

**B.M.S. College of Engineering**  
*(Autonomous Institution affiliated to VTU, Belagavi)*  
**Department of Computer Science and Engineering**



**LAB**

**OBJECT ORIENTED JAVA  
PROGRAMMING REPORT**

**23CS3PCOOJ**

**(December 2023-March 2024)**

*Submitted by:*

**VANSH VERMA**

**B.M.S. College of Engineering**  
**Department of Computer Science and Engineering**



**Laboratory Certificate**

This is to certify that VANSH VERMA has satisfactorily completed the course of Experiments in Practical OBJECT ORIENTED PROGRAMMING prescribed by the Department during the odd semester 24.

Name of the Candidate: VANSH VERMA

USN No.: 1BM22CS318    Semester: III    Section: F

Marks	
Max. Marks	Obtained
10	
Marks in Words	

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

```
class quadratic {  
    int a, b, c;    double  
    r1, r2, d;
```

```
    void getd() {  
        Scanner s = new Scanner(System.in);  
        System.out.println("Enter the values of a, b, c");  
        a = s.nextInt();    b = s.nextInt();    c =  
        s.nextInt();  
    }
```

```
    void compute() {  
        while (a == 0) {  
            System.out.println("Not a quadratic equation");  
            System.out.println("Enter a non-zero value of a");  
            Scanner s = new Scanner(System.in);  
            a = s.nextInt();  
        }    d = (b * b) - (4 *
```

```
        a * c);
```

```
        if (d == 0) {  
            r1 = -b / (2 * (double) a);  
            System.out.println("Roots are real and equal");  
            System.out.println("Roots are Root1=Root2=" + r1);  
        }  
        if (d > 0) {  
            r1 = (-b + Math.sqrt(d)) / (2 * (double) a);  
            r2 = (-b - Math.sqrt(d)) / (2 * (double) a);  
            System.out.println("Roots are real and distinct");  
            System.out.println("Root1=" + r1 + " and Root2=" + r2);  
        }  
        if (d < 0) {  
            r1 = (-b + Math.sqrt(d)) / (2 * (double) a);  
            r2 = (-b - Math.sqrt(d)) / (2 * (double) a);  
            System.out.println("Roots are complex and distinct");  
            System.out.println("Root1=" + r1 + " and Root2=" + r2);  
        }  
    }  
}
```

```

        System.out.println("Roots are imaginary and Root1="
+ "+i" + r2 + " and Root2=" + r1 + "-i" + r2);
    }
}
}

```

```

class QuadraticMain {
    public static void main(String[] args) {
        System.out.println("Bhuvana M");
        System.out.println("1BM22CS071");
        quadratic q = new quadratic();    q.getd();
        q.compute();
    } }

```

## **Output:**

```

cd "/Users/apple/Documents/Java/" && javac QuadraticMain.java && java QuadraticMain
apple@apples-MacBook-Pro-3 Java % cd "/Users/apple/Documents/Java/" && javac QuadraticMain.java && java QuadraticMain
Bhuvana M
1BM22CS071
Enter the values of a,b,c
10 20 30
Roots are imaginary and Root1=-100.0i1.4142135623730951and Root2=-100.0-i1.4142135623730951
apple@apples-MacBook-Pro-3 Java % cd "/Users/apple/Documents/Java/" && javac QuadraticMain.java && java QuadraticMain
Bhuvana M
1BM22CS071
Enter the values of a, b, c
0 0 0
Not a quadratic equation
Enter a non-zero value of a
20
Roots are real and equal
Roots are Root1=Root2=0.0
apple@apples-MacBook-Pro-3 Java %

```

**2. Create a class Book that contains four members: name, author, price, and 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 details of the book. Do not use any package.**

```
class books {  
    String name;  
    String author;  
    int price;    int  
    numPages;
```

```
    books(String name, String author, int price, int numPages)  
    this.name = name;    this.author = author;  
        this.price = price;  
        this.numPages = numPages;  
    }
```

```
        public String toString() {    return "Book  
Name: " + this.name + "\n" +    "Author  
Name: " + this.author + "\n" +  
        "Book Price: " + this.price + "\n" +  
        "Number of pages: " + this.numPages + "\n";  
    }  
}
```

```
class booksMain {  
    public static void main(String[] args) {  
        System.out.println("Bhuvana M");  
        System.out.println("1BM22CS071");  
        Scanner s = new Scanner(System.in);    int  
        n;  
        String name;
```

```

        books[] b;        b
= new books[n];

        for (int i = 0; i < n; i++) {
            System.out.println("Book " + (i + 1) + ":");
            System.out.println("Enter the book name");
            s.nextLine();        name = s.nextLine();
            System.out.println("Enter the author");
            author = s.nextLine();
            System.out.println("Enter the price");
            price = s.nextInt();
            System.out.println("Enter the number of pages");
            numPages = s.nextInt();

            b[i] = new books(name, author, price, numPages);
        }

        for (int i = 0; i < n; i++) {
            System.out.println("Book " + (i + 1) + "\n" + b[i]);
        }
    }
} Output:

```

```
● apple@apples-MacBook-Pro-3 Java % cd "/Users/apple/Documents/Java/" && javac booksMain.java && java booksMain
Bhuvana M
1BM22CS071
Enter the number of books:
2
Book 1:
Enter the book name
00JP TB
Enter the author
XYZ
Enter the price
1000
Enter the number of pages
500
Book 2:
Enter the book name
DS TB
Enter the author
ABC
Enter the price
1000
Enter the number of pages
450
Book 1
Book Name: 00JP TB
Author Name: XYZ
Book Price: 1000
Number of pages: 500

Book 2
Book Name: DS TB
Author Name: ABC
Book Price: 1000
Number of pages: 450

○ apple@apples-MacBook-Pro-3 Java %
```

***3. Write a Java program to create a class Student with members USN, name, marks(6 subjects). Include methods to accept student details and marks, Also include a method to calculate the percentage and display appropriate details. (Array of student object to be created).***

**Ans:**

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

```
class student {    String  
USN;    String name;  
int marks[] = new int[6];  
float percentage = 0;
```

```
    void getd(int i) {  
        Scanner s = new Scanner(System.in);  
        System.out.println("Enter USN: ");  
        USN = s.next();  
        System.out.println("Enter Name:");  
        name = s.next();  
        System.out.println("Enter Student" + i + " Marks");  
        for (int j = 0; j < 6; j++) {  
            System.out.println("Enter Marks of Subject" + j + ":");  
            marks[j] = s.nextInt();  
            percentage += marks[j];  
        }  
    }  
}
```



```

class studentMain {    public static void
main(String[] args) {
System.out.println("Bhuvana M");
    System.out.println("1BM22CS071");
    System.out.println("Enter the number of Students");
    Scanner sc = new Scanner(System.in);
    int n = sc.nextInt();
student s[] = new student[n];
for (int i = 0; i < n; i++) {
s[i] = new student();
    s[i].getd(i);
    }
    for (int i = 0; i < n; i++) {
        s[i].calculatePercentage(i);
    }
} }

```

### **Output:**

```

● apple@apples-MacBook-Pro-3 Java % cd "/Users/apple/Documents/Java/" && javac studentMain.java && java studentMain
Bhuvana M
1BM22CS071
Enter the number of Students
1
Enter USN:
1BM22CS001
Enter Name:
Rahul
Enter Student0 Marks
Enter Marks of Subject0:
90
Enter Marks of Subject1:
92
Enter Marks of Subject2:
98
Enter Marks of Subject3:
90
Enter Marks of Subject4:
95
Enter Marks of Subject5:
96

```

**Ans:**

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

```
abstract class shape {
```

```
    int dim1;
```

```
    int dim2;
```

```
    shape(int dim1, int dim2) {
```

```
        this.dim1 = dim1;        this.dim2
```

```
        = dim2;
```

```
    }
```

```
    abstract void printArea();
```

```
}
```

```
class rectangle extends shape {
```

```
    rectangle(int length, int breadth) {
```

```
        super(length, breadth);
```

```
    }
```

```
    void printArea() {
```

```
        double area = dim1 * dim2;
```

```
        System.out.println("Area of rectangle = " + area);
```

```
    }
```

```
}
```

```
class triangle extends shape {
```

```
    triangle(int height, int base) {
```

```
}
```

```
class circle extends shape {  
    circle(int radius) {  
        super(radius, 0);  
    }  

```

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

```
public class AbstractMain {    public  
    static void main(String[] args) {  
        System.out.println("Bhuvana M");  
        System.out.println("1BM22CS071");  
        Scanner s = new Scanner(System.in);
```

```
        System.out.println("Enter the length and breadth of the  
        rectangle");        int l = s.nextInt();        int b = s.nextInt();  
        System.out.println("Enter base and height of the triangle");  
        int ba = s.nextInt();        int h = s.nextInt();
```

```
        System.out.println("Enter the radius of the circle");  
        int r = s.nextInt();
```

```
        rectangle re = new rectangle(l, b);
```

## Output:

```
● apple@apples-MacBook-Pro-3 Java % cd "/Users/apple/Documents/Java/" && javac abstractmain.java && java abstr
Bhuvana M
1BM22CS071
Enter the length and breadth of the rectangle
5 10
Enter base and height of the triangle
4 5
Enter the radius of the circle
8
Area of rectangle =50.0
Area of triangle =10.0
Area of circle =200.96
○ apple@apples-MacBook-Pro-3 Java % █
```

***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 the 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:***

- Accept deposit from customer and update the balance.***
- Display the balance.***
- Compute and deposit interest***
- Permit withdrawal and update the balance***

***Check for the minimum balance, impose penalty if necessary and update the balance. Ans:***

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

```
Account(String name, int number, String type, double
initialBalance) {      customerName = name;
accountNumber = number;      accountType = type;
    balance = initialBalance;
}
```

```
void deposit(double amount) {
    if (amount > 0) {
balance += amount;
        System.out.println("Deposit of INR " + amount + "
successful");
    } else {
        System.out.println("Invalid deposit amount. Please en
positive value.");
    }
}
```

```
void displayBalance() {
    System.out.println("Account Number: " + accountNumber);
    System.out.println("Customer Name: " + customerName);
    System.out.println("Account Type: " + accountType);
    System.out.println("Balance: INR " + balance);
}
```

```
void withdraw(double amount) {
if (balance >= amount) {
balance -= amount;
```

```

    void computeInterest() {
    }
    void checkMinimumBalance(double minBalance, double
serviceCharge) {
    }
}

```

```

class SavAcct extends Account {
double interestRate = 0.05;

```

```

    SavAcct(String name, int number, String type, double
initialBalance) {
        super(name, number, type, initialBalance);
    }

```

```

    void computeInterest() {
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest of INR " + interest + " added
the account");
    }
}

```

```

class CurAcct extends Account {
double minBalance = 1000;    double
serviceCharge = 50;

```

```
        System.out.println("Service charge of INR " +
serviceCharge + " imposed");
        balance -= serviceCharge;
    }
}
}
```

```
public class Bank {
    public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
System.out.print("Enter the number of users: ");
int numUsers = scanner.nextInt();
```

```
        Account[] accounts = new Account[numUsers];

        for (int i = 0; i < numUsers; i++) {
System.out.println("\nUser " + (i + 1));
System.out.print("Enter customer name: ");
            scanner.nextLine();
            String name = scanner.nextLine();
            System.out.print("Enter account number: ");
int accNumber = scanner.nextInt();
            System.out.print("Enter initial deposit amount: INR ");
double initialDeposit = scanner.nextDouble();
            System.out.print("Enter account type (Savings/Current
");
            scanner.nextLine();
```

```
        } else if (accType.equalsIgnoreCase("Current")) {
accounts[i] = new CurAcct(name, accNumber, accType,
initialDeposit);
        } else {
            System.out.println("Invalid account type entered.
Defaulting to Account.");
            accounts[i] = new Account(name, accNumber,
"Account", initialDeposit);
        }
    }
}
```

```
    boolean exit = false;
while (!exit) {
    System.out.println("\nChoose an option:");
    System.out.println("1. Deposit");
    System.out.println("2. Withdraw");
    System.out.println("3. Display Balance");
    System.out.println("4. Compute Interest (Savings only)");
    System.out.println("5. Exit");
    System.out.print("Enter your choice: ");
while (!scanner.hasNextInt()) {
        System.out.println("Invalid input. Please enter a
number.");
        scanner.next();
    }
    int choice = scanner.nextInt();
}
```



```

        for (Account acc : accounts) {
    if (acc.accountNumber == accNum) {
        acc.deposit(depositAmount);
    }
}
break;
case 2:
    System.out.print("Enter account number: ");
    accNum = scanner.nextInt();
    System.out.print("Enter withdrawal amount: INR ");
    double withdrawAmount = scanner.nextDouble();
    for (Account acc : accounts) {
    if (acc.accountNumber == accNum) {
        acc.withdraw(withdrawAmount);
    }
}
break;
case 3:
    System.out.print("Enter account number: ");
    accNum = scanner.nextInt();
    for (Account acc : accounts) {
        if (acc.accountNumber == accNum) {
            acc.displayBalance();
        }
    }
break;
case 4:

```

```

        }
        break;

case 5:
    exit = true;
    break;
default:
    System.out.println("Invalid choice. Please enter a
valid option.");
    }
    }

} }

```

## **Output:**

```

● apple@apples-MacBook-Pro-3 Java % cd "/Users/apple/Documents/Java/" && javac bankmain.java && java bankmain
Bhuvana M
1BM22CS071
Enter the number of users: 1

User 1
Enter customer name: Rahul
Enter account number: 123
Enter initial deposit amount: INR 6000
Enter account type (Savings/Current): Savings

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 1
Enter account number: 123
Enter deposit amount: INR 1500
Deposit of INR 1500.0 successful

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 2
Enter account number: 123
Enter withdrawal amount: INR 500
Withdrawal of INR 500.0 successful

Choose an option:
1. Deposit
2. Withdraw

```

```

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 4
Enter account number (for Savings account): 123
Interest of INR 350.0 added to the account

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 5
○ apple@apples-MacBook-Pro-3 Java % █

```

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

- ***Create a folder CIE and save the programs Student.java and Internals.java within it.***
- ***Create a folder SEE and save the program External.java within it.***
- ***Save the Main program outside these two folders.***
- ***Compile Main.java and Execute the Main.class***

**Ans:**

```
    public String usn;  
    public String name;  
    public int sem;
```

```
    public student(String usn, String name, int sem) {  
        this.usn = usn;        this.name = name;  
        this.sem = sem;  
    }  
}
```

```
//internals.java package CIE; public
```

```
class internals extends student {
```

```
    public int[] internalMarks;
```

```
    public internals(String usn, String name, int sem, int[]  
internalMarks) {        super(usn, name, sem);  
        this.internalMarks = internalMarks;  
    }  
}
```

```
//SEE Package
```

```
//externals.java package
```

```
SEE; import
```

```

    public externals(String usn, String name, int sem, int[]
seeMarks) {
        super(usn, name, sem);
        this.seeMarks = seeMarks;
    }
}

```

//main1.java

```

import CIE.internals; import
SEE.externals; import
java.util.Scanner;

```

```

public class main1 {    public static void
main(String[] args) {
    System.out.println("Bhuvana M");
        System.out.println("1BM22CS071");

```

```

        Scanner scanner = new Scanner(System.in);

```

```

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

```

```

        internals[] cieStudents = new internals[n];
externals[] seeStudents = new externals[n];

```

```

        // Input CIE marks
for (int i = 0; i < n; i++) {

```

```
String name = scanner.next();
System.out.print("Semester: ");          int
sem = scanner.nextInt();
```

```
int[] cieMarks = new int[5];
System.out.print("Enter CIE marks for 5 courses: ");
for (int j = 0; j < 5; j++) {           cieMarks[j] =
scanner.nextInt();
}
```

```
cieStudents[i] = new internals(usn, name, sem, cieMa
}
```

```
// Input SEE marks
for (int i = 0; i < n; i++) {
    System.out.println("Enter details for SEE of student "
+ 1));
    System.out.print("USN: ");
    String usn = scanner.next();
    System.out.print("Name: ");
    String name = scanner.next();
    System.out.print("Semester: ");      int
    sem = scanner.nextInt();
```

```
int[] seeMarks = new int[5];
System.out.print("Enter SEE marks for 5 courses: ");
for (int j = 0; j < 5; j++) {
```

```

for (int i = 0; i < n; i++) {
    System.out.println("\nDetails of Student " + (i + 1));
    System.out.println("USN: " + cieStudents[i].usn);
    System.out.println("Name: " + cieStudents[i].name);
    System.out.println("Semester: " + cieStudents[i].sem);
    System.out.println("CIE Marks: ");
    for (int j = 0; j < 5; j++) {
        System.out.print(cieStudents[i].internalMarks[j] + ' ');
    }
    System.out.println("\nSEE Marks: ");
    for (int j = 0; j < 5; j++) {
        System.out.print(seeStudents[i].seeMarks[j] + " ");
    }
}
}
}

```

## **Output:**

```

● apple@apples-MacBook-Pro-3 Java % javac main1.java
● apple@apples-MacBook-Pro-3 Java % java main1
Bhuvana M
1BM22CS071
Enter the number of students: 1
Enter details for CIE of student 1
USN: 1BM22CS001
Name: Rahul
Semester: 3
Enter CIE marks for 5 courses: 40 39 40 38 40
Enter details for SEE of student 1
USN: 1BM22CS001
Name: Rahul
Semester: 3
Enter SEE marks for 5 courses: 100 99 98 100 97

Final Marks of Students:

Details of Student 1
USN: 1BM22CS001
Name: Rahul
Semester: 3
CIE Marks:

```

*class, implement a constructor that cases both father and so age and throws an exception if son's age is  $\geq$  father's age.*

**Ans:**

```
class WrongAgeException extends Exception {  
    WrongAgeException(String message) {  
        super(message);  
    }  
}
```

```
class Father {  
    private int age;
```

```
    public Father(int age) throws WrongAgeException {  
        if (age < 0) {  
            throw new WrongAgeException("Father's age cannot  
negative");  
        }  
        this.age = age;  
    }
```

```
    public int getAge() {  
        return age;  
    }
```

```
}  
class Son extends Father {  
    int age;
```



```
        throw new WrongAgeException("Son's age should be  
than Father's age");  
    }
```

```
    this.sonAge = sonAge;  
}
```

```
public int getSonAge() {  
    return sonAge;  
}  
}
```

```
public class InheritanceExceptionDemo {  
    public static void main(String[] args) {  
try {  
        Father father = new Father(40);  
        System.out.println("Father's age: " + father.getAge());  
  
        Son son = new Son(40, 20); // This will throw an  
exception due to son's age being >= father's age  
        System.out.println("Son's age: " + son.getSonAge());  
    } catch (WrongAgeException e) {  
        System.out.println("Exception: " + e.getMessage());  
    }  
}  
}
```

***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.***

**Ans:**

```
class DisplayThread extends Thread {  
    private String message;  
    private int intervalMillis;  
  
    public DisplayThread(String message, int intervalMillis) {  
        this.message = message;  
        this.intervalMillis = intervalMillis;  
    }  
  
    public void run() {  
        while (true) {  
            try {  
                System.out.println(message);  
                Thread.sleep(intervalMillis);  
            } catch (InterruptedException e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}
```

```
public class DisplayProgram {
```

```
        thread1.start();
thread2.start();
    } }
```

**Output:**

[illegible]

Dept. of CSE, B.M.S College of Engineering