println("Enter Initial Balance:");
double balance = in.nextDouble();
```

```
Account account;
```

```
if (type.equals("savings"))
{
    account = new Saving(name, accNo, balance);
}
else
{
    account = new Current(name, accNo, balance);
}
```

```
int choice;
do {
    System.out.println("\nOptions Available");
    System.out.println("1. Deposit");
    System.out.println("2. Withdraw");
    System.out.println("3. Display Balance");
    if (account instanceof Saving)
        System.out.println("4. Compute and Deposit Interest");
    System.out.println("5 . Exit");
    System.out.print("Enter choice: ");
    choice = in.nextInt();
```

```
    switch (choice) {
        case 1:
            System.out.print("Enter amount to deposit: ");
            double depositAmount = in.nextDouble();
            account.deposit(depositAmount);
            break;
```

case 2:

```
System.out.print("Enter amount to withdraw: ");
double withdrawAmount = in.nextDouble();
if(account instanceof Saving) {
    ((Saving) account).withdraw(withdrawAmount);
} else {
    ((Current) account).withdraw(withdrawAmount);
}
break;
```

case 3:

```
account.displayBalance();
break;
```

case 4:

```
if (account instanceof Saving) {
    System.out.print("Enter number of years for interest: ");
    int years = in.nextInt();
    ((Saving) account).computeAndDepositInterest(years);
} else {
    System.out.println("Interest computation not available for Current Account.");
}
break;
```

case 5:

```
System.out.println("Exiting Program ");
System.exit(0);
```

default:

```
System.out.println("Invalid choice. Try again.");
```

```
}
```

```
} while (choice != 0);
```

```
in.close();
```

```
}
```

```
}
```

//Output:

```
PS C:\Users\Admin\Desktop\1bf24cs261> cd "c:\Users\Admin\Desktop\1bf24cs261\" ; if ($?) { javac bank1.java } ; if ($?) { java bank1 }
Enter Customer Name:
sagarmatha
Enter Account Number:
1
Enter Account Type (savings/current):
savings
Enter Initial Balance:
1000000

Options Available
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest
5 . Exit
Enter choice: 1
Enter amount to deposit: 2333
Deposited: 2333.0

Options Available
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest
5 . Exit
Enter choice: 2
Enter amount to withdraw: 10000
Withdrawn:10000.0

Options Available
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest
5 . Exit
Enter choice: 3
Current Balance: Rs. 992333.0

Options Available
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest
```

Run Terminal Help ← → 1bf24cs261

OUTPUT PROBLEMS 13 DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER

```
3. Display Balance
4. Compute and Deposit Interest
5 . Exit
Enter choice: 1
Enter amount to deposit: 2333
Deposited: 2333.0

Options Available
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest
5 . Exit
Enter choice: 2
Enter amount to withdraw: 10000
Withdrawn:10000.0

Options Available
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest
5 . Exit
Enter choice: 3
Current Balance: Rs. 992333.0

Options Available
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest
5 . Exit
Enter choice: 4
Enter number of years for interest: 3
Interest of Rs. 156416.49 added to your account.

Options Available
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest
5 . Exit
Enter choice: 
```

```
PS C:\Users\Admin\Desktop\1bf24cs261> cd "c:\Users\Admin\Desktop\1bf24cs261\" ; if ($?) { javac bank1.java } ; if ($?) { java bank1 }
Enter Customer Name:
Sagar
Enter Account Number:
2
Enter Account Type (savings/current):
current
Enter Initial Balance:
199999

Options Available
1. Deposit
2. Withdraw
3. Display Balance
5 . Exit
Enter choice: 1
Enter amount to deposit: 12323
Deposited: 12323.0

Options Available
1. Deposit
2. Withdraw
3. Display Balance
5 . Exit
Enter choice: 2
Enter amount to withdraw: 11111
Withdrawn: Rs. 11111.0

Options Available
1. Deposit
2. Withdraw
3. Display Balance
5 . Exit
Enter choice: 3
Current Balance: Rs. 201211.0

Options Available
1. Deposit
2. Withdraw
3. Display Balance
5 . Exit
Enter choice: 5
Exiting Program
PS C:\Users\Admin\Desktop\1bf24cs261> 
```

Program 6:

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

//Code:



CIE

internals .java

```
package CIE;
public class Internals extends Student {
    private int[] internalMarks;

    public Internals(String usn, String name, int sem, int[] internalMarks) {
        super(usn, name, sem);
        this.internalMarks = new int[5];
        System.arraycopy(internalMarks, 0, this.internalMarks, 0, 5);
    }
    public int[] getInternalMarks() {
        int[] copy = new int[5];
        System.arraycopy(internalMarks, 0, copy, 0, 5);
        return copy;
    }

    public int getInternalMark(int idx) {
        return internalMarks[idx];
    }
}
```

Student.java

```
package CIE;
```

```
public class Student {  
    protected String usn;  
    protected String name;  
    protected int sem;  
  
    public Student(String usn, String name, int sem) {  
        this.usn = usn;  
        this.name = name;  
        this.sem = sem;  
    }  
  
    public String getUsn() { return usn; }  
    public String getName() { return name; }  
    public int getSem() { return sem; }  
  
    @Override  
    public String toString() {  
        return "USN: " + usn + ", Name: " + name + ", Sem: " + sem;  
    }  
}
```




SEE

External.java

```
package SEE;
```

```
import CIE.Student;
```

```
public class External extends Student {  
    private int[] externalMarks;
```

```
    public External(String usn, String name, int sem, int[] externalMarks) {  
        super(usn, name, sem);  
        this.externalMarks = new int[5];  
        System.arraycopy(externalMarks, 0, this.externalMarks, 0, 5);  
    }
```

```
    public int[] getExternalMarks() {  
        int[] copy = new int[5];  
        System.arraycopy(externalMarks, 0, copy, 0, 5);  
        return copy;  
    }
```

```
    public int getExternalMark(int idx) {  
        return externalMarks[idx];  
    }  
}
```

FinalMarksApp.java / main.java

```
import CIE.Internals;
import SEE.External;
import java.util.*;

public class FinalMarksApp {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Number of students: ");
        int n = sc.nextInt();

        Internals[] in = new Internals[n];
        External[] ex = new External[n];

        for (int i = 0; i < n; i++) {
            System.out.println("\nStudent " + (i + 1));

            System.out.print("USN: ");
            String usn = sc.next();

            sc.nextLine();
            System.out.print("Name: ");
            String name = sc.nextLine();

            System.out.print("Semester: ");
            int sem = sc.nextInt();

            int[] im = new int[5];
            System.out.print("Enter 5 internal marks: ");
            for (int j = 0; j < 5; j++) im[j] = sc.nextInt();

            int[] em = new int[5];
            System.out.print("Enter 5 external marks: ");
            for (int j = 0; j < 5; j++) em[j] = sc.nextInt();

            in[i] = new Internals(usn, name, sem, im);
            ex[i] = new External(usn, name, sem, em);
        }
    }
}
```

```
}

System.out.println("\nFinal Marks:");
for (int i = 0; i < n; i++) {
    System.out.println("\n" + in[i].getName() + " (" + in[i].getUsn() + ")");
    System.out.print("Marks: ");
    for (int j = 0; j < 5; j++) {
        System.out.print(in[i].getInternalMark(j) + ex[i].getExternalMark(j) + " ");
    }
}

sc.close();
}
}
```

//Output:

```
Enter Semester: 2
Enter CIE marks for 5 subjects:
CIE Mark 1: 41
CIE Mark 2: 45
CIE Mark 3: 44
CIE Mark 4: 50
CIE Mark 5: 47
Enter SEE marks for 5 subjects:
SEE Mark 1: 50
SEE Mark 2: 40
SEE Mark 3: 44
SEE Mark 4: 41
SEE Mark 5: 38
```

```
Enter details of student 2
Enter USN: 1BF24CS260
Enter Name: Sagarmatha Khatri
Enter Semester: 2
Enter CIE marks for 5 subjects:
CIE Mark 1: 35
CIE Mark 2: 45
CIE Mark 3: 43
CIE Mark 4: 41
CIE Mark 5: 40
Enter SEE marks for 5 subjects:
SEE Mark 1: 45
SEE Mark 2: 41
SEE Mark 3: 42
SEE Mark 4: 40
SEE Mark 5: 38
```

FINAL MARKS

```
Student 1:
USN: 1BF24CS253
Name: Rishi Kumar Chourasia
Semester: 2
Final marks (CIE + SEE):
Subject 1: 91
Subject 2: 85
Subject 3: 88
Subject 4: 91
Subject 5: 85
```

```
Student 2:
USN: 1BF24CS260
Name: Sagarmatha Khatri
Semester: 2
Final marks (CIE + SEE):
Subject 1: 80
Subject 2: 86
Subject 3: 85
Subject 4: 81
Subject 5: 78
```

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.*;

class InputScanner {
    Scanner sc;

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

    int getInt() {
        return sc.nextInt();
    }
}

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

class father extends InputScanner {
    int fatherAge;

    father() throws WrongAge {
        System.out.print("Enter Father's Age: ");
        fatherAge = getInt();
    }
}
```

```

        if (fatherAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
    }

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

class Son extends father {
    int sonAge;

    Son() throws WrongAge {
        super();

        System.out.print("Enter Son's Age: ");
        sonAge = getIn();

        if (sonAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
        int gap= fatherAge-sonAge;
        if (gap<=18){
            throw new WrongAge("Impossible age difference");
        }

        if (sonAge >= fatherAge) {
            throw new WrongAge("Son's age cannot be greater than or equal to father's age");
        }
    }

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

public class lab7 {

```

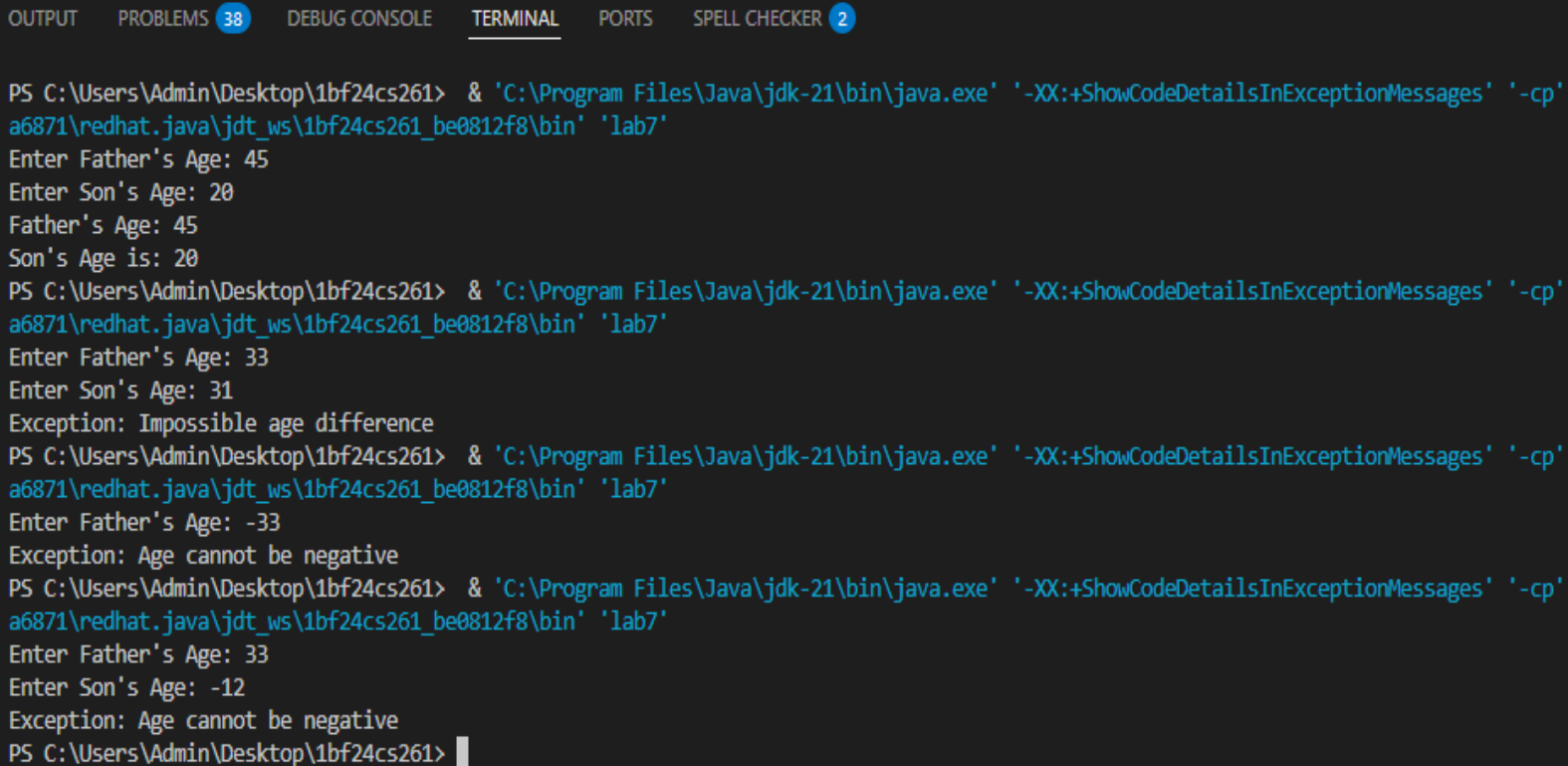
```

    public static void main(String[] args) {
    try {
        Son s = new Son();
        s.display();

    } catch (WrongAge e) {
        System.out.println("Exception: " + e.getMessage());
    }
    }
}

```

//Output:



The screenshot shows an IDE terminal window with the following tabs: OUTPUT, PROBLEMS (38), DEBUG CONSOLE, TERMINAL (selected), PORTS, and SPELL CHECKER (2). The terminal displays the execution of a Java program with the following sequence of events:

```

PS C:\Users\Admin\Desktop\1bf24cs261> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp'
a6871\redhat.java\jdt_ws\1bf24cs261_be0812f8\bin' 'lab7'
Enter Father's Age: 45
Enter Son's Age: 20
Father's Age: 45
Son's Age is: 20
PS C:\Users\Admin\Desktop\1bf24cs261> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp'
a6871\redhat.java\jdt_ws\1bf24cs261_be0812f8\bin' 'lab7'
Enter Father's Age: 33
Enter Son's Age: 31
Exception: Impossible age difference
PS C:\Users\Admin\Desktop\1bf24cs261> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp'
a6871\redhat.java\jdt_ws\1bf24cs261_be0812f8\bin' 'lab7'
Enter Father's Age: -33
Exception: Age cannot be negative
PS C:\Users\Admin\Desktop\1bf24cs261> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp'
a6871\redhat.java\jdt_ws\1bf24cs261_be0812f8\bin' 'lab7'
Enter Father's Age: 33
Enter Son's Age: -12
Exception: Age cannot be negative
PS C:\Users\Admin\Desktop\1bf24cs261>

```

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 {
            while (true) {
                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 {
            while (true) {
                System.out.println("CSE");
                Thread.sleep(2000); // 2 seconds
            }
        } catch (InterruptedException e) {
            System.out.println("CSEThread interrupted");
        }
    }
}

public class main{
    public static void main(String[] args) {
        CollegeThread t1 = new CollegeThread();
        CSEThread t2 = new CSEThread();
    }
}
```


//Output:

The image shows a screenshot of an IDE's interface. At the top, there is a menu bar with the following items: OUTPUT, PROBLEMS (with a blue circle containing the number 24), DEBUG CONSOLE, TERMINAL (which is currently selected and underlined), PORTS, and SPELL CHECKER. On the right side of the top bar, there are icons for window management (a plus sign, a downward arrow, and an ellipsis) and a close button (an 'X'). Below the menu bar, the main area is a terminal window. It contains the following text:

```
PS C:\Users\Admin\Desktop\1bf24cs261> cd "c:\Users\Admin\Desktop\1bf24cs261" ; if ($?) { javac main.java } ; if ($?) { java main }
Error: Could not find or load main class main
Caused by: java.lang.ClassNotFoundException: main
PS C:\Users\Admin\Desktop\1bf24cs261> cd "c:\Users\Admin\Desktop\1bf24cs261" ; if ($?) { javac main.java } ; if ($?) { java main }
Error: Could not find or load main class main
Caused by: java.lang.ClassNotFoundException: main
PS C:\Users\Admin\Desktop\1bf24cs261> cd "c:\Users\Admin\Desktop\1bf24cs261" ; if ($?) { javac main.java } ; if ($?) { java main }
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
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
```

The text is color-coded: 'cd' is green, 'javac' and 'java' are yellow, and the rest is white. On the right side of the IDE, there is a sidebar. It has two buttons: 'Code' (with a document icon) and 'Run: main' (with a play icon). The 'Code' button is currently selected and highlighted in a dark gray color.