

**B.M.S. COLLEGE OF ENGINEERING**  
Basavanagudi, Bengaluru- 560019  
**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



**LABORATORY RECORD**

**ON**

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

Submitted By :  
**AYUSH KAPOOR**  
**1BM22CS064**

*In partial fulfilment of*  
**BACHELOR OF ENGINEERING**  
In  
**COMPUTER SCIENCE AND ENGINEERING**  
2023-24

Faculty-In-Charge  
**Shravya A R**  
Assistant Professor  
Department of Computer Science and Engineering

INDEX

Sl. No.	Date	Experiment Title
1	8-1-2024	Laboratory Program-1
2	8-01-2024	Laboratory Program-2
3	22-01-2024	Laboratory Program-3
4	22-01-2024	Laboratory Program-4
5	19-02-2024	Laboratory Program-5
6	19-02-2024	Laboratory Program-6
7	8-01-2024	Laboratory Program-7
8	29-01-2024	Laboratory Program-8

## LABORATORY 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.

```
import java.util.*;
```

```
class QuadraticEquationSolver {
```

```
    double a, b, c;
```

```
    QuadraticEquationSolver(double a, double b, double c) {
```

```
        this.a = a;
```

```
        this.b = b;
```

```
        this.c = c;
```

```
    }
```

```
    void takeInput() {
```

```
        Scanner scanner = new Scanner(System.in);
```

```
        System.out.println("Enter the coefficients (a, b, c) of the quadratic equation:");
```

```
        System.out.print("a: ");
```

```
        this.a = scanner.nextDouble();
```

```
        System.out.print("b: ");
```

```
        this.b = scanner.nextDouble();
```

```
        System.out.print("c: ");
```

```
        this.c = scanner.nextDouble();
```

```
}
```

```
void calculateRoots() {
```

```
    double discriminant = b * b - 4 * a * c;
```

```
    if (discriminant > 0) {
```

```
        double root1 = (-b + Math.sqrt(discriminant)) / (2 * a);
```

```
        double root2 = (-b - Math.sqrt(discriminant)) / (2 * a);
```

```
        System.out.println("Real Solutions:");
```

```
        System.out.println("Root 1: " + root1);
```

```
        System.out.println("Root 2: " + root2);
```

```
    } else if (discriminant == 0) {
```

```
        double root = -b / (2 * a);
```

```
        System.out.println("Real Solution:");
```

```
        System.out.println("Root: " + root);
```

```
    } else {
```

```
        System.out.println("No real solutions exist.");
```

```
    }
```

```
}
```

```
}
```

```
public class Quad {
```

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

```
        QuadraticEquationSolver solver = new QuadraticEquationSolver(0,  
0, 0);
```

```
        solver.takeInput();
```

```
        solver.calculateRoots();
```

```
    }
```

}

OUTPUT:

```
Enter the coefficients (a, b, c) of the quadratic equation:
```

```
a: 1
```

```
b: -3
```

```
c: 2
```

```
Real Solutions:
```

```
Root 1: 1.0
```

```
Root 2: 2.0
```

```
Enter the coefficients (a, b, c) of the quadratic equation:
```

```
a: 1
```

```
b: 0
```

```
c: -1
```

```
No real solutions exist.
```

## LABORATORY 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.

```
import java.util.*;

class Student {
    String USN;
    String name;
    int[] marks = new int[6]; // Assuming 6 subjects

    // Method to accept student details and marks
    void acceptDetails() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter USN:");
        USN = scanner.nextLine();
        System.out.println("Enter name:");
        name = scanner.nextLine();
        System.out.println("Enter marks for 6 subjects:");
        for (int i = 0; i < 6; i++) {
            System.out.print("Subject " + (i + 1) + " marks: ");
            marks[i] = scanner.nextInt();
        }
    }

    // Method to calculate percentage
    double calculatePercentage() {
        int totalMarks = 0;
        for (int mark : marks) {
            totalMarks += mark;
        }
        return (double) totalMarks / 6;
    }

    // Method to display student details
    void displayDetails() {
```

```
System.out.println("USN: " + USN);
System.out.println("Name: " + name);
System.out.println("Marks:");
for (int i = 0; i < 6; i++) {
    System.out.println("Subject " + (i + 1) + ": " + marks[i]);
}
System.out.println("Percentage: " + calculatePercentage() + "%");
}
}
```

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

        // Accept details for each student
        for (int i = 0; i < numStudents; i++) {
            System.out.println("Enter details for student " + (i + 1) + ":");
            students[i] = new Student();
            students[i].acceptDetails();
        }

        // Display details for each student
        System.out.println("\nDetails of all students:");
        for (int i = 0; i < numStudents; i++) {
            System.out.println("\nDetails of student " + (i + 1) + ":");
            students[i].displayDetails();
        }
    }
}
```

## OUTPUT:

```
Enter the number of students: 2
Enter details for student 1:
Enter USN: 123456789
Enter name: John Doe
Enter marks for 6 subjects: 90 80 70 60 50 40
Enter details for student 2:
Enter USN: 987654321
Enter name: Jane Doe
Enter marks for 6 subjects: 80 90 100 70 60 50
```

Details of all students:

Details of student 1:

USN: 123456789

Name: John Doe

Marks:

Subject 1: 90

Subject 2: 80

Subject 3: 70

Subject 4: 60

Subject 5: 50

Subject 6: 40

Percentage: 66.66666666666666

Details of student 2:

USN: 987654321

Name: Jane Doe

Marks:

Subject 1: 80

Subject 2: 90

Subject 3: 100

Subject 4: 70

Subject 5: 60

Subject 6: 50

Percentage: 75.0



## LABORATORY PROGRAM - 3

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.

```
abstract class Shape {  
    protected int dimension1;  
    protected int dimension2;  
  
    public Shape(int dimension1, int dimension2) {  
        this.dimension1 = dimension1;  
        this.dimension2 = dimension2;  
    }  
  
    public abstract void printArea();  
}  
  
class Rectangle extends Shape {  
    public Rectangle(int length, int width) {  
        super(length, width);  
    }  
  
    @Override  
    public void printArea() {  
        int area = dimension1 * dimension2;  
        System.out.println("Area of Rectangle: " + area);  
    }  
}
```

```
}  
}  
  
class Triangle extends Shape {  
    public Triangle(int base, int height) {  
        super(base, height);  
    }  
  
    @Override  
    public void printArea() {  
        double area = 0.5 * dimension1 * dimension2;  
        System.out.println("Area of Triangle: " + area);  
    }  
}
```

```
class Circle extends Shape {  
    public Circle(int radius) {  
        super(radius, 0);  
    }  
  
    @Override  
    public void printArea() {  
        double area = Math.PI * dimension1 * dimension1;  
        System.out.println("Area of Circle: " + area);  
    }  
}
```

```
public class ShapeResult {  
    public static void main(String[] args) {  
        Rectangle rectangle = new Rectangle(5, 10);  
        rectangle.printArea();  
  
        Triangle triangle = new Triangle(4, 6);  
        triangle.printArea();  
  
        Circle circle = new Circle(3);  
        circle.printArea();  
    }  
}
```

OUTPUT:

```
Area of Rectangle: 50  
Area of Triangle: 12.0  
Area of Circle: 28.274333882308138
```

## LABORATORY PROGRAM – 4

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

```
import java.util.Scanner;
class account{
    boolean check;
    String name;
    long accno;
    String acctype;
    double bal;
    Scanner sc = new Scanner(System.in);
    account(String name, long accno, String acctype, double bal, boolean
cheque)
    {
        this.name = name;
        this.accno=accno;
        this.acctype= acctype;
        this.bal= bal;
        this.check = cheque;
```

```

}

void Dep (){
    System.out.println("Enter amount to be deposited");
    double amt=sc.nextDouble();
    bal += amt;
}
void Displaybal() {
    System.out.println("Balance: " + bal);
}
public void interest()
{
    System.out.println("Enter rate: ");
    Double rate=sc.nextDouble();
    if ("Savings". equals (acctype))
    {
        double interest=bal*(rate/100);
        bal += interest;
    }
}
public void withdraw () {
    System.out.println("Enter amount the be withdrawn");
    double amt=sc.nextDouble();
    if (amt <= bal){
        bal-=amt;
    }
    else{
        System.out.println("Insufficient funds");
    }
}
}

class savacc extends account{
    public savacc(String name, long accno, double bal, boolean cheque)
    {
        super(name, accno, "Savings", bal, cheque);
    }
}

```

```

}
}

class curracct extends account{
    double minbal;
    double serv;
    public curracct(String name, long accno, double bal, double minbal,
double serv, boolean cheque)
    {
        super(name, accno, "current", bal, cheque);
        this.minbal=minbal;
        this.serv=serv;
    }
    public void withdraw() {
        System.out.println("Enter amt to be withdrawn");
        double amt=sc.nextDouble();
        if (amt<= bal-minbal) {
            bal-=amt;
        }
        else{
            System.out.println("Insufficient funds");
        }
    }
    public void check() {
        if (bal > minbal) {
            bal -= serv;
            System.out.println("Service charges applied");
        }
    }
}

class bank{

    public static void main(String[] args) {
        System.out.println("USN:1BM22CS066");
        System.out.println("Name: Bhanoday Kurma");
    }
}

```

```
savacc s = new savacc ("joseph",123456,10000.0,false);
curracct c= new curracct ("John Mike",938462,20000.0,500.0,
50.0,true);
    System.out.println("Savings Acc");
    s.Displaybal();
    s.Dep();
    s.Displaybal();
    s.interest();
    s.Displaybal();
    s.withdraw();
    s.Displaybal();
    System.out.println("current acc");
    c.Displaybal();
    c.Dep();
    c.Displaybal();
    c.withdraw();
    c.Displaybal();
    c.check();
    c.Displaybal();
}
}
```

OUTPUT :

USN:1BM22CS064

Name: Ayush Kapoor

Savings Acc

Balance: 10000.0

Enter amount to be deposited

5000.0

Balance: 15000.0

Enter rate:

3.5

Balance: 15525.0

Enter amount the be withdrawn

2000.0

Balance: 13525.0

current acc

Balance: 20000.0

Enter amt to be withdrawn

10000.0

Balance: 10000.0

Enter amount the be withdrawn

5000.0

Insufficient funds

Balance: 10000.0

Service charges applied

Balance: 9950.0



## LABORATORY PROGRAM - 5

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.

```
class MessageThread extends Thread {
    private String message;
    private int interval;

    public MessageThread(String message, int interval) {
        this.message = message;
        this.interval = interval;
    }

    public void run() {
        try {
            while (true) {
                System.out.println(message);
                Thread.sleep(interval);
            }
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}

public class Main {
```

```
public static void main(String[] args) {  
    // Creating and starting threads  
    Thread thread1 = new MessageThread("BMS College of  
Engineering", 10000); // 10 seconds  
    Thread thread2 = new MessageThread("CSE", 2000); // 2 seconds  
    thread1.start();  
    thread2.start();  
}  
}
```

OUTPUT:

```
C:\Users\bhano\OneDrive\Desktop\Ayush(1BM22CS064) >java T1  
CSE  
BMS College of Engineering  
CSE  
CSE  
CSE  
CSE  
BMS College of Engineering  
CSE  
CSE  
CSE  
CSE  
CSE  
BMS College of Engineering  
CSE  
CSE  
CSE  
CSE  
CSE  
BMS College of Engineering  
CSE  
CSE  
CSE  
CSE  
BMS College of Engineering  
CSE
```

## LABORATORY PROGRAM - 6

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.

// Custom Exception Class

```
class WrongAge extends Exception {  
    public WrongAge() {  
        super("Age cannot be negative.");  
    }  
}
```

// Base class

```
class Father {  
    int age;  
  
    public Father(int age) throws WrongAge {  
        if (age < 0) {  
            throw new WrongAge();  
        }  
        this.age = age;  
    }  
}
```

// Derived class

```
class Son extends Father {  
    int sonAge;
```

```
public Son(int fatherAge, int sonAge) throws WrongAge {  
    super(fatherAge);  
    if (sonAge >= fatherAge) {  
        throw new WrongAge();  
    }  
    this.sonAge = sonAge;  
}  
}
```

// Main class

```
public class Main {  
    public static void main(String[] args) {  
        try {  
            Father father = new Father(50);  
            System.out.println("Father's age: " + father.age);  
  
            // This will throw an exception because son's age is greater than or equal to  
            father's age  
            Son son = new Son(50, 55);  
            System.out.println("Son's age: " + son.sonAge);  
        } catch (WrongAge e) {  
            System.out.println("Error: " + e.getMessage());  
        }  
    }  
}
```

OUTPUT:

```
C:\Users\bhano\OneDrive\Desktop>java Main  
Father's age: 50  
Error: Age cannot be negative.
```

## LABORATORY PROGRAM - 7

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.

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

```
class Book {  
    private String name;  
    private String author;  
    private double price;  
    private int num_pages;  
  
    public Book(String name, String author, double price, int num_pages) {  
        this.name = name;  
        this.author = author;  
        this.price = price;  
        this.num_pages = num_pages;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public String getAuthor() {  
        return author;  
    }  
  
    public double getPrice() {  
        return price;  
    }  
  
    public int getNumPages() {  
        return num_pages;  
    }  
  
    // Setter methods
```

```
public void setName(String name) {
    this.name = name;
}

public void setAuthor(String author) {
    this.author = author;
}

public void setPrice(double price) {
    this.price = price;
}

public void setNumPages(int num_pages) {
    this.num_pages = num_pages;
}

@Override
public String toString() {
    return "Book Details: \nName: " + name + "\nAuthor: " + author + "\nPrice: $"
+ price + "\nNumber of Pages: " + num_pages;
}
}

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

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

        Book[] books = new Book[n];

        for (int i = 0; i < n; i++) {
            System.out.println("\nEnter details for Book " + (i + 1) + ":");
            scanner.nextLine(); // Consume newline
            System.out.print("Name: ");
            String name = scanner.nextLine();
            System.out.print("Author: ");
            String author = scanner.nextLine();
            System.out.print("Price: ");
            double price = scanner.nextDouble();
        }
    }
}
```

```
System.out.print("Number of Pages: ");  
int num_pages = scanner.nextInt();
```

```
    books[i] = new Book(name, author, price, num_pages);  
}
```

```
System.out.println("\nDetails of all books:");  
for (int i = 0; i < n; i++) {  
    System.out.println("\nBook " + (i + 1) + ":");  
    System.out.println(books[i]);  
}
```

```
scanner.close();  
}  
}
```



OUTPUT:

Enter the number of books: 2

Enter details for Book 1:

Name: The Great Gatsby

Author: F. Scott Fitzgerald

Price: 8.99

Number of Pages: 180

Enter details for Book 2:

Name: To Kill a Mockingbird

Author: Harper Lee

Price: 9.99

Number of Pages: 281

Details of all books:

Book 1:

Book Details:

Name: The Great Gatsby

Author: F. Scott Fitzgerald

Price: \$8.99

Number of Pages: 180

Book 2:

Book Details:

Name: To Kill a Mockingbird

Author: Harper Lee

Price: \$9.99

Number of Pages: 281

## LABORATORY PROGRAM - 8

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

```
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 {
```

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

```
    public Internals(int[] internalMarks) {
```

```
        this.internalMarks = internalMarks;
    }
}

package SEE;
import CIE.Student;

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

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

import CIE.Student;
import CIE.Internals;
import SEE.External;

import java.util.Scanner;

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

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

```
Student[] students = new Student[n];
int[][] finalMarks = new int[n][5];
for (int i = 0; i < n; i++) {
    System.out.println("\nEnter details for Student " + (i + 1) + ":");
    scanner.nextLine(); // Consume newline
    System.out.print("USN: ");
    String usn = scanner.nextLine();
    System.out.print("Name: ");
    String name = scanner.nextLine();
    System.out.print("Semester: ");
    int sem = scanner.nextInt();

    System.out.println("Enter Internal marks for Student " + (i + 1) +
":");
    int[] internalMarks = new int[5];
    for (int j = 0; j < 5; j++) {
        System.out.print("Internal marks for Course " + (j + 1) + ": ");
        internalMarks[j] = scanner.nextInt();
    }

    System.out.println("Enter External marks for Student " + (i + 1) +
":");
    int[] externalMarks = new int[5];
    for (int j = 0; j < 5; j++) {
        System.out.print("External marks for Course " + (j + 1) + ": ");
        externalMarks[j] = scanner.nextInt();
    }
}
```

```
students[i] = new External(usn, name, sem, externalMarks);
```

```
for (int j = 0; j < 5; j++) {
```

```
    finalMarks[i][j] = internalMarks[j] + externalMarks[j];
```

```
}
```

```
}
```

```
System.out.println("\nFinal Marks of all Students:");
```

```
for (int i = 0; i < n; i++) {
```

```
    System.out.println("\nStudent " + (i + 1) + " - USN: " +  
students[i].usn + ", Name: " + students[i].name + ", Semester: " +  
students[i].sem);
```

```
    System.out.println("Final Marks:");
```

```
    for (int j = 0; j < 5; j++) {
```

```
        System.out.println("Course " + (j + 1) + ": " + finalMarks[i][j]);
```

```
    }
```

```
}
```

```
scanner.close();
```

```
}
```

```
}
```

## OUTPUT:

Enter the number of students: 3

Enter details for Student 1:

USN: 1BM22IS089

Name: Naruto Uzumaki

Semester: 3

Enter Internal marks for Student 1:

Internal marks for Course 1: 45

Internal marks for Course 2: 46

Internal marks for Course 3: 41

Internal marks for Course 4: 38

Internal marks for Course 5: 40

Enter External marks for Student 1:

External marks for Course 1: 90

External marks for Course 2: 98

External marks for Course 3: 96

External marks for Course 4: 80

External marks for Course 5: 86

Enter details for Student 2:

USN: 1BM22ME123

Name: Sasuke Uchicha

Semester: 4

Enter Internal marks for Student 2:

Internal marks for Course 1: 43

Internal marks for Course 2: 42

Internal marks for Course 3: 48

Internal marks for Course 4: 47

Internal marks for Course 5: 49

Enter External marks for Student 2:  
External marks for Course 1: 98  
External marks for Course 2: 97  
External marks for Course 3: 91  
External marks for Course 4: 92  
External marks for Course 5: 99

Enter details for Student 3:  
USN: 1BM22IS081  
Name: Sakura Haruno  
Semester: 3

Enter Internal marks for Student 3:  
Internal marks for Course 1: 43  
Internal marks for Course 2: 44  
Internal marks for Course 3: 48  
Internal marks for Course 4: 50  
Internal marks for Course 5: 49  
Enter External marks for Student 3:  
External marks for Course 1: 99  
External marks for Course 2: 93  
External marks for Course 3: 91  
External marks for Course 4: 97  
External marks for Course 5: 100

## Final Marks of all Students:

Student 1 – USN: 1BM22CS077, Name: Naruto Uzumaki, Semester: 3

Final Marks:

Course 1: 88

Course 2: 85

Course 3: 77

Course 4: 84

Course 5: 96

Student 2 – USN: 1BM22IS061, Name: Sasuke Uchicha, Semester: 3

Final Marks:

Course 1: 88

Course 2: 94

Course 3: 98

Course 4: 82

Course 5: 82

Student 3 – USN: 1BM22CS089, Name: Sakura Haruno, Semester: 3

Final Marks:

Course 1: 86

Course 2: 93

Course 3: 96

Course 4: 94

Course 5: 80