Java Programming Day 4

By Kavuri Santosh Kumar

Basic Understanding:

1. What are looping statements in Java? Why are they used?
2. List the different types of looping (iterative) statements available in Java.

Syntax and Usage:

3. Write the syntax of a `for` loop in Java.
4. How does a `while` loop work in Java? Provide an example.
5. What is the difference between a `while` loop and a `do-while` loop in Java?

Practice Problem statements

based on **for**, **while**, and **do-while** loops:

For Loop Problems:

1. **Print Numbers from 1 to N**: Write a program to print numbers from 1 to N using a for loop.
2. **Sum of First N Natural Numbers**: Write a program to find the sum of the first N natural numbers using a for loop.
3. **Multiplication Table**: Write a program to print the multiplication table of a number N using a for loop.
4. **Factorial of a Number**: Write a program to find the factorial of a given number using a for loop.
5. Write a program to generate the **Fibonacci series** up to a certain number of terms n.?
6. Write a program to check the given number is **prime or not**.?
7. **Find Prime Numbers in a Range**: Write a program to print all prime numbers between 1 and N using a for loop.

While Loop Problems:

1. **Print Numbers from 1 to N**: Write a program to print numbers from 1 to N using a while loop.
2. **Sum of Digits of a Number**: Write a program to calculate the sum of digits of a given number using a while loop.
3. **Find the Largest Digit in a Number**: Write a program to find the largest digit in a given number using a while loop.
4. **Count Even Numbers in a Range**: Write a program to count how many even numbers are there between 1 and N using a while loop.

Do-While Loop Problems:

1. **Print Numbers from 1 to N**: Write a program to print numbers from 1 to N using a do-while loop.
2. **Find the Sum of Digits of a Number**: Write a program to find the sum of digits of a number using a do-while loop.
3. **Calculate Power of a Number**: Write a program to calculate the power of a number (base raised to the exponent) using a do-while loop.

2. Looping Statements

These allow repeated execution of a block of code.

a. for Loop

Executes a block of code a fixed number of times.

Syntax:

```
for (initialization; condition; increment/decrement) {
   // code to execute
}
```

Example 1:

```
class NumPrint {
   public static void main(String[] args) {
      for (int i = 1; i <= 10; i++) {
         System.out.println(i);
      }
   }
}
```

Step-by-Step Iteration:

1. **Initialization**:
   - The `for` loop starts with `int i = 1;`.
   - Variable `i` is initialized to `1`.
2. **Condition Check (i <= 10)**:
   - The loop checks if `i` is less than or equal to `10`.
   - If `true`, the loop body is executed.
   - If `false`, the loop ends.
3. **Loop Body Execution**:
   - Inside the loop, the current value of `i` is printed using `System.out.println(i)`.
4. **Increment (i++)**:
   - After executing the loop body, `i` is incremented by `1` (i.e., `i = i + 1`).
5. **Repeat**:

- The condition `i <= 10` is checked again. If true, steps 3 and 4 are repeated.
- If false, the loop exits.

Iterative Steps in Detail:

Iteration	`i` Value	Condition (`i <= 10`)	Action
1	1	True	Print `1`, Increment `i` (2)
2	2	True	Print `2`, Increment `i` (3)
3	3	True	Print `3`, Increment `i` (4)
4	4	True	Print `4`, Increment `i` (5)
5	5	True	Print `5`, Increment `i` (6)
6	6	True	Print `6`, Increment `i` (7)
7	7	True	Print `7`, Increment `i` (8)
8	8	True	Print `8`, Increment `i` (9)
9	9	True	Print `9`, Increment `i` (10)
10	10	True	Print `10`, Increment `i` (11)
11	11	False	Exit loop

Example 2:

```
class EvenNumPrint {
   public static void main(String[] args) {
      for (int i = 2; i <= 20; i += 2) {
         System.out.println(i);
      }
   }
}
```

Example 3:

```
import java.util.Scanner;
class NumPrint {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter a number: ");
      int num = scanner.nextInt();
      for (int i = 1; i <= num; i++) {
         System.out.println(i);
      }
   }
}
```

b. while Loop

Executes a block of code as long as the condition is true.

Syntax:

```
while (condition) {
   // code to execute
}
```

Example 2:

```
import java.util.Scanner;
class WhileLoopExample {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter a number: ");
      int num = scanner.nextInt();
      int i = 1;
      while (i <= num) {
         System.out.println(i);
         i++;
      }
   }
}
```

c. do-while Loop

Executes the block of code at least once, then repeats as long as the condition is true.

```
do {
   // code to execute
} while (condition);
```

Example 1:

```
import java.util.Scanner;
class DoWhileLoopExample {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter a number: ");
      int num = scanner.nextInt();
      int i = 1;
      do {
```

```
            System.out.println(i);
            i++;
        } while (i <= num);
    }
}
```

---

3. Jump Statements

These alter the flow of execution by skipping or terminating loops.

a. break

Exits a loop or switch statement prematurely.

```
for (int i = 0; i < 10; i++) {
    if (i == 5) {
        break; // exit the loop
    }
}
```

b. continue

Skips the current iteration and proceeds to the next iteration.

```
for (int i = 0; i < 10; i++) {
    if (i % 2 == 0) {
        continue; // skip even numbers
    }
    System.out.println(i);
}
```

c. return

Exits from a method and optionally returns a value.

```
int sum(int a, int b) {
    return a + b; // returns the result
}
```

Answers:

For Loop Problems:

1. **Print Numbers from 1 to N**:

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

```
class PrintNumbers {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      int N = sc.nextInt();
      for (int i = 1; i <= N; i++) {
         System.out.println(i);
      }
   }
}
```

2. **Sum of First N Natural Numbers**:

```
import java.util.Scanner;
class SumOfNaturalNumbers {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      int N = sc.nextInt();
      int sum = 0;
      for (int i = 1; i <= N; i++) {
         sum += i;
      }
      System.out.println("Sum: " + sum);
   }
}
```

3. **Multiplication Table**:

```
import java.util.Scanner;
class MultiplicationTable {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      int N = sc.nextInt();
      for (int i = 1; i <= 10; i++) {
         System.out.println(N + " * " + i + " = " + (N * i));
      }
   }
}
```

4. **Factorial of a Number**:

```
import java.util.Scanner;
class Factorial {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      int N = sc.nextInt();
      int fact = 1;
      for (int i = 1; i <= N; i++) {
         fact *= i;
      }
      System.out.println("Factorial: " + fact);
   }
```

```
}
```

5. **Fibonacci series up to a certain number of terms n.?**

```
import java.util.Scanner;

class Main {

   public static void main(String[] args) {

      int a=0;

      int b=1;

      int c;

      for(int i=1;i<=10;i++){

         System.out.println(a+" ");

         c=a+b;

         a=b;

         b=c;

      }

   }

}
```

6. Write a program to check the given number is **prime or not**.?

```
class Main {
   public static void main(String[] args) {
      int num=15;
      int count=0;
      for(int i=1;i<=num;i++){
         if(num%i==0){
            count++;
         }
      }
      if(count==2){
         System.out.println("is a prime");
      }
```

```
            else{
                System.out.println("Is not a prime");
            }

        }
    }
```

7. **Find Prime Numbers in a Range**: Write a program to print all prime numbers between 1 and N using a for loop.?

   CODE:

```
class Main {
    public static void main(String[] args) {
      int N=20;
     for(int num=2;num<=N;num++){
        boolean isPrime=true;
        for(int i=2;i<=num/2;i++){
            if(num%i==0){
                isPrime=false;
                break;
            }
        }
        if(isPrime){
            System.out.println(num);
        }
     }

    }
}
```

While Loop Problems:

1. **Print Numbers from 1 to N**:

```
import java.util.Scanner;
class PrintNumbersWhile {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int N = sc.nextInt();
        int i = 1;
        while (i <= N) {
            System.out.println(i);
            i++;
        }
    }
}
```

2. **Sum of Digits of a Number**:

```
import java.util.Scanner;
class SumOfDigits {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      int number = sc.nextInt();
      int sum = 0;
      while (number > 0) {
         sum += number % 10;
         number /= 10;
      }
      System.out.println("Sum of Digits: " + sum);
   }
}
```

3. **Find the Largest Digit in a Number**:

```
import java.util.Scanner;
class LargestDigit {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      int number = sc.nextInt();
      int largestDigit = 0;
      while (number > 0) {
         int digit = number % 10;
         if (digit > largestDigit) {
            largestDigit = digit;
         }
         number /= 10;
      }
      System.out.println("Largest Digit: " + largestDigit);
   }
}
```

4. **Count Even Numbers in a Range**:

```
import java.util.Scanner;
class CountEvenNumbers {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      int N = sc.nextInt();
      int count = 0;
      int i = 1;
      while (i <= N) {
         if (i % 2 == 0) {
            count++;
         }
```

```
            i++;
        }
        System.out.println("Count of Even Numbers: " + count);
    }
}
```

Do-While Loop Problems:

1. **Print Numbers from 1 to N**:

```
import java.util.Scanner;
class PrintNumbersDoWhile {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int N = sc.nextInt();
        int i = 1;
        do {
            System.out.println(i);
            i++;
        } while (i <= N);
    }
}
```

2. **Find the Sum of Digits of a Number**:

```
import java.util.Scanner;
class SumOfDigitsDoWhile {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int number = sc.nextInt();
        int sum = 0;
        do {
            sum += number % 10;
            number /= 10;
        } while (number > 0);
        System.out.println("Sum of Digits: " + sum);
    }
}
```

3. **Calculate Power of a Number**:

```
import java.util.Scanner;
class PowerOfNumber {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int base = sc.nextInt();
        int exponent = sc.nextInt();
        int result = 1;
        int i = 1;
```

```
        do {
            result *= base;
            i++;
        } while (i <= exponent);
        System.out.println("Result: " + result);
    }
}
```