GOVERNMENT OF KERALA

DEPARTMENT OF TECHNICAL EDUCATION

RAJIV GANDHI INSTITUTE OF TECHNOLOGY

(GOVT. ENGINEERING COLLEGE)

KOTTAYAM - 686501

RECORD BOOK

GOVERNMENT OF KERALA

DEPARTMENT OF TECHNICAL EDUCATION

RAJIV GANDHI INSTITUTE OF TECHNOLOGY

(GOVT. ENGINEERING COLLEGE)

KOTTAYAM - 686501

20MCA132
OBJECT ORIENTED PROGRAMMING LAB

Name: JOEL S RAPHAEL

Branch: Master of Computer Applications

Semester: 2

Roll No: 30

CERTIFIED BONAFIDE RECORD WORK DONE BY

Reg No. ....

STAFF IN CHARGE

INTERNAL EXAMINER **EXTERNAL EXAMINER**

Contents

Experiment No: 1 **Date: 11/02/2025**

Even-Odd Classification

Aim

Write a Java program to check whether an input number is even or odd.

Algorithm

```
1. Start
2. Read the input.
3. Use an if-else statement to check if the number is even or odd.
4. Print the result.
5. Stop
```

Source Code

```java
import java.util.Scanner;

public class EvenOdd{
        public static  void main(String[] args){
                Scanner scanner = new Scanner(System.in);
                System.out.print("Enter a number: ");
                int number = scanner.nextInt();

                if(number % 2 == 0){
                        System.out.println(number + " is even.");
                }else{
                        System.out.println(number + " is odd.");
                }
}
}
```

Result

The program was executed successfully.
When the input ”5” was provided, the output was: ”5 is odd”

```
Enter a number: 5
5 is odd.
```

Experiment No: 2 **Date: 11/02/2025**

Sum of First n Natural Numbers

Aim

Write a Java program to compute the sum of the first n natural numbers.

Algorithm

```
1.Start
2.Declare an integer variable n and sum initialized to 0.
3.Display the message "Enter n:" to prompt the user to enter a number.
4.Read the integer value n from the user input.
    For i = 1 to i = n:

    Add i to sum (i.e., sum = sum + i).

5.After the loop ends, display the value of sum as the result.
6.End
```

Source Code

```
import java.util.Scanner;
public class naturalnos
{
        public static void main(String arg[]){
                Scanner s=new Scanner(System.in);
                System.out.print("Enter n:");
                int n=s.nextInt();
    int sum=0;
                for(int i=1;i<=n;i++)
                        sum=sum+i;
                System.out.print("Sum="+sum);
        }
}
```

Result

The program was executed successfully.
When the input ”5” was provided, the output was: ”15”

```
Enter n:5
Sum=15
```

Experiment No: 3 **Date: 11/02/2025**

Factorial of a Number

Aim

Write a Java program to compute the factorial of a given number.

Algorithm

```
1. Start
2. Take an integer as input from the user.
3. Compute the factorial using either a for loop or a while loop.
4. Print the result.
5. Stop
```

Source Code

```java
import java.util.Scanner;
public class factorial
{
        public static void main(String arg[]){
                Scanner s=new Scanner(System.in);
                System.out.print("Enter the number:");
                int n=s.nextInt();
                int fact=1;
                for(int i=1;i<=n;i++)
                        fact=fact*i;
                System.out.print("Factorial="+fact);
        }
}
```

Result

The program was executed successfully.
When the input ”5” was provided, the output was: ”120”

```
Enter the number:5
Factorial=120
```

Experiment No: 4 **Date: 11/02/2025**

Assigning Grades Based on Numeric Score

Aim

Write a Java program that assigns a grade based on a numeric score.

Algorithm

```
1. Start
2. Take a numeric score (0{100) as input from the user.
3. Use either an if-else if-else structure or a switch-case statement to assign a
    grade:
       • 90{100 → A
       • 80{89 → B
       • 70{79 → C
       • 60{69 → D
       • Below 60 → F
4. Print the assigned grade.
5. Stop
```

Source Code

```java
import java.util.Scanner;
public class grade{
        public static void main(String arg[]){
                Scanner s=new Scanner(System.in);
                System.out.print("Enter the mark:");
                int m=s.nextInt();
                if(m>=90){
                        System.out.print("Grade A");
                }
                else if(m>=80){
                        System.out.print("Grade B");
                }
    else if(m>=70){
                        System.out.print("Grade C");
                }
    else if(m>=60){
                        System.out.print("Grade D");
                }
    else{
                        System.out.print("Grade F");
                }
        }
}
```

Result

The program was executed successfully.
When the input ”67” was provided, the output was: ”Grade D”

```
Enter the mark:67
Grade D
```

Experiment No: 5 **Date: 11/02/2025**

Find Product with Lowest Price

Aim

Write a Java program to define a class Product with data members pcode, pname, and price. Find and display the product with the lowest price.

Algorithm

```
1. Start
2. Define a class Product with attributes pcode, pname, and price.
3. Create a function findLowest to compare product prices and return the lowest.
4. Read details of three products from the user.
5. Call findLowest and display the product with the lowest price.
6. Stop
```

Source Code

```java
import java.util.Scanner;
public class Product{
        String pcode,pname;
        double price;
        public Product(String pcode, String pname, double price){
                this.pcode=pcode;
                this.pname=pname;
                this.price=price;
        }
        public static Product getLowestPrice(Product items[]){
                Product lowest=items[0];
                for(int i=1;i<items.length;i++){
                        if(items[i].price<lowest.price) lowest=items[i];
                }
                return lowest;
        }
        public void display(){
                System.out.println("Product code: "+this.pcode+"\nProduct name: "+this.pname+"\nPrice: "+price);
        }
        public static void main(String args[]){
                Scanner s=new Scanner(System.in);
                System.out.println("Enter number of products: ");
                int n=s.nextInt();
                Product items[]=new Product[n];
                for(int i=0;i<n;i++){
                        System.out.println("Enter product code of product "+i+": ");
                        String pcode=s.next();
                        System.out.println("Enter product name of product "+i+": ");
                        String pname=s.next();
                        System.out.println("Enter price of product "+i+
   ": ");
                        double price=s.nextDouble();
                        items[i]= new Product(pcode,pname,price);
                }
                Product lowest=Product.getLowestPrice(items);
                System.out.println("Item with lowest Price : ");
                lowest.display();
        }
}
```

Result

```
Enter number of products: 3
Enter product code of product 0: 1
Enter product name of product 0: laptop
Enter price of product 0: 6000
Enter product code of product 1: 2
Enter product name of product 1: computer
Enter price of product 1: 34500
Enter product code of product 2: 3
Enter product name of product 2: tv
Enter price of product 2: 17000
Item with lowest Price :
Product code: 1
Product name: laptop
Price: 6000.0
```

Experiment No: 6 **Date: 11/02/2025**

Complex Number Operations

Aim

Write a Java program to perform addition and multiplication of complex numbers, with inputs provided by the user.

Algorithm

```
1. Start
2. Define a class Complex with attributes real and imag.
3. Implement methods add and multiply to perform operations on complex numbers.
4. Read two complex numbers from the user.
5. Compute their sum and product using the respective methods.
6. Display the results.
7. Stop.
```

Source Code

```java
import java.util.Scanner;

class Complex {import java.util.Scanner;

class Complex {
    double real, imaginary;

    Complex(double real, double imaginary) {
        this.real = real;
        this.imaginary = imaginary;
    }


    Complex add(Complex c) {
        return new Complex(this.real + c.real, this.imaginary + c.imaginary);
    }

    Complex multiply(Complex c) {
        double realPart = this.real * c.real - this.imaginary * c.imaginary;
        double imaginaryPart = this.real * c.imaginary + this.imaginary * c.real;
        return new Complex(realPart, imaginaryPart);
    }


    void display() {
        if (imaginary < 0) {
            System.out.println(real + " - " + Math.abs(imaginary) + "i");
        } else {
            System.out.println(real + " + " + imaginary + "i");
        }
    }
}

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

        System.out.println("Enter the real and imaginary parts of the first complex number:");
        System.out.print("Real: ");
        double real1 = sc.nextDouble();
        System.out.print("Imaginary: ");
        double imaginary1 = sc.nextDouble();


        System.out.println("Enter the real and imaginary parts of the second complex number:");
        System.out.print("Real: ");
        double real2 = sc.nextDouble();
        System.out.print("Imaginary: ");
        double imaginary2 = sc.nextDouble();


        Complex c1 = new Complex(real1, imaginary1);
        Complex c2 = new Complex(real2, imaginary2);


        Complex additionResult = c1.add(c2);
        System.out.print("Addition of the two complex numbers: ");
        additionResult.display();

        Complex multiplicationResult = c1.multiply(c2);
        System.out.print("Multiplication of the two complex numbers: ");
        multiplicationResult.display();
    }
}
```

Result

```
Enter the real and imaginary parts of the first complex number:
Real: 5
Imaginary: 2
Enter the real and imaginary parts of the second complex number:
Real: 3
Imaginary: 4
Addition of the two complex numbers: 8.0 + 6.0i
Multiplication of the two complex numbers: 7.0 + 26.0i
```

Experiment No: 7 **Date: 11/02/2025**

Matrix Additions

Aim

Write a Java program to perform matrix addition.

Algorithm

```
1. Start
2. Read the number of rows and columns of the matrices.
3. Read elements of the first matrix.
4. Read elements of the second matrix.
5. Perform element-wise addition to obtain the sum matrix.
6. Display the sum matrix.
7. Stop
```

Source Code

```java
import java.util.Scanner;

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


        System.out.print("Enter number of rows: ");
        int rows = sc.nextInt();
        System.out.print("Enter number of columns: ");
        int cols = sc.nextInt();

        int[][] matrix1 = new int[rows][cols];
        int[][] matrix2 = new intWrite a Java program to perform matrix
  addition.[rows][cols];
        int[][] sumMatrix = new int[rows][cols];

        System.out.println("Enter elements of the first matrix:");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                matrix1[i][j] = sc.nextInt();
            }
        }
        System.out.println("Enter elements of the second matrix:");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                matrix2[i][j] = sc.nextInt();
            }
        }

        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                sumMatrix[i][j] = matrix1[i][j] + matrix2[i][j];
            }
        }

        System.out.println("Sum of the two matrices:");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                System.out.print(sumMatrix[i][j] + " ");
            }
            System.out.println();
        }

        sc.close();
    }
}
```

Result

```
Enter number of rows: 2
Enter number of columns: 2
Enter elements of the first matrix:
1 2
3 4
Enter elements of the second matrix:
4 5
6 7
Sum of the two matrices:
5 7
9 11
```

Experiment No: 8 **Date: 25/02/2025**

Employee Search Using an Array of Objects

Aim

Write a Java program to store employee details including employee number, name, and salary, and search for an employee by employee number.

Algorithm

```
1. Start
2.Input the number of employees.
3.Store each employee's number, name, and salary in a list.
4.Input an employee number to search.
5.Search the list for the employee number.
6.Display the employee details if found; otherwise, print "Employee not found."
7. Stop
```

Source Code

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

class Employee {
    int empNumber;
    String empName;
    double empSalary;

    Employee(int empNumber, String empName, double empSalary) {
        this.empNumber = empNumber;
        this.empName = empName;
        this.empSalary = empSalary;
    }

    void displayEmployeeDetails() {
        System.out.println("Employee Number: " + empNumber);
        System.out.println("Employee Name: " + empName);
        System.out.println("Employee Salary: " + empSalary);
    }
}

public class EmployeeDetails {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        ArrayList<Employee> employeeList = new ArrayList<>();

        System.out.print("Enter the number of employees: ");
        int numberOfEmployees = scanner.nextInt();
        scanner.nextLine();

        for (int i = 0; i < numberOfEmployees; i++) {
            System.out.println("\nEnter details for employee " + (i +
  1));
```

```java
            System.out.print("Enter employee number: ");
            int empNumber = scanner.nextInt();
            scanner.nextLine();
            System.out.print("Enter employee name: ");
            String empName = scanner.nextLine();
            System.out.print("Enter employee salary: ");
            double empSalary = scanner.nextDouble();
            scanner.nextLine();

            employeeList.add(new Employee(empNumber, empName, empSalary));
        }

        System.out.print("\nEnter employee number to search: ");
        int empNumberToSearch = scanner.nextInt();
        boolean found = false;

        for (Employee emp : employeeList) {
            if (emp.empNumber == empNumberToSearch) {
                emp.displayEmployeeDetails();
                found = true;
                break;
            }
        }

        if (!found) {
            System.out.println("Employee not found with employee number: " + empNumberToSearch);
        }

        scanner.close();
    }
}
```

Result

```
Enter the number of employees: 2

Enter details for employee 1
Enter employee number: 23
Enter employee name: james
Enter employee salary: 10000

Enter details for employee 2
Enter employee number: 24
Enter employee name: jafar
Enter employee salary: 12999

Enter employee number to search: 2
Employee not found with employee number: 2
```

Experiment No: 9 **Date: 25/02/2025**

String Search in an Array

Aim

Write a Java program to store ‘n‘ strings in an array. Search for a given string. If found, print its index; otherwise, display ”String not found.

Algorithm

```
1. Start
2.Input the number of strings (n).
3.Create an array of strings to store n strings.
4.Input the n strings from the user.
5.Input the string to search (searchString).
6.Search the array for searchString:
    If found, display the index of the string.
    If not found, display "String not found."
7. Stop
```

Source Code

```java
import java.util.Scanner;

public class StringSearch {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of strings you want to store
    : ");
        int n = scanner.nextInt();
        scanner.nextLine();
        String[] strings = new String[n];
        System.out.println("Enter the strings:");
        for (int i = 0; i < n; i++) {
            System.out.print("String " + (i + 1) + ": ");
            strings[i] = scanner.nextLine();
        }
        System.out.print("\nEnter the string to search: ");
        String searchString = scanner.nextLine();
        boolean found = false;
        for (int i = 0; i < n; i++) {
            if (strings[i].equals(searchString)) {
                System.out.println("String found at index: " + i);
                found = true;
                break;
            }
        }
        if (!found) {
            System.out.println("String not found.");
        }
        scanner.close();
    }
}
```

Result

```
Enter the number of strings you want to store: 3
Enter the strings:
String 1: hello
String 2: ma
String 3: boy

Enter the string to search: 2
String not found.
```

Experiment No: 10 **Date: 25/02/2025**

String Manipulations

Aim

Write a Java program to perform various string manipulations, including finding the length, converting to uppercase and lowercase, extracting characters and substrings, and reversing the string.

Algorithm

```
1. Start
2.Input a string from the user.
3.Calculate and print the length of the string.
4.Convert the string to uppercase and print it.
5.Convert the string to lowercase and print it.
6.Get and print the first character of the string.
7.Extract and print a substring from index 2 to 5.
8.Reverse the string and print it.
9. Stop
```

Source Code

```java
import java.util.Scanner;

public class StringManipulations {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String inputString = scanner.nextLine();

        int length = inputString.length();
        System.out.println("Length of the string: " + length);

        String upperCaseString = inputString.toUpperCase();
        System.out.println("String in uppercase: " + upperCaseString);

        String lowerCaseString = inputString.toLowerCase();
        System.out.println("String in lowercase: " + lowerCaseString);

        char firstChar = inputString.charAt(0);
        System.out.println("First character: " + firstChar);

        String substring = inputString.substring(2, 5);
        System.out.println("Substring from index 2 to 5: " + substring);

        String reversedString = new StringBuilder(inputString).reverse().toString();
        System.out.println("Reversed string: " + reversedString);

        scanner.close();
        }
}
```

Result

```
Enter a string: hello
Length of the string: 5
String in uppercase: HELLO
String in lowercase: hello
First character: h
Substring from index 2 to 5: llo
Reversed string: olleh
```

Experiment No: 11 **Date: 25/02/2025**

Inheritance in Java

Aim

Write a Java program to implement hierarchical inheritance for a book management system. Define a base class ‘Publisher‘, a derived class ‘Book‘, and two subclasses ‘Literature‘ and ‘Fiction‘. Include methods to read and display book details and demonstrate the functionality using user input.

Algorithm

```
1. Start
2.Input the details for a Literature book:
    Publisher name
    Book title
    Author name
    Genre
3.Input the details for a Fiction book:
    Publisher name
    Book title
    Author name
    Category
4.Create objects of the Literature and Fiction classes using the input data.
5.Display the details for the Literature book:
    Publisher, book title, author, and genre.
6.Display the details for the Fiction book:
    Publisher, book title, author, and category.
7. Stop
```

Source Code

```java
import java.util.Scanner;

// Base class: Publisher
class Publisher {
    String publisherName;

    // Constructor
    Publisher(String publisherName) {
        this.publisherName = publisherName;
    }

    // Method to display publisher details
    void displayPublisher() {
        System.out.println("Publisher: " + publisherName);
    }
}

// Derived class: Book (Inherits Publisher)
```

```java
class Book extends Publisher {
    String bookTitle;
    String authorName;

    // Constructor
    Book(String publisherName, String bookTitle, String authorName) {
        super(publisherName); // Call to Publisher class constructor
        this.bookTitle = bookTitle;
        this.authorName = authorName;
    }

    // Method to display book details
    void displayBook() {
        displayPublisher();
        System.out.println("Book Title: " + bookTitle);
        System.out.println("Author: " + authorName);
    }
}

// Subclass 1: Literature (Inherits Book)
class Literature extends Book {
    String genre;

    Literature(String publisherName, String bookTitle, String authorName, String genre) {
        super(publisherName, bookTitle, authorName);
        this.genre = genre;
    }

    void display() {
        System.out.println("\n[Literature Book Details]");
        displayBook();
        System.out.println("Genre: " + genre);
    }
}

// Subclass 2: Fiction (Inherits Book)
class Fiction extends Book {
    String category;

    Fiction(String publisherName, String bookTitle, String authorName, String category) {
        super(publisherName, bookTitle, authorName);
        this.category = category;
    }

    void display() {
        System.out.println("\n[Fiction Book Details]");
        displayBook();
        System.out.println("Category: " + category);
    }
}

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

        // Get Literature book details
        System.out.println("Enter details for Literature book:");
        System.out.print("Publisher Name: ");
        String pub1 = sc.nextLine();
        System.out.print("Book Title: ");
        String title1 = sc.nextLine();
        System.out.print("Author Name: ");
        String author1 = sc.nextLine();
        System.out.print("Genre: ");
        String genre = sc.nextLine();

        // Get Fiction book details
        System.out.println("\nEnter details for Fiction book:");
        System.out.print("Publisher Name: ");
        String pub2 = sc.nextLine();
        System.out.print("Book Title: ");
        String title2 = sc.nextLine();
        System.out.print("Author Name: ");
        String author2 = sc.nextLine();
        System.out.print("Category: ");
        String category = sc.nextLine();

        // Create objects
        Literature litBook = new Literature(pub1, title1, author1, genre);
        Fiction ficBook = new Fiction(pub2, title2, author2, category);

        // Display book details
        litBook.display();
        ficBook.display();

        sc.close();
    }
}
```

Result

```
Enter details for Literature book:
Publisher Name: joel
Book Title: THE TRUTH
Author Name: james
Genre: murder

Enter details for Fiction book:
Publisher Name: rambo
Book Title: SEVEN
Author Name: david
Category: comedy

[Literature Book Details]
Publisher: joel
Book Title: THE TRUTH
Author: james
Genre: murder

[Fiction Book Details]
Publisher: rambo
Book Title: SEVEN
Author: david
Category: comedy
```