
Course Outcome 1**Experiment 1****Date: 13.03.2024****Basic Java Programs****Aim:**

Write the following programs

- i) Print the prime numbers up to a limit
- ii) Print the 3-digit Armstrong numbers between two intervals.

Program

// Print the prime numbers up to a limit

```
import java.util.Scanner;

public class primelimit {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int i, j, n, c;
        System.out.print("enter a limit: ");
        n = scanner.nextInt();
        System.out.print("prime numbers are: ");

        for (i = 2; i <= n; i++) {
            c = 0;
            for (j = 1; j <= i; j++) {
                if (i % j == 0) {
                    c++;
                }
            }
            if (c == 2) {
                System.out.print(i + " ");
            }
        }
    }
}
```

Output

enter a limit: 10

prime numbers are: 2 3 5 7

Program

// Print the 3-digit Armstrong numbers between two intervals.

```
import java.util.*;

public class armstrong{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the starting interval: ");
        int start = scanner.nextInt();
        System.out.print("Enter the ending interval: ");
        int end = scanner.nextInt();
        System.out.println("Armstrong numbers between " + start + " and " + end + ":");
        for (int i = start; i <= end; i++) {
            int originalNumber = i;
            int n = 0;
            int result = 0;
            while (originalNumber != 0) {
                originalNumber /= 10;
                ++n;
            }
            originalNumber = i;
            while (originalNumber != 0) {
                int remainder = originalNumber % 10;
                result += Math.pow(remainder, n);
                originalNumber /= 10;
            }
            if (result == i && n == 3) {
                System.out.println(i);
            }
        }
    }
}
```

Output

```
Enter the starting interval: 100
Enter the ending interval: 500
Armstrong numbers between 100 and 500:
153
370
371
407
```

Experiment 2**Date: 13.03.2024****One-Dimensional Array****Aim**

Write a Java program to search an element in an array

Program

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

```
public class searches {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        System.out.print("Enter the size of the array: ");  
        int n = scanner.nextInt();  
        int[] arr = new int[n];  
        System.out.println("Enter the elements of the array:");  
        for (int i = 0; i < n; i++) {  
            arr[i] = scanner.nextInt();  
        }  
        System.out.print("Enter the element to search for: ");  
        int target = scanner.nextInt();  
        int index = -1;  
        for (int i = 0; i < n; i++) {  
            if (arr[i] == target) {  
                index = i;  
                break;  
            }  
        }  
        if (index == -1) {  
            System.out.println("Element not found in the array.");  
        } else {  
            System.out.println("Element found at index " + index + " of the array.");  
        }  
    }  
}
```

Output

Enter the size of the array: 5

Enter the elements of the array:

1 2 3 4 5

Enter the element to search for: 1

Element found at index 0 of the array.

Experiment 3**Date: 13.03.2024****Two-Dimensional Array****Aim**

Write a program to read a matrix from the console and check whether it is symmetric or not.

Program

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

```
public class symmetric {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of rows and columns: ");
        int n = scanner.nextInt();
        int matrix[][] = new int[n][n];
        System.out.println("Enter the elements of the matrix:");
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                matrix[i][j] = scanner.nextInt();
            }
        }
        scanner.close();
        boolean isSymmetric = true;
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                if (matrix[i][j] != matrix[j][i]) { // Compare with corresponding element
                    // across the diagonal
                    isSymmetric = false;
                    break;
                }
            }
            if (!isSymmetric) {
                break;
            }
        }
        if (isSymmetric) {
            System.out.println("The matrix is symmetric.");
        } else {
            System.out.println("The matrix is not symmetric.");
        }
    }
}
```

Output

Enter the number of rows and columns: 3

Enter the elements of the matrix:

1 2 3

2 4 5

3 5 6

The matrix is symmetric.

Enter the number of rows and columns: 3

Enter the elements of the matrix:

1 2 3

3 4 5

3 4 2

The matrix is not symmetric.

Experiment 4**Date: 13.03.2024****String Handling Methods- 1****Aim**

Perform the following operations on strings

- i. Find the length of the string
- ii. Character at second and fourth position
- iii. Find the sub string using start index only
- iv. Find the sub string using start index and end index
- v. Compare two strings lexicographically.
- vi. Compare two strings lexicographically, ignoring case differences.
- vii. Concatenate a given string to the end of another string.
- viii. Replace a specified character with another character.
- ix. Check whether a given string starts with another string.
- x. Convert all characters in a string to lowercase
- xii. Convert all characters in a string to uppercase.

Program

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

```
public class string {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        System.out.println("Enter the first string:");  
        String str1 = scanner.nextLine();  
        System.out.println("Enter the second string:");  
        String str2 = scanner.nextLine();  
        int length1 = str1.length();  
        int length2 = str2.length();  
        System.out.println("Length of the first string: " + length1);  
        System.out.println("Length of the second string: " + length2);  
        char charAt2 = str1.charAt(1);  
        char charAt4 = str1.charAt(3);  
        System.out.println("Character at second position in first string: " + charAt2);  
        System.out.println("Character at fourth position in first string: " + charAt4);  
        System.out.println("Enter the start index for substring:");  
        int startIndex = scanner.nextInt();  
        String subString1 = str1.substring(startIndex);  
        System.out.println("Substring from start index in first string: " + subString1);  
        System.out.println("Enter the start index for substring:");  
        int startIndex2 = scanner.nextInt();  
        System.out.println("Enter the end index for substring:");  
        int endIndex = scanner.nextInt();  
        String subString2 = str2.substring(startIndex2, endIndex);  
        System.out.println("Substring from start to end index in second string: " +  
            subString2);  
        int lexComparison = str1.compareTo(str2);  
    }  
}
```

```
System.out.println("Lexicographical comparison of the two strings: " +  
    lexComparison);  
int ignoreCaseComparison = str1.compareToIgnoreCase(str2);  
System.out.println("Lexicographical comparison ignoring case: " +  
    ignoreCaseComparison);  
String concatenatedString = str1.concat(str2);  
System.out.println("Concatenated string: " + concatenatedString);  
System.out.println("Enter the character to be replaced:");  
char charToReplace = scanner.next().charAt(0);  
System.out.println("Enter the character to replace with:");  
char replaceWith = scanner.next().charAt(0);  
String replacedString = str1.replace(charToReplace, replaceWith);  
System.out.println("String after replacement: " + replacedString);  
System.out.println("Enter the prefix string to check:");  
String prefix = scanner.next();  
boolean startsWithPrefix = str1.startsWith(prefix);  
System.out.println("Does the first string start with the prefix? " +  
    startsWithPrefix);  
String lowerCaseString = str1.toLowerCase();  
System.out.println("String in lowercase: " + lowerCaseString);  
String upperCaseString = str2.toUpperCase();  
System.out.println("String in uppercase: " + upperCaseString);  
scanner.close();  
}  
}
```

Output

Enter the first string:

hello

Enter the second string:

world

Length of the first string: 5

Length of the second string: 5

Character at second position in first string: e

Character at fourth position in first string: l

Enter the start index for substring:

2

Substring from start index in first string: llo

Enter the start index for substring:

1

Enter the end index for substring:

3

Substring from start to end index in second string: or

Lexicographical comparison of the two strings: -15

Lexicographical comparison ignoring case: -15

Concatenated string: helloworld

Enter the character to be replaced:

```
l
Enter the character to replace with:
x
String after replacement: hexxo
Enter the prefix string to check:
hel
Does the first string start with the prefix? true
String in lowercase: hello
String in uppercase: WORLD
```


Experiment 5**Date: 13.03.2024****String Handling Methods- 2****Aim**

Write a java program to

- i. Check whether a given string is palindrome or not.
- ii. Sorting a given list of names in ascending order

Program

// Check whether a given string is palindrome or not.

```
import java.util.Scanner;

class palindromes
{
    public static void main(String[] args)
    {
        int flag=0;
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter the string:");
        String str = scanner.nextLine();
        int len=str.length();
        for (int i = 0; i < len/2; i++)
        {
            if (str.charAt(i) != str.charAt(len-i-1))
            {
                flag=1;
                break;
            }
        }
        if(flag==1)
        {
            System.out.println("Not Palindrome");
        }
        else
        {
            System.out.println(str+" is Palindrome");
        }
    }
}
```

Output

Enter the string:

malayalam

malayalam is Palindrome

Enter the string:

hello

Not Palindrome

Program

// Sorting a given list of names in ascending order

```
import java.util.Scanner;

class sortnames
{
    public static void main(String args[])
    {
        String temp;
        Scanner SC = new Scanner(System.in);
        System.out.print("Enter the value of N: ");
        int N= SC.nextInt();
        SC.nextLine();
        String names[] = new String[N];

        System.out.println("Enter names: ");
        for(int i=0; i<N; i++)
        {
            System.out.print("Enter name [ " + (i+1) + " ]: ");
            names[i] = SC.nextLine();
        }
        for(int i=0; i<5; i++)
        {
            for(int j=1; j<5; j++)
            {
                if(names[j-1].compareTo(names[j])>0)
                {
                    temp=names[j-1];
                    names[j-1]=names[j];
                    names[j]=temp;
                }
            }
        }
        System.out.println("\nSorted names are in Ascending Order: ");
        for(int i=0;i<N;i++)
        {
            System.out.println(names[i]);
        }
    }
}
```

Output

Enter the value of N: 5

Enter names:

Enter name [1]: Anu

Enter name [2]: Vinu

Enter name [3]: Rinu

Enter name [4]: Binu

Enter name [5]: Kiran

Sorted names are in Ascending Order:

Anu

Binu

Kiran

Rinu

Vinu

Experiment 6**Date: 13.03.2024****StringBuffer Class Methods****Aim**

Write a program in java for string handling which performs the following

- i. Check the capacity of the StringBuffer object.
- ii. Reverse the content of this string and convert the resultant string in upper case
- iii. Read another string and append it to the resultant string of above.

Program

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

```
public class StringBufferHandling {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        System.out.print("Enter a string: ");  
        String inputString = scanner.nextLine();  
        StringBuffer stringBuffer = new StringBuffer(inputString);  
        System.out.println("Capacity of StringBuffer: " + stringBuffer.capacity());  
        stringBuffer.reverse();  
        stringBuffer = new StringBuffer(stringBuffer.toString().toUpperCase());  
        System.out.println("Reversed and Upper cased string: " + stringBuffer);  
        System.out.print("Enter another string to append: ");  
        String anotherString = scanner.nextLine();  
        stringBuffer.append(anotherString);  
        System.out.println("String after appending: " + stringBuffer);  
        scanner.close();  
    }  
}
```

Output

```
Enter a string: Hello  
Capacity of StringBuffer: 21  
Reversed and Upper cased string: OLLEH  
Enter another string to append: World  
String after appending: OLLEHWorld
```

Course Outcome 2**Experiment 7****Date: 15.03.2024****Initialize Instance Variables inside the Class****Aim**

Program to demonstrate use of command line arguments to initialize values to member variables in a class and to display them.

Hint:- Create a class containing Rlno, stud_name, engmark, mathsmark, totalmark. While executing the program we have to pass arguments through command line. These values are obtained in an array which is passed as argument to main function, here it is args[]. The marks are converted correspondingly and then passed to constructor where values are stored to class variables. Find the total marks and later displayed using display function.

Program

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

```
class Student {
    String Name;
    int rollNumber;
    int englishMarks;
    int mathsMarks;
    int total;
    public Student(int roll, String name, int engMarks, int mathMarks) {
        rollNumber = roll;
        Name = name;
        englishMarks = engMarks;
        mathsMarks = mathMarks;
        total = engMarks + mathsMarks;
    }

    public void display() {
        System.out.println("Roll Number: " + rollNumber);
        System.out.println("Student Name: " + Name);
        System.out.println("English Marks: " + englishMarks);
        System.out.println("Maths Marks: " + mathsMarks);
        System.out.println("Total Marks: " + total);
    }

    public static void main(String[] args) {
        int rollNumber = Integer.parseInt(args[0]);
        String Name = args[1];
        int englishMarks = Integer.parseInt(args[2]);
        int mathsMarks = Integer.parseInt(args[3]);
        Student s1 = new Student(rollNumber, Name, englishMarks, mathsMarks);
        s1.display();
    }
}
```

Output

Roll Number: 101

Student Name: Meenu

English Marks: 55

Maths Marks: 75

Total Marks: 130

Experiment 8**Date: 15.03.2024****Initialize Instance Variables inside the Class using Constructor****Aim**

Program to demonstrate use of constructors to initialize values to member variables in a class and to display them.

Hint:- empno, empname and salary are the class members of the class employee1. From the main function we are passing the values directly to a constructor, the constructor initializes the values to member variables. The display function is used to display the stored values of the member variables.

Program

```
import java.util.*;

class Employee1 {
    int empno;
    String empname;
    double salary;
    Employee1(int empno, String empname, double salary) {
        this.empno = empno;
        this.empname = empname;
        this.salary = salary;
    }
    void display() {
        System.out.println("Employee Number: " + empno);
        System.out.println("Employee Name: " + empname);
        System.out.println("Salary: $" + salary);
    }
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        System.out.println("Enter Employee Number: ");
        int empno = in.nextInt();
        System.out.println("Enter Employee Name: ");
        String empname = in.next();
        System.out.println("Enter Salary: $");
        double salary = in.nextDouble();
        Employee1 ob = new Employee1(empno, empname, salary);
        ob.display();
    }
}
```

Output

```
Enter Employee Number:
1001
Enter Employee Name:
Rajeev
Enter Salary: $
50000
Employee Number: 1001
```

Employee Name: Rajeev
Salary: \$50000.0

Experiment 9**Date: 15.03.2024****Matrix Operations****Aim**

Read 2 matrices from the console and perform matrix addition and multiplication using class and object.

Program

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

```
public class MatrixOperations {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of rows for matrix 1: ");
        int rows1 = scanner.nextInt();
        System.out.print("Enter the number of columns for matrix 1: ");
        int cols1 = scanner.nextInt();

        System.out.print("Enter the number of rows for matrix 2: ");
        int rows2 = scanner.nextInt();
        System.out.print("Enter the number of columns for matrix 2: ");
        int cols2 = scanner.nextInt();

        if (cols1 != rows2) {
            System.out.println("Matrices cannot be multiplied.");
            return;
        }
        int[][] matrix1 = inputMatrix(rows1, cols1, "Matrix 1", scanner);
        int[][] matrix2 = inputMatrix(rows2, cols2, "Matrix 2", scanner);
        int[][] sum = addMatrices(matrix1, matrix2);
        System.out.println("Sum of the matrices:");
        printMatrix(sum);
        int[][] product = multiplyMatrices(matrix1, matrix2);
        System.out.println("Product of the matrices:");
        printMatrix(product);
    }
    private static int[][] inputMatrix(int rows, int cols, String name, Scanner scanner)
    {
        System.out.println("Enter elements for " + name + ":");
        int[][] matrix = new int[rows][cols];
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                System.out.print(name + "[" + (i + 1) + "][" + (j + 1) + "]: ");
                matrix[i][j] = scanner.nextInt();
            }
        }
        return matrix;
    }
}
```

```
}
private static int[][] addMatrices(int[][] matrix1, int[][] matrix2) {
    int rows = matrix1.length;
    int cols = matrix1[0].length;
    int[][] sum = new int[rows][cols];
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            sum[i][j] = matrix1[i][j] + matrix2[i][j];
        }
    }
    return sum;
}

private static int[][] multiplyMatrices(int[][] matrix1, int[][] matrix2) {
    int rows1 = matrix1.length;
    int cols1 = matrix1[0].length;
    int cols2 = matrix2[0].length;
    int[][] product = new int[rows1][cols2];
    for (int i = 0; i < rows1; i++) {
        for (int j = 0; j < cols2; j++) {
            for (int k = 0; k < cols1; k++) {
                product[i][j] += matrix1[i][k] * matrix2[k][j];
            }
        }
    }
    return product;
}

private static void printMatrix(int[][] matrix) {
    int rows = matrix.length;
    int cols = matrix[0].length;
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            System.out.print(matrix[i][j] + " ");
        }
        System.out.println();
    }
}
}
```

Output

Enter the number of rows for matrix 1: 2
Enter the number of columns for matrix 1: 2
Enter the number of rows for matrix 2: 2
Enter the number of columns for matrix 2: 2
Enter elements for Matrix 1:
Matrix 1[1][1]: 1
Matrix 1[1][2]: 2
Matrix 1[2][1]: 3

Matrix 1[2][2]: 4

Enter elements for Matrix 2:

Matrix 2[1][1]: 5

Matrix 2[1][2]: 6

Matrix 2[2][1]: 7

Matrix 2[2][2]: 8

Sum of the matrices:

6 8

10 12

Product of the matrices:

19 22

43 50

Experiment 10**Date: 15.03.2024****Complex Number Addition****Aim**

Write a Java program add two complex numbers using object as argument

Program

```
import java.io.*;
import java.util.*;
class Complex
{
    float real;
    float img;
    Complex()
    {
        real=0.0f; img=0.0f;
    }
    void read() throws IOException
    {
        DataInputStream in =new DataInputStream(System.in);
        System.out.println("Enter the real part:");
        real=Float.valueOf(in.readLine());
        System.out.println("Enter the imaginary part:");
        img=Float.valueOf(in.readLine());
    }
    void display()
    {
        System.out.println(real+"+i" +img);
    }
    void sum(Complex c1,Complex c2)
    {
        real=c1.real+c2.real;
        img=c1.img+c2.img;
    }
}
public class complexarithmetic
{
    public static void main(String args[])throws IOException
    {
        Complex c1=new Complex();
        Complex c2=new Complex();
        Complex res=new Complex();
        System.out.println("Enter the first complex number:");
        c1.read();
        System.out.println("Enter the second complex number:");
        c2.read();
        System.out.println("first complex number:");
        c1.display();
```

```
System.out.println("Second complex number:");  
c2.display();  
res.sum(c1,c2);  
System.out.println("The resultant complex number:");  
res.display();  
}}
```

Output

Enter the first complex number:

Enter the real part:

1

Enter the imaginary part:

2

Enter the second complex number:

Enter the real part:

3

Enter the imaginary part:

4

first complex number:

1.0+i2.0

Second complex number:

3.0+i4.0

The resultant complex number:

4.0+i6.0

Experiment 11**Date: 15.03.2024****Class and Objects****Aim**

Define a class 'product' with data members pcode, pname and price. Create 3 objects of the class and find the product having the lowest price.

Program

```
import java.util.*;
class Product {
    private int pcode;
    private String pname;
    private double price;
    Product(int pcode, String pname, double price) {
        this.pcode = pcode;
        this.pname = pname;
        this.price = price;
    }
    int getPcode() {
        return pcode;
    }
    String getPname() {
        return pname;
    }
    double getPrice() {
        return price;
    }
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        Product[] products = new Product[3];
        for (int i = 0; i < products.length; i++) {
            System.out.println("Enter details for product " + (i + 1) + ":");
            System.out.print("Product Code: ");
            int pcode = in.nextInt();
            in.nextLine();
            System.out.print("Product Name: ");
            String pname = in.nextLine();
            System.out.print("Price: ");
            double price = in.nextDouble();
            products[i] = new Product(pcode, pname, price);
        }
        Product lowestPriceProduct = products[0];
        for (int i = 1; i < products.length; i++) {
            if (products[i].getPrice() < lowestPriceProduct.getPrice()) {
                lowestPriceProduct = products[i];
            }
        }
    }
}
```

```
        System.out.println("Product with lowest price:");  
        System.out.println("Product Code: " + lowestPriceProduct.getPcode());  
        System.out.println("Product Name: " + lowestPriceProduct.getPname());  
        System.out.println("Price: " + lowestPriceProduct.getPrice());  
    }  
}
```

Output

Enter details for product 1:
Product Code: 111
Product Name: Soap
Price: 50
Enter details for product 2:
Product Code: 231
Product Name: Powder
Price: 75

Experiment 12**Date: 15.03.2024****Inner Class and Static Nested Class****Aim**

Create CPU with attribute price. Create inner class Processor with attributes no. of cores, manufacturer and static nested class RAM with attributes memory and manufacturer. Create an object of CPU class and print information of Processor and RAM.

Program

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

```
class CPU {
    double price;
    class Processor {
        int cores;
        String manufacturer;
        Processor(int cores, String manufacturer) {
            this.cores = cores;
            this.manufacturer = manufacturer;
        }
        void display() {
            System.out.println("Processor Cores: " + cores);
            System.out.println("Processor Manufacturer: " + manufacturer);
        }
    }
    static class RAM {
        int memory;
        String manufacturer;
        RAM(int memory, String manufacturer) {
            this.memory = memory;
            this.manufacturer = manufacturer;
        }
        void display() {
            System.out.println("RAM Memory: " + memory + "GB");
            System.out.println("RAM Manufacturer: " + manufacturer);
        }
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        CPU cpu = new CPU();
        System.out.print("Enter CPU Price: $");
        cpu.price = scanner.nextDouble();
        System.out.print("Enter Processor Cores: ");
        int cores = scanner.nextInt();
        scanner.nextLine();
        System.out.print("Enter Processor Manufacturer: ");
        String processorManufacturer = scanner.nextLine();
    }
}
```

```
CPU.Processor processor = cpu.new Processor(cores, processorManufacturer);
System.out.print("Enter RAM Memory (GB): ");
int memory = scanner.nextInt();
scanner.nextLine();
System.out.print("Enter RAM Manufacturer: ");
String ramManufacturer = scanner.nextLine();
CPU.RAM ram = new CPU.RAM(memory, ramManufacturer);
scanner.close();
System.out.println("\nCPU Price: $" + cpu.price);
System.out.println("\nProcessor Information:");
processor.display();
System.out.println("\nRAM Information:");
ram.display();
}
}
```

Output

Enter CPU Price: \$8000
Enter Processor Cores: 10
Enter Processor Manufacturer: INTEL
Enter RAM Memory (GB): 128
Enter RAM Manufacturer: Micron

CPU Price: \$8000.0

Processor Information:
Processor Cores: 10
Processor Manufacturer: INTEL

RAM Information:
RAM Memory: 128GB
RAM Manufacturer: Micron

Experiment 13**Date: 15.03.2024****Array of Objects****Aim**

Program to create a class for Employee having attributes eNo, eName, eSalary. Read 'n' employee information and Search for an employee given eNo, using the concept of array of Objects.

Program

```
import java.util.Scanner;
class Employee {
    int eNo;
    String eName;
    double eSalary;
    Employee(int eNo, String eName, double eSalary) {
        this.eNo = eNo;
        this.eName = eName;
        this.eSalary = eSalary;
    }
    void displayEmployeeInfo() {
        System.out.println("Employee Details:");
        System.out.println("Employee Number: " + this.eNo);
        System.out.println("Employee Name: " + this.eName);
        System.out.println("Employee Salary: " + this.eSalary);
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter number of employees: ");
        int n = scanner.nextInt();
        Employee[] employees = new Employee[n];
        for (int i = 0; i < n; i++) {
            System.out.println("Enter details for employee " + (i + 1));
            System.out.print("Enter employee number: ");
            int eNo = scanner.nextInt();
            scanner.nextLine();
            System.out.print("Enter employee name: ");
            String eName = scanner.nextLine();
            System.out.print("Enter employee salary: ");
            double eSalary = scanner.nextDouble();
            employees[i] = new Employee(eNo, eName, eSalary);
        }
        System.out.print("Enter employee number to search: ");
        int searchNo = scanner.nextInt();
        for (Employee employee : employees) {
            if (employee.eNo == searchNo) {employee.displayEmployeeInfo();
                break;
            }
        }
    }
}
```

```
}  
}
```

Output

```
Enter number of employees: 3  
Enter details for employee 1  
Enter employee number: 1001  
Enter employee name: Rajeev  
Enter employee salary: 40000  
Enter details for employee 2  
Enter employee number: 1002  
Enter employee name: Sajeev  
Enter employee salary: 50000  
Enter details for employee 3  
Enter employee number: 1003  
Enter employee name: Sreeni  
Enter employee salary: 60000  
Enter employee number to search: 1001  
Employee Details:  
Employee Number: 1001  
Employee Name: Rajeev  
Employee Salary: 40000.0
```