

Institute of Computer Technology
B. Tech. Computer Science and Engineering

Semester: III

Sub: Object-Oriented Programming
Course Code: 2CSE303

Practical Number:5

Objective:

To understand the concept of class, object, constructor and loop.

Q1. Mr. Malhotra has just started Programming, he is in first year of Engineering. Malhotra is reading about Relational Operators. Relational Operators are operators which check relationship between two values. Given two numerical values A and B you need to help Malhotra in finding the relationship between them that is, first one is greater

Code :

```
import java.util.Scanner;

public class RelationalOperators {
    public static void main(String[] args) {
        System.out.print("Enter Number of test case:");
        Scanner scanner = new Scanner(System.in);
        int T = scanner.nextInt(); // Number of test cases

        for (int i = 0; i < T; i++) {
            int A = scanner.nextInt();
            int B = scanner.nextInt();

            if (A > B) {
                System.out.println(">");
            } else if (A < B) {
                System.out.println("<");
            } else {
                System.out.println("=");
            }
        }
        scanner.close();
    }
}
```

Output :

```
Enter Number of test case: 33
1 2
<
22 33
<
22 22
=
```

Q2. Write an appropriate program of the following.

1. Generate series of natural number like: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
2. Generate table of any random number like: 5, 10, 15, 20, 25, 30, 35, 40, 45, 50,...
3. Generate series like: 6,11,16,21,26,.....
4. Generate Fibonacci series like: 0 1 1 2 3 5 8 13 21 34

Code :

```
public class SeriesGeneration {
    public static void main(String[] args) {
        generateNaturalNumbers(10);
        generateMultiplicationTable(5, 10);
        generateCustomSeries(6, 5, 10);
        generateFibonacciSeries(10);
    }

    // 1. Natural Number Series
    public static void generateNaturalNumbers(int n) {
        for (int i = 1; i <= n; i++) {
            System.out.print(i + " ");
        }
        System.out.println();
    }

    // 2. Multiplication Table
    public static void generateMultiplicationTable(int num, int length) {
        for (int i = 1; i <= length; i++) {
            System.out.print(num * i + " ");
        }
        System.out.println();
    }

    // 3. Custom Series
    public static void generateCustomSeries(int start, int step, int length) {
```

```
        for (int i = 0; i < length; i++) {
            System.out.print(start + (step * i) + " ");
        }
        System.out.println();
    }

    // 4. Fibonacci Series
    public static void generateFibonacciSeries(int length) {
        int a = 0, b = 1;
        System.out.print(a + " " + b + " ");
        for (int i = 2; i < length; i++) {
            int c = a + b;
            System.out.print(c + " ");
            a = b;
            b = c;
        }
        System.out.println();
    }
}
```

Output :

```
1 2 3 4 5 6 7 8 9 10
5 10 15 20 25 30 35 40 45 50
6 11 16 21 26 31 36 41 46 51
0 1 1 2 3 5 8 13 21 34
```

Q3. Write an appropriate program of the following.

1. Check, whether the user input one three-digit number is Armstrong number or not.
2. Check, whether the user input one three-digit number is palindrome number or not.
3. Check, whether the user input number is perfect number or not.

Code :

```
import java.util.Scanner;

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

        System.out.print("Enter a three-digit number: ");
        int num = scanner.nextInt();

        System.out.println("Is Armstrong: " + isArmstrong(num));
        System.out.println("Is Palindrome: " + isPalindrome(num));
    }
}
```

```
        System.out.println("Is Perfect: " + isPerfect(num));

        scanner.close();
    }

    // Check if Armstrong number
    public static boolean isArmstrong(int num) {
        int original = num, sum = 0;
        while (num > 0) {
            int digit = num % 10;
            sum += digit * digit * digit;
            num /= 10;
        }
        return sum == original;
    }

    // Check if Palindrome
    public static boolean isPalindrome(int num) {
        int original = num, reversed = 0;
        while (num > 0) {
            int digit = num % 10;
            reversed = reversed * 10 + digit;
            num /= 10;
        }
        return reversed == original;
    }

    // Check if Perfect number
    public static boolean isPerfect(int num) {
        int sum = 0;
        for (int i = 1; i <= num / 2; i++) {
            if (num % i == 0) {
                sum += i;
            }
        }
        return sum == num;
    }
}
```

Output :

```
Enter a three-digit number: 324
Is Armstrong: false
Is Palindrome: false
Is Perfect: false
```

Q4. If Given an integer N. Write a program to obtain the sum of the first and last digits of this number.

Code :

```
import java.util.Scanner;

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

        for (int i = 0; i < T; i++) {
            int num = scanner.nextInt();
            System.out.println(sumFirstLastDigits(num));
        }
        scanner.close();
    }

    public static int sumFirstLastDigits(int num) {
        int lastDigit = num % 10;
        while (num >= 10) {
            num /= 10;
        }
        int firstDigit = num;
        return firstDigit + lastDigit;
    }
}
```

Output :

```
3
1234
5
23454
6
34334
7
```