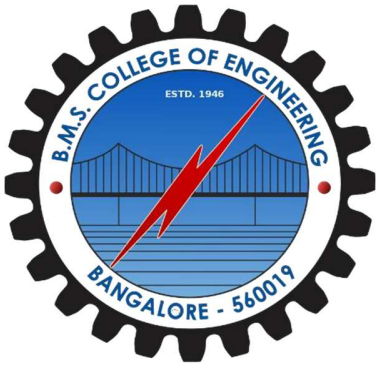
B.M.S. College of Engineering

(Autonomous College Affiliated to Visvesvaraya Technological University, Belgaum)
Bull Temple Road, Basavanagudi, Bengaluru – 560019



Department of

Computer Science & Engineering (CSE)

Lab Programs Report

Course Title: Object Oriented Java Programming

Course Code: 23CS3PCOOJ

BY

Darshna Mimrot (1BM22CS081)

B.M.S. College of Engineering

(Autonomous College Affiliated to Visvesvaraya Technological University, Belgaum)
Bull Temple Road, Basavanagudi, Bengaluru – 560019

Department of
Computer Science & Engineering (CSE)

CERTIFICATE

This is to certify that the report on "**Java Lab Programs**" has been carried out by **Darshna Mimrot** bearing USN **1BM22CS081** as a part of AAT for the course **Object Oriented Java Programming** with course code **23CS3PCOOJ**,
Computer Science and Engineering from Visvesvaraya Technological University, Belgaum during the year 2023–24. It is certified that all corrections/suggestions indicated for Internal Assessments have been incorporated in the report.

Darshna Mimrot
1BM22CS081

Shravya AR
Assistant Professor
Department of CSE
BMSCE, Bengaluru-19

Table of contents

LAB PROGRAM 1

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 percentage of a student.

CODE

```
import java.util.Scanner;

class Student {
    String usn;
    String name;
    int[] credits = new int[8];
    int[] marks = new int[8];
    // Method to accept student details
    public void acceptDetails() {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter USN: ");
        usn = scanner.nextLine();

        System.out.print("Enter Name: ");
        name = scanner.nextLine();

        System.out.println("Enter details for each subject : \n");

        for (int i = 0; i < credits.length; i++) {
            System.out.print("\nEnter credits for Subject " + (i + 1) + ": ");
            credits[i] = scanner.nextInt();

            System.out.print("\nEnter marks for Subject " + (i + 1) + ": ");
            marks[i] = scanner.nextInt();
        }
        scanner.close();
    }

    // Method to calculate SGPA
    public double calculateSGPA() {
        int totalCredits = 0;
        int weightedSum = 0;
        double ans;

        for (int i = 0; i < credits.length; i++) {
            totalCredits += credits[i];
            int gradePoints;

            gradePoints = (marks[i]/10)+1;

            if(gradePoints == 11){
                gradePoints=10;
            }
```

```
            else if(gradePoints<=4){
                gradePoints=0;
            }

            weightedSum += gradePoints * credits[i];
        }

        ans = (double) weightedSum / (double) totalCredits;
        return ans;
    }
}

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

        // Create a Student object
        Student student = new Student();

        // Accept and display details
        student.acceptDetails();

        System.out.println("\nStudent Details :");

        System.out.println("USN : " + student.usn);
        System.out.println("Name : " + student.name);
        // Calculate and display SGPA
        double sgpa = student.calculateSGPA();
        System.out.println("\nSGPA: " + sgpa);
        scanner.close();
    }
}
```

OUTPUT

```
C:\Users\Darshna Mimrot\OneDrive\Desktop\codes>java sgpa
Enter USN: 1BM22CS008
Enter Name: DHANUSH K
Enter details for each subject :


Enter credits for Subject 1: 4

Enter marks for Subject 1: 79

Enter credits for Subject 2: 4

Enter marks for Subject 2: 90

Enter credits for Subject 3: 3

Enter marks for Subject 3: 89

Enter credits for Subject 4: 3

Enter marks for Subject 4: 88

Enter credits for Subject 5: 2

Enter marks for Subject 5: 98

Enter credits for Subject 6: 1

Enter marks for Subject 6: 99

Enter credits for Subject 7: 1

Enter marks for Subject 7: 99

Enter credits for Subject 8: 4

Enter marks for Subject 8: 89

Student Details :
USN : 1BM22CS008
Name : DHANUSH K

SGPA: 9.181818181818182

C:\Users\Darshna Mimrot\OneDrive\Desktop\codes>
```

Lab program 2

Develop a Java program that prints all real solutions to the quadratic equation ax2+bx+c = 0.Read in a, b, c and use the quadratic formula. If the discriminate b2-4ac is negative, display amessage stating that there are no real solutions.

CODE

```
import java.util.Scanner;
class Quad {
int a, b, c;
double root1, root2, d;
Scanner s = new Scanner(System.in);
void input()
{
System.out.println("Quadratic equation is in the form : ax^2 + bx + c");
System.out.print("Enter a:");
a = s.nextInt();
System.out.print("Enter b:");
b = s.nextInt();
System.out.print("Enter c:");
c = s.nextInt();
}
void discriminant() {
d= (b*b)-(4*a*c);
}
void calculateRoots() {
if(d>0)
{
System.out.println("Roots are real and unequal");
root1 = ( - b + Math.sqrt(d))/(2*a);
root2 = (-b - Math.sqrt(d))/(2*a);
System.out.println("First root is:"+root1);
System.out.println("Second root is:"+root2);
}
else if(d == 0)
{
System.out.println("Roots are real and equal");
root1 = (-b+Math.sqrt(d))/(2*a);
System.out.println("Root:"+root1);
}
else
{
System.out.println("No real solutions. Roots are imaginary");
double real = -b / (2 * a);
```

```
double imaginary = Math.sqrt(-d) / (2 * a);
System.out.println("The equation has two complex roots: " + real + " + " + imaginary + "i and "
+ real + " - " + imaginary + "i");
}
}
}
public class QMain {
public static void main(String[] args) {
Quad q= new Quad();
q.input();
q.discriminant();
q.calculateRoots();
}
}
```

OUTPUT

```
C:\Users\Darshna Mimrot\OneDrive\Desktop\codes>javac QMain.java

C:\Users\Darshna Mimrot\OneDrive\Desktop\codes>java QMain
Quadratic equation is in the form : ax^2 + bx + c
Enter a: 1
Enter b:1
Enter c:1
No real solutions. Roots are imaginary
The equation has two complex roots: 0.0 + 0.86602540378443864i and 0.0 - 0.86602540378443864i

C:\Users\Darshna Mimrot\OneDrive\Desktop\codes>java QMain
Quadratic equation is in the form : ax^2 + bx + c
Enter a:1
Enter b:-2
Enter c:1
Roots are real and equal
Root:1.0

C:\Users\Darshna Mimrot\OneDrive\Desktop\codes>java QMain
Quadratic equation is in the form : ax^2 + bx + c
Enter a:1
Enter b:5
Enter c:2
Roots are real and unequal
First root is:-0.4384471871911697
Second root is:-4.561552812808831

C:\Users\Darshna Mimrot\OneDrive\Desktop\codes>java QMain
Quadratic equation is in the form : ax^2 + bx + c
Enter a:-1
Enter b:-2
Enter c:-5
No real solutions. Roots are imaginary
The equation has two complex roots: -1.0 + -2.0i and -1.0 - -2.0i

C:\Users\Darshna Mimrot\OneDrive\Desktop\codes>
```

Lab 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;
    double price;
    int num_pages;

    Books(String Name, String Author, double price, int num_pages) {
        this.Name = Name;
        this.Author = Author;
        this.num_pages = num_pages;
        this.price = price;
    }

    void setDetails() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter Name:");
        Name = s.nextLine();
        System.out.println("Enter Author Name:");
        Author = s.nextLine();
        System.out.println("Enter price:");
        price = s.nextDouble();
        System.out.println("Enter Number of pages:");
        num_pages = s.nextInt();
        System.out.println("----------");
    }

    void getDetails() {
        System.out.println("Book Name: " + Name);
        System.out.println("Author Name: " + Author);
        System.out.println("Price: " + price);
        System.out.println("Number of pages: " + num_pages);
    }
```

```
    public String toString() {
        return "Book Name: " + Name + "\nAuthor Name: " + Author + "\nPrice: " + price + "\nNumber
of pages: " + num_pages;
    }
}

class BookMain {
    public static void main(String args[]) {
        Scanner s = new Scanner(System.in);
        int n, i;
        System.out.println("Enter number of books:");
        n = s.nextInt();
        Books[] books = new Books[n];

        for (i = 0; i < n; i++) {
            System.out.println("Enter details of book " + (i + 1));
            books[i] = new Books("", "", 0.0, 0);
            books[i].setDetails();
        }

        for (i = 0; i < n; i++) {
            System.out.println("Details of book " + (i + 1));
            books[i].getDetails();
        }
    }
}
```

OUTPUT

```
C:\Users\Darshna Mimrot\OneDrive\Desktop\codes>java BookMain
Enter number of books:
2
Enter details of book 1
Enter Name:
It Ends With Us
Enter Author Name:
Collen Hoover
Enter price:
250
Enter Number of pages:
200
----------
Enter details of book 2
Enter Name:
Twilight Love
Enter Author Name:
Ana Hwang
Enter price:
380
Enter Number of pages:
400
----------
Details of book 1
Book Name: It Ends With Us
Author Name: Collen Hoover
Price: 250.0
Number of pages: 200
Details of book 2
Book Name: Twilight Love
Author Name: Ana Hwang
Price: 380.0
Number of pages: 400
```

Lab 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

```
abstract class Shape {
    // Two integers representing dimensions
    protected int dimension1;
    protected int dimension2;

    // Constructor
    public Shape(int dimension1, int dimension2) {
        this.dimension1 = dimension1;
        this.dimension2 = dimension2;
    }

    // Abstract method to be implemented by subclasses
    public abstract void printArea();
}

class Rectangle extends Shape {
    // Constructor
    public Rectangle(int length, int width) {
        super(length, width);
    }

    // Implementation of printArea for Rectangle
    @Override
    public void printArea() {
        int area = dimension1 * dimension2;
        System.out.println("Area of Rectangle: " + area);
    }
}

class Triangle extends Shape {
    // Constructor
    public Triangle(int base, int height) {
        super(base, height);
    }

    // Implementation of printArea for Triangle
    @Override
    public void printArea() {
        double area = 0.5 * dimension1 * dimension2;
```

```
        System.out.println("Area of Triangle: " + area);
    }
}

class Circle extends Shape {
    // Constructor
    public Circle(int radius) {
        super(radius, 0); // Only one dimension needed for a circle
    }

    // Implementation of printArea for Circle
    @Override
    public void printArea() {
        double area = Math.PI * dimension1 * dimension1;
        System.out.println("Area of Circle: " + area);
    }
}

public class Main {
    public static void main(String[] args) {
        // Creating objects of each shape
        Rectangle rectangle = new Rectangle(5, 10);
        Triangle triangle = new Triangle(8, 6);
        Circle circle = new Circle(4);

        // Printing areas
        rectangle.printArea();
        triangle.printArea();
        circle.printArea();
    }
}
```

OUTPUT

```
C:\Users\Darshna Mimrot\OneDrive\Desktop\codes>javac Main.java

C:\Users\Darshna Mimrot\OneDrive\Desktop\codes>java Main
Area of Rectangle: 50
Area of Triangle: 24.0
Area of Circle: 50.26548245743669
```

Lab 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;
    long accountNumber;ye
    String accountType;
    double balance;

    // Constructor
    public Account(String customerName, long accountNumber, String accountType, double
balance) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = balance;
    }

    // Method to accept deposit and update balance
    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposit successful. Updated balance: " + balance);
    }

    // Method to display balance
```

```
    public void displayBalance() {
        System.out.println("Balance for account " + accountNumber + ": " + balance);
    }

    // Method to compute and deposit interest
    public void depositInterest(double rate) {
        if ("Savings".equals(accountType)) {
            double interest = balance * (rate / 100);
            balance += interest;
            System.out.println("Interest deposited. Updated balance: " + balance);
        } else {
            System.out.println("No interest for current account.");
        }
    }

    // Method to permit withdrawal and update balance
    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawal successful. Updated balance: " + balance);
        } else {
            System.out.println("Insufficient funds for withdrawal.");
        }
    }
}

class SavAcct extends Account {
    // Constructor
    public SavAcct(String customerName, long accountNumber, double balance) {
        super(customerName, accountNumber, "Savings", balance);
    }
}

class CurrAcct extends Account {
    double minBalance;
    double serviceCharge;

    // Constructor
```

```
    public CurrAcct(String customerName, long accountNumber, double balance, double
minBalance, double serviceCharge) {
        super(customerName, accountNumber, "Current", balance);
        this.minBalance = minBalance;
        this.serviceCharge = serviceCharge;
    }

    // Override withdraw method to check for minimum balance
    @Override
    public void withdraw(double amount) {
        if (amount <= balance - minBalance) {
            balance -= amount;
            System.out.println("Withdrawal successful. Updated balance: " + balance);
        } else {
            System.out.println("Insufficient funds for withdrawal. Minimum balance requirement not
met.");
        }
    }

    // Method to check and impose penalty if minimum balance is not maintained
    public void checkMinimumBalance() {
        if (balance < minBalance) {
            balance -= serviceCharge;
            System.out.println("Service charge imposed for falling below minimum balance. Updated
balance: " + balance);
        }
    }
}

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

        // Creating savings account
        SavAcct savingsAccount = new SavAcct("John Doe", 123456789, 1000.0);

        // Creating current account with minimum balance and service charge
        CurrAcct currentAccount = new CurrAcct("Jane Doe", 987654321, 2000.0, 500.0, 50.0);
```

```
        // Performing operations on savings account
        System.out.println("Savings Account Operations:");
        savingsAccount.displayBalance();
        savingsAccount.deposit(500.0);
        savingsAccount.depositInterest(5.0);
        savingsAccount.withdraw(200.0);

        // Performing operations on current account
        System.out.println("\nCurrent Account Operations:");
        currentAccount.displayBalance();
        currentAccount.deposit(1000.0);
        currentAccount.withdraw(800.0);
        currentAccount.checkMinimumBalance();

        scanner.close();
    }
}
```

OUTPUT

```
C:\Users\Darshna Mimrot\OneDrive\Desktop\codes>java Bank
Savings Account Operations:
Balance for account 123456789: 1000.0
Deposit successful. Updated balance: 1500.0
Interest deposited. Updated balance: 1575.0
Withdrawal successful. Updated balance: 1375.0

Current Account Operations:
Balance for account 987654321: 2000.0
Deposit successful. Updated balance: 3000.0
Withdrawal successful. Updated balance: 2200.0
```

Lab 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

```
package CIE;

public class Student {
    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;
    }
}

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;
    }
}

package SEE;
import CIE.Student;
public class externals extends Student {
    public int[] seeMarks;

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

```
import CIE.Internals;
import SEE.externals;
import java.util.Scanner;

public class main {
    public static void main(String[] args) {
        System.out.println("Darshna Mimrot");
        System.out.println("1BM22CS081");
        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++) {
            System.out.println("Enter details for CIE of student " + (i + 1));
            System.out.print("USN: ");
            String usn = scanner.next();
            System.out.print("Name: ");
            String name = scanner.next();
             scanner.nextLine();
            System.out.print("Semester: ");
            int sem;
           while (true) {
        try {
            sem = Integer.parseInt(scanner.nextLine());
            break; // Exit the loop if the input is successfully converted to an integer
        } catch (NumberFormatException e) {
            System.out.println("Invalid input. Please enter a valid integer for Semester.");
        }
    }

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

```
            System.out.print("Semester: ");
            int sem;
            int[] seeMarks = new int[5];
            System.out.print("Enter SEE marks for 5 courses: ");
            for (int j = 0; j < 5; j++) {
                seeMarks[j] = scanner.nextInt();
            }
            seeStudents[i] = new externals(usn, name, sem, seeMarks);
        }

        // Displaying final marks
        System.out.println("\nFinal Marks of Students:");
        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

```
C:\Users\Darshna Mimrot\OneDrive\Desktop\codes>java main
Darshna Mimrot
1BM22CS081
Enter the number of students: 1
Enter details for CIE of student 1
USN: 1BM22CS081
Name: darshna.m
Semester: 3
Enter CIE marks for 5 courses: 33 35 40 28 20
Enter details for SEE of student 1
USN: 1BM22CS081
Name: darshna.m
Semester: 3
Enter SEE marks for 5 courses: 99 90 88 78 81

Final Marks of Students:

Details of Student 1
USN: 1BM22CS081
Name: darshna.m
Semester: 3
CIE Marks:
33 35 40 28 20
SEE Marks:
99 90 88 78 81
C:\Users\Darshna Mimrot\OneDrive\Desktop\codes>
```

Lab program 7

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age<0. In Son class, implement a constructor that cases both father and son’s age and throws an exception if son’s age is >=father’s age.

CODE

```
import java.util.Scanner;

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

class Father {
    int fatherAge;

    Father() throws WrongAge {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter Father’s Age:");
        fatherAge = s.nextInt();
        if (fatherAge < 0) {
            throw new WrongAge("Age cannot be negative"); // Fixed: 'Throw' to 'throw'
        }
    }

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

class Son extends Father {
    int sonAge;

    Son() throws WrongAge {
        super();
        Scanner s = new Scanner(System.in);
        System.out.println("Enter Son’s Age:");
        sonAge = s.nextInt();
```

```
        if (sonAge > fatherAge) {
            throw new WrongAge("Son’s age cannot be greater than Father’s age");
        } else if (sonAge == fatherAge) {
            throw new WrongAge("Son’s age cannot be equal to Father’s Age"); // Fixed: 'Throw' to
'throw'
        } else if (sonAge < 0) {
            throw new WrongAge("Age cannot be negative"); // Fixed: 'Throw' to 'throw'
        }
    }

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

public class Main {
    public static void main(String[] args) {
        try {
            Son s = new Son();
            s.display();
        } catch (WrongAge e) {
            System.out.println(e.getMessage());
        }
    }
}
```

OUTPUT

```
C:\Users\Darshna Mimrot\OneDrive\Desktop\codes>javac Main.java

C:\Users\Darshna Mimrot\OneDrive\Desktop\codes>java Main
Enter Father?s Age:
40
Enter Son?s Age:
18
Father Age is: 40
Son?s age is: 18

C:\Users\Darshna Mimrot\OneDrive\Desktop\codes>java Main
Enter Father?s Age:
30
Enter Son?s Age:
30
Son?s age cannot be equal to Father?s Age

C:\Users\Darshna Mimrot\OneDrive\Desktop\codes>java Main
Enter Father?s Age:
-20
Age cannot be negative
```

Lab 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 BMS implements Runnable {
    public void run() {
        while (true) {
            try {
                System.out.println("BMS college of engineering"); // Fixed: System with uppercase 'S'
                Thread.sleep(10000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

class CSE implements Runnable {
    public void run() {
        while (true) {
            try {
                System.out.println("CSE"); // Fixed: System with uppercase 'S'
                Thread.sleep(2000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

public class Main {
    public static void main(String[] args) {
        Thread t1 = new Thread(new BMS());
        Thread t2 = new Thread(new CSE());
        t1.start(); // Fixed: 'start' with lowercase 's'
        t2.start(); // Fixed: 'start' with lowercase 's'
    }
}
```

OUTPUT

```
C:\Users\Darshna Mimrot\OneDrive\Desktop\codes>java Main
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
```

THANK YOU