Week 1 – Core Java Fundamentals (Java Developer Intern)

Aim: NUMBER UTILITY PROGRAM

Description: Create a Java program that performs operations like prime check, palindrome, factorial, Fibonacci, etc.

Code :

```
import java.util.Scanner;


public class NumberUtils {


    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        boolean running = true;


        while (running) {
            displayMenu();


            System.out.print("\nYour choice: ");
            int choice = getValidInt(sc);


            switch (choice) {
                case 1:
                    handlePrimeCheck(sc);
                    break;
                case 2:
                    handlePalindrome(sc);
                    break;
                case 3:
                    handleFactorial(sc);
                    break;
                case 4:
```

```
                handleFibonacci(sc);
                break;
            case 5:
                handleArmstrong(sc);
                break;
            case 6:
                System.out.println("\nThanks for using the program!");
                running = false;
                break;
            default:
                System.out.println("\nInvalid choice. Pick something between 1-6.");
        }

        if (running) {
            System.out.println("\n" + "=".repeat(40));
        }
    }

    sc.close();
}

private static void displayMenu() {
    System.out.println("\n=== Number Utilities ===");
    System.out.println("1. Check if Prime");
    System.out.println("2. Check if Palindrome");
    System.out.println("3. Calculate Factorial");
    System.out.println("4. Generate Fibonacci Sequence");
    System.out.println("5. Check Armstrong Number");
    System.out.println("6. Exit");
}
```

```
private static boolean isPrime(int num) {
    if (num <= 1) return false;
    if (num == 2) return true;
    if (num % 2 == 0) return false;

    for (int i = 3; i * i <= num; i += 2) {
        if (num % i == 0) {
            return false;
        }
    }
    return true;
}

private static void handlePrimeCheck(Scanner sc) {
    System.out.print("\nEnter a number: ");
    int num = getValidInt(sc);

    if (num < 0) {
        System.out.println("Prime check works with positive numbers only.");
        return;
    }

    if (isPrime(num)) {
        System.out.println(num + " is a prime number!");
    } else {
        System.out.println(num + " is not a prime number.");
    }
}

private static boolean isPalindrome(int num) {
    int original = num;
```

```
    int reversed = 0;

    while (num > 0) {
        int digit = num % 10;
        reversed = reversed * 10 + digit;
        num /= 10;
    }

    return original == reversed;
}

private static void handlePalindrome(Scanner sc) {
    System.out.print("\nEnter a number: ");
    int num = getValidInt(sc);

    if (num < 0) {
        System.out.println("Palindrome check works with positive numbers only.");
        return;
    }

    if (isPalindrome(num)) {
        System.out.println(num + " is a palindrome!");
    } else {
        System.out.println(num + " is not a palindrome.");
    }
}

private static long factorial(int n) {
    if (n < 0) return -1;
    if (n == 0 || n == 1) return 1;
```

```
    long result = 1;
    for (int i = 2; i <= n; i++) {
        if (result > Long.MAX_VALUE / i) {
            return -1;
        }
        result *= i;
    }
    return result;
}

private static void handleFactorial(Scanner sc) {
    System.out.print("\nEnter a number: ");
    int num = getValidInt(sc);

    if (num < 0) {
        System.out.println("Factorial isn't defined for negative numbers.");
        return;
    }

    if (num > 20) {
        System.out.println("Number too large - would cause overflow. Try something <= 20.");
        return;
    }

    long result = factorial(num);
    if (result == -1) {
        System.out.println("Overflow occurred!");
    } else {
        System.out.println(num + "! = " + result);
    }
}
```

```
private static void generateFibonacci(int terms) {
    if (terms <= 0) {
        System.out.println("Number of terms must be positive.");
        return;
    }

    System.out.print("Fibonacci sequence: ");

    long first = 0, second = 1;

    for (int i = 1; i <= terms; i++) {
        System.out.print(first);
        if (i < terms) System.out.print(", ");

        long next = first + second;
        first = second;
        second = next;
    }
    System.out.println();
}

private static void handleFibonacci(Scanner sc) {
    System.out.print("\nHow many terms? ");
    int terms = getValidInt(sc);

    if (terms > 50) {
        System.out.println("That's too many terms. Let's keep it under 50.");
        return;
    }
```

```
        generateFibonacci(terms);
    }

    private static boolean isArmstrong(int num) {
        int original = num;
        int sum = 0;
        int digits = String.valueOf(num).length();

        while (num > 0) {
            int digit = num % 10;
            sum += Math.pow(digit, digits);
            num /= 10;
        }

        return sum == original;
    }

    private static void handleArmstrong(Scanner sc) {
        System.out.print("\nEnter a number: ");
        int num = getValidInt(sc);

        if (num < 0) {
            System.out.println("Armstrong check works with positive numbers only.");
            return;
        }

        if (isArmstrong(num)) {
            System.out.println(num + " is an Armstrong number!");
        } else {
            System.out.println(num + " is not an Armstrong number.");
        }
```

```
    }

    private static int getValidInt(Scanner sc) {
        while (!sc.hasNextInt()) {
            System.out.print("That's not a valid number. Try again: ");
            sc.next();
        }
        return sc.nextInt();
    }
}
```

Output:

```
=== Number Utilities ===
1. Check if Prime
2. Check if Palindrome
3. Calculate Factorial
4. Generate Fibonacci Sequence
5. Check Armstrong Number
6. Exit

Your choice: 1

Enter a number: 13
13 is a prime number!

========================================
```

2) Palindrome:

```
=== Number Utilities ===
1. Check if Prime
2. Check if Palindrome
3. Calculate Factorial
4. Generate Fibonacci Sequence
5. Check Armstrong Number
6. Exit

Your choice: 2

Enter a number: 121
121 is a palindrome!

========================================
```

3) Factorial :

```
=== Number Utilities ===
1. Check if Prime
2. Check if Palindrome
3. Calculate Factorial
4. Generate Fibonacci Sequence
5. Check Armstrong Number
6. Exit

Your choice: 3

Enter a number: 5
5! = 120

========================================
```

4) Fibonacci Sequence:

```
=== Number Utilities ===
1. Check if Prime
2. Check if Palindrome
3. Calculate Factorial
4. Generate Fibonacci Sequence
5. Check Armstrong Number
6. Exit

Your choice: 4

How many terms? 6
Fibonacci sequence: 0, 1, 1, 2, 3, 5

========================================
```

5) Armstrong Number:

```
=== Number Utilities ===
1. Check if Prime
2. Check if Palindrome
3. Calculate Factorial
4. Generate Fibonacci Sequence
5. Check Armstrong Number
6. Exit

Your choice: 5

Enter a number: 371
371 is an Armstrong number!

========================================
```