

**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: ANAMIKA M S**

**Branch: Master of Computer Applications**

**Semester: 2**

**Roll No: 11**

**CERTIFIED BONAFIDE RECORD WORK DONE BY**  
**Reg No. ....**

**STAFF IN CHARGE**

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

# Contents

1. Even-Odd Classification	2
2. Sum of First n Natural Numbers	3
3. Factorial of a Number	4
4. Assigning Grades Based on Numeric Score	5
5. Find Product with Lowest Price	7
6. Complex Number Operations	9
7. Matrix Addition	11
8. Employee Search Using an Array of Objects	13
9. String Search in an Array	15
10. String Manipulations	17
11. Inheritance in Java	19

## Even-Odd Classification

### Aim

Create an object-oriented Java program to check whether an input number is even or odd.

### Algorithm

1. Start
2. Prompt the user to enter a number.
3. Read the input and store it in the variable "number".
4. Use an if-else statement to check if the number is even or odd.
5. Print the Result accordingly.
6. Stop

### Source Code

```
1 import java.util.Scanner;
2 public class EvenOdd
3 {
4     public static void main(String[] args)
5     {
6         Scanner scanner = new Scanner(System.in);
7         System.out.print("Enter a number : ");
8         int number = scanner.nextInt();
9         if(number % 2 == 0)
10        {
11            System.out.println(number + " is Even !");
12        }
13        else
14        {
15            System.out.println(number + " is Odd !");
16        }
17    }
18 }
```

### Result

The program was executed successfully.

When the input 9 was provided, the output was: "9 is Odd !"

When the input 12 was provided, the output was: "12 is Even !"

```
Enter a number : 9
9 is Odd !
Enter a number : 12
12 is Even !
```

## Sum of First n Natural Numbers

### Aim

Create an object-oriented Java program to compute the sum of the first n natural numbers.

### Algorithm

1. Start
2. Prompt the user to enter a number.
3. Read the input and store it in the variable "n".
4. Initialize sum = 0.
5. Use a for loop to iterate from 1 to n and add the current value to sum.
6. Print the sum of the first n natural numbers.
7. Stop

### Source Code

```
1 import java . util . Scanner ;
2 public class SumOfNumbers
3 {
4     public static void main ( String [] args )
5     {
6         Scanner scanner = new Scanner ( System . in ) ;
7         System . out . print ( " Enter a number : " ) ;
8         int n = scanner . nextInt ( ) ;
9         int sum = 0;
10        for ( int i = 1; i <= n ; i ++ )
11        {
12            sum += i ;
13        }
14        System . out . println ( " The sum of the first " + n + "
15        numbers is : " +
16        sum ) ;
17    }
```

### Result

The program was executed successfully.

When the input 5 was provided, the output was:

"The sum of the first 5 numbers is : 15"

Enter a number : 5

The sum of the first 5 numbers is : 15

## Factorial of a Number

### Aim

Create an object-oriented Java program to compute the factorial of a given number.

### Algorithm

1. Start
2. Prompt the user to enter a number.
3. Read the input and store it in the variable "number".
4. Initialize factorial = 1 and i = 1.
5. Use a while loop to iterate while i ≤ number.
6. Multiply factorial by i and increment i.
7. Print the factorial of the given number.
8. Stop

### Source Code

```
1 import java . util . Scanner ;
2 public class Factorial
3 {
4     public static void main ( String [] args )
5     {
6         Scanner scanner = new Scanner ( System . in ) ;
7         System . out . print ( " Enter a number : " ) ;
8         int number = scanner . nextInt ( ) ;
9         int factorial = 1;
10        int i = 1;
11        while ( i <= number )
12        {
13            factorial *= i ;
14            i ++;
15        }
16        System . out . println ( " The factorial of " + number + " is :
17        " + factorial ) ;
18    }
```

### Result

The program was executed successfully.

When the input 4 was provided, the output was:

"The factorial of 4 is : 24"

Enter a number : 4

The factorial of 4 is : 24

## Assigning Grades Based on Numeric Score

### Aim

Create an object-oriented Java program that assigns a grade based on a numeric score.

### Algorithm

1. Start
2. Prompt the user to enter the score.
3. Read the input and store it in the variable "score".
4. Use if-else to assign the grade based on the score.
5. Print the assigned grade.
6. Stop

### Source Code

```
1 import java.util.Scanner;
2 public class GradeClassification
3 {
4     public static void main(String[] args)
5     {
6         Scanner scanner = new Scanner(System.in);
7         System.out.print("Enter the score: ");
8         int score = scanner.nextInt();
9         char grade;
10
11         if (score > 100 || score < 0)
12         {
13             System.out.println("Invalid score!");
14             return;
15         } else if (score >= 90)
16         {
17             grade = 'A';
18         } else if (score >= 80)
19         {
20             grade = 'B';
21         } else if (score >= 70)
22         {
23             grade = 'C';
24         } else if (score >= 60)
25         {
26             grade = 'D';
27         } else
28         {
29             grade = 'F';
30         }
31         System.out.println("Your grade is: " + grade);
32     }
33 }
```

## Result

The program was executed successfully.

When the input 85 was provided, the output was:

"Your grade is : B"

```
Enter the score : 85
```

```
Your grade is : B
```



## Find Product with Lowest Price

### Aim

Create an object-oriented Java program to find product with lowest price.

### Algorithm

1. Start
2. Define a class Product with attributes pcode, pname, 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 lowest price.
6. Stop

### Source Code

```
1 import java.util.Scanner;
2
3 class Product
4 {
5     String pcode, pname;
6     double price;
7
8     Product(String pcode, String pname, double price)
9     {
10         this.pcode = pcode;
11         this.pname = pname;
12         this.price = price;
13     }
14
15     static Product findLowest(Product[] products)
16     {
17         Product lowest = products[0];
18         for (Product p : products)
19         {
20             if (p.price < lowest.price)
21             {
22                 lowest = p;
23             }
24         }
25         return lowest;
26     }
27
28     public static void main(String[] args)
29     {
30         Scanner sc = new Scanner(System.in);
31         Product[] products = new Product[3];
32
33         for (int i = 0; i < 3; i++)
34         {
```

```

35         System.out.println("Enter details for product " + (i + 1) +
        ":"");
36         System.out.print("Pcode: ");
37         String pcode = sc.next();
38         System.out.print("Pname: ");
39         String pname = sc.next();
40         System.out.print("Price: ");
41         double price = sc.nextDouble();
42         products[i] = new Product(pcode, pname, price);
43     }
44
45     Product lowest = findLowest(products);
46     System.out.println("\nProduct with Lowest Price:");
47     System.out.println("Pcode: " + lowest.pcode + ", Pname: " +
lowest.pname + ", Price: " + lowest.price);
48 }
49 }

```

## Result

The program was executed successfully.

Enter details for product 1:

Pcode: 101

Pname: chair

Price: 300

Enter details for product 2:

Pcode: 102

Pname: table

Price: 500

Enter details for product 3:

Pcode: 103

Pname: fan

Price: 200

Product with Lowest Price:

Pcode: 103, Pname: fan, Price: 200.0

## Complex Number Operations

### Aim

Create an object-oriented 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 respective methods.
6. Display the results.
7. Stop

### Source Code

```
1 import java.util.Scanner;
2
3 class Complex
4 {
5     double real, imag;
6
7     Complex(double real, double imag)
8     {
9         this.real = real;
10        this.imag = imag;
11    }
12
13    Complex add(Complex c)
14    {
15        return new Complex(this.real + c.real, this.imag + c.imag);
16    }
17
18    Complex multiply(Complex c)
19    {
20        double realPart = (this.real * c.real) - (this.imag * c.imag);
21        double imagPart = (this.real * c.imag) + (this.imag * c.real);
22        return new Complex(realPart, imagPart);
23    }
24
25    public String toString()
26    {
27        return real + " + " + imag + "i";
28    }
29
30    public static void main(String[] args)
31    {
```

```

32     Scanner sc = new Scanner(System.in);
33
34     System.out.print("Enter real and imaginary part of first
complex number: ");
35     Complex c1 = new Complex(sc.nextDouble(), sc.nextDouble());
36
37     System.out.print("Enter real and imaginary part of second
complex number: ");
38     Complex c2 = new Complex(sc.nextDouble(), sc.nextDouble());
39
40     Complex sum = c1.add(c2);
41     Complex product = c1.multiply(c2);
42
43     System.out.println("Sum: " + sum);
44     System.out.println("Product: " + product);
45
46 }
47 }

```

## Result

The program was executed successfully.

```

Enter real and imaginary part of first complex number: 5
3
Enter real and imaginary part of second complex number: 6
8
Sum: 11.0 + 11.0i
Product: 6.0 + 58.0i

```

## Matrix Addition

### Aim

Create an object-oriented Java program to perform matrix addition.

### Algorithm

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

### Source Code

```
1 import java.util.Scanner;
2
3 class MatrixAddition
4 {
5     public static void main(String[] args)
6     {
7         Scanner sc = new Scanner(System.in);
8
9         System.out.print("Enter number of rows and columns: ");
10        int rows = sc.nextInt();
11        int cols = sc.nextInt();
12
13        int[][] matrix1 = new int[rows][cols];
14        int[][] matrix2 = new int[rows][cols];
15        int[][] sumMatrix = new int[rows][cols];
16
17        System.out.println("Enter elements of first matrix:");
18        for (int i = 0; i < rows; i++)
19            for (int j = 0; j < cols; j++)
20                matrix1[i][j] = sc.nextInt();
21
22        System.out.println("Enter elements of second matrix:");
23        for (int i = 0; i < rows; i++)
24            for (int j = 0; j < cols; j++)
25                matrix2[i][j] = sc.nextInt();
26
27
28        for (int i = 0; i < rows; i++)
29            for (int j = 0; j < cols; j++)
30                sumMatrix[i][j] = matrix1[i][j] + matrix2[i][j];
31
32        System.out.println("Sum of matrices:");
```

```
33     for (int i = 0; i < rows; i++) {  
34         for (int j = 0; j < cols; j++)  
35             System.out.print(sumMatrix[i][j] + " ");  
36         System.out.println();  
37     }  
38 }  
39 }
```

## Result

The program was executed successfully.

```
Enter number of rows and columns: 2  
2  
Enter elements of first matrix:  
1 2  
3 1  
Enter elements of second matrix:  
2 4  
1 3  
Sum of matrices:  
3 6  
4 4
```

## Employee Search Using an Array of Objects

### Aim

Create an object-oriented Java program to store employee details including employee number, name, and salary, and search for an employee by employee number.

### Algorithm

1. Start
2. Create an array to store employee details.
3. Get the number of employees and their details from user.
4. Prompt the user to enter an employee number to search.
5. Search for the employee in the array.
6. If found, display the employee details.
7. Stop

### Source Code

```
1 import java.util.Scanner;
2
3 class Employee {
4     int empNo;
5     String name;
6     double salary;
7
8     Employee(int empNo, String name, double salary) {
9         this.empNo = empNo;
10        this.name = name;
11        this.salary = salary;
12    }
13
14    void display() {
15        System.out.println("Employee Number: " + empNo);
16        System.out.println("Name: " + name);
17        System.out.println("Salary: " + salary);
18    }
19 }
20
21 public class EmployeeSearch {
22     public static void main(String[] args) {
23         Scanner scanner = new Scanner(System.in);
24         System.out.print("Enter number of employees: ");
25         int n = scanner.nextInt();
26         Employee[] employees = new Employee[n];
27
28         for (int i = 0; i < n; i++) {
29             System.out.print("Enter Employee Number: ");
30             int empNo = scanner.nextInt();
31             scanner.nextLine();
```

```

32         System.out.print("Enter Name: ");
33         String name = scanner.nextLine();
34         System.out.print("Enter Salary: ");
35         double salary = scanner.nextDouble();
36         employees[i] = new Employee(empNo, name, salary);
37     }
38
39     System.out.print("Enter Employee Number to search: ");
40     int searchNo = scanner.nextInt();
41     boolean found = false;
42
43     for (Employee emp : employees) {
44         if (emp.empNo == searchNo) {
45             System.out.println("Employee Found:");
46             emp.display();
47             found = true;
48             break;
49         }
50     }
51
52     if (!found) {
53         System.out.println("Employee not found.");
54     }
55 }
56 }

```

## Result

The program was executed successfully.

```

Enter number of employees: 2
Enter Employee Number: 101
Enter Name: anjali
Enter Salary: 20000
Enter Employee Number: 102
Enter Name: minna
Enter Salary: 50000

Enter Employee Number to Search: 101
Employee Found :
Employee Number: 101
Name: anjali
salary: 20000.0

```



## String Search in an Array

### Aim

Create an object-oriented 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. Get the number of strings and store them in an array.
3. Prompt the user to enter the string to search.
4. Search for the string in the array.
5. If found, print its index; otherwise, print "String not found".
6. Stop

### Source Code

```
1 import java.util.Scanner;
2
3 public class StringSearch {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         System.out.print("Enter number of strings: ");
7         int n = scanner.nextInt();
8         scanner.nextLine();
9         String[] strings = new String[n];
10
11         for (int i = 0; i < n; i++) {
12             System.out.print("Enter string " + (i + 1) + ": ");
13             strings[i] = scanner.nextLine();
14         }
15
16         System.out.print("Enter string to search: ");
17         String searchStr = scanner.nextLine();
18         boolean found = false;
19
20         for (int i = 0; i < n; i++) {
21             if (strings[i].equals(searchStr)) {
22                 System.out.println("String found at index: " + i);
23                 found = true;
24                 break;
25             }
26         }
27
28         if (!found) {
29             System.out.println("String not found.");
30         }
31     }
32 }
```

## Result

The program was executed successfully.

```
Enter number of strings: 2
Enter string 1: anu
Enter string 2: minnu
Enter string to search: minnu
String found at index: 1
```

## String Manipulations

### Aim

Create an object-oriented 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. Get a string from the user.
3. Find and print the length of string.
4. Convert and print the string in uppercase and lowercase.
5. Extract and print specific characters and substrings.
6. Reverse and print the string.
7. Stop

### Source Code

```
1 import java.util.Scanner;
2
3 public class StringManipulation {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         System.out.print("Enter a string: ");
7         String str = scanner.nextLine();
8
9         System.out.println("Length: " + str.length());
10
11        System.out.println("Uppercase: " + str.toUpperCase());
12        System.out.println("Lowercase: " + str.toLowerCase());
13
14        if (str.length() > 2) {
15            System.out.println("First character: " + str.charAt(0));
16            System.out.println("Substring (first 3 characters): " + str
17                .substring(0, 3));
18        }
19
20        String reversed = new StringBuilder(str).reverse().toString();
21        System.out.println("Reversed: " + reversed);
22    }
23 }
```

### Result

The program was executed successfully.

```
Enter a string: anamika
Length: 7
Uppercase: ANAMIKA
Lowercase: anamika
First character: a
Substring: ana
Reversed: akimana
```

## Inheritance in Java

### Aim

Create an object-oriented 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. Define a base class Publisher with publisher details.
3. Create a derived class Book that extends Publisher and adds book.
4. Create two subclasses Literature and Fiction that extends Book.
5. Implement methods to read and display book details in each class.
6. Get user input to enter deetails for books in both categories.
7. Display the entered book details.
8. Stop

### Source Code

```
1 import java.util.Scanner;
2
3 class Publisher {
4     String publisherName;
5
6     Publisher(String publisherName) {
7         this.publisherName = publisherName;
8     }
9
10    void displayPublisher() {
11        System.out.println("Publisher: " + publisherName);
12    }
13 }
14
15 class Book extends Publisher {
16     String bookTitle;
17     String author;
18     double price;
19
20     Book(String publisherName, String bookTitle, String author, double
price) {
21         super(publisherName);
22         this.bookTitle = bookTitle;
23         this.author = author;
24         this.price = price;
25     }
26 }
```

```

27     void displayBookDetails() {
28         displayPublisher();
29         System.out.println("Book Title: " + bookTitle);
30         System.out.println("Author: " + author);
31         System.out.println("Price: $" + price);
32     }
33 }
34
35 class Literature extends Book {
36     Literature(String publisherName, String bookTitle, String author,
37         double price) {
38         super(publisherName, bookTitle, author, price);
39     }
40     void display() {
41         System.out.println("\n**Literature Book Details**");
42         displayBookDetails();
43     }
44 }
45
46 class Fiction extends Book {
47     Fiction(String publisherName, String bookTitle, String author,
48         double price) {
49         super(publisherName, bookTitle, author, price);
50     }
51     void display() {
52         System.out.println("\n**Fiction Book Details**");
53         displayBookDetails();
54     }
55 }
56
57 public class BookManagementSystem {
58     public static void main(String[] args) {
59         Scanner scanner = new Scanner(System.in);
60
61         System.out.println("Enter Literature Book Details:");
62         System.out.print("Publisher: ");
63         String litPublisher = scanner.nextLine();
64         System.out.print("Title: ");
65         String litTitle = scanner.nextLine();
66         System.out.print("Author: ");
67         String litAuthor = scanner.nextLine();
68         System.out.print("Price: ");
69         double litPrice = scanner.nextDouble();
70         scanner.nextLine();
71         Literature literature = new Literature(litPublisher, litTitle,
litAuthor, litPrice);
72
73         System.out.println("\nEnter Fiction Book Details:");
74         System.out.print("Publisher: ");
75         String ficPublisher = scanner.nextLine();
76         System.out.print("Title: ");
77         String ficTitle = scanner.nextLine();
78         System.out.print("Author: ");
79         String ficAuthor = scanner.nextLine();
80         System.out.print("Price: ");
81         double ficPrice = scanner.nextDouble();

```

```

82         Fiction fiction = new Fiction(ficPublisher, ficTitle, ficAuthor
      , ficPrice);
83
84         literature.display();
85         fiction.display();
86     }
87 }

```

## Result

The program was executed successfully.

```

Enter Literature Book Details:
Publisher: Penguin Random House
Title: The Great Gatsby
Author: F. Scott Fitzgerald
Price: 5999

```

```

Enter Fiction Book Details:
Publisher: HarperCollins
Title: The Night Circus
Author: Erin Morgenstern
Price: 7999

```

```

**Literature Book Details**
Publisher: Penguin Random House
Book Title: The Great Gatsby
Author: F. Scott Fitzgerald
Price: $5999.0

```

```

**Fiction Book Details**
Publisher: HarperCollins
Book Title: The Night Circus
Author: Erin Morgenstern
Price: $7999.0

```