Ministry of Higher Education
Jami University
Scientific assistantship
Faculty (Computer Science)
Department of Information Systems

while loop in java

The `while` loop in Java is used to execute a block of code repeatedly as long as a specified condition evaluates to `true`. It is particularly useful when the number of iterations is not known beforehand. The condition is checked before the execution of the loop's body, making it possible for the loop to not execute at all if the condition is `false` at the outset.

### **Syntax:**

```java
while (condition) {
    // code to be executed
}
```

### **10 Examples of `while` Loop in Java:**

#### **Example 1: Basic `while` Loop**

Print numbers from 0 to 4.

```java
int i = 0;
while (i < 5) {
    System.out.println("i = " + i);
    i++;
}
```

Output:

```
i = 0
i = 1
i = 2
i = 3
i = 4
```

#### **Example 2: Summing Numbers**

Calculate the sum of the first 5 natural numbers.

```java
int sum = 0;
int i = 1;
while (i <= 5) {
    sum += i;  // Add i to sum
    i++;
}
System.out.println("Sum = " + sum);
```

Output:

```
Sum = 15
```

#### **Example 3: User Input**

Keep asking the user for input until they enter "exit."

```java
import java.util.Scanner;

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

        System.out.println("Type 'exit' to quit.");
        while (true) {
            System.out.print("Enter something: ");
            input = scanner.nextLine();
            if (input.equalsIgnoreCase("exit")) {
                break;  // Exit the loop when user types "exit"
            }
            System.out.println("You entered: " + input);
        }
        scanner.close();
    }
}
```

Output: (Sample run)

```
Type 'exit' to quit.
Enter something: Hello
You entered: Hello
Enter something: exit
```

#### **Example 4: Factorial Calculation**

Calculate the factorial of a number.

```java
int number = 5;
int factorial = 1;
int i = 1;

while (i <= number) {
    factorial *= i;  // Multiply factorial by i
    i++;
}
System.out.println("Factorial of " + number + " = " + factorial);
```

Output:

```
Factorial of 5 = 120
```

#### **Example 5: Counting Down**

Use a `while` loop to count down from 10 to 1.

```java
int count = 10;
while (count > 0) {
    System.out.println("Countdown: " + count);
    count--;
}
System.out.println("Liftoff!");
```

Output:

```
Countdown: 10
Countdown: 9
Countdown: 8
Countdown: 7
Countdown: 6
Countdown: 5
Countdown: 4
Countdown: 3
Countdown: 2
Countdown: 1
Liftoff!
```

#### **Example 6: Infinite Loop**

Create an infinite loop (use with caution).

```java
int count = 0;
while (true) {
    System.out.println("Count: " + count);
    count++;
```

    if (count == 5) {
        break; // Break the loop when count reaches 5
    }
}
```
**Output:**
```
Count: 0
Count: 1
Count: 2
Count: 3
Count: 4
~~~

Example 7: Checking Prime Numbers
Check if a number is prime.
```java
int num = 29;
boolean isPrime = true;
int i = 2;

while (i <= Math.sqrt(num)) {
    if (num % i == 0) {
        isPrime = false; // Not a prime number
        break;
    }
    i++;
~~~

```
}
System.out.println(num + " is prime: " + isPrime);
```

Output:

```
29 is prime: true
```

#### **Example 8: Print Digits of a Number**

Print each digit of a number.

```java
int number = 12345;
while (number > 0) {
    int digit = number % 10; // Get the last digit
    System.out.println("Digit: " + digit);
    number /= 10; // Remove the last digit
}
```

Output:

```
Digit: 5
Digit: 4
Digit: 3
Digit: 2
Digit: 1
```

#### **Example 9: Reverse a Number**

Reverse the digits of a number.

```java
int num = 12345;
int reversed = 0;

while (num != 0) {
    int digit = num % 10; // Get the last digit
    reversed = reversed * 10 + digit; // Append the digit
    num /= 10; // Remove the last digit
}
System.out.println("Reversed Number: " + reversed);
```

Output:

```
Reversed Number: 54321
```

#### **Example 10: Collatz Conjecture**

Implement a simple version of the Collatz conjecture.

```java
int n = 6; // Starting number
System.out.print("Collatz sequence: " + n + " ");

while (n != 1) {
    if (n % 2 == 0) {
        n /= 2; // Even
    } else {
        n = 3 * n + 1; // Odd
    }
    System.out.print(n + " ");
}
```

Output:

```
Collatz sequence: 6 3 10 5 16 8 4 2 1
```

### Summary

- The `while` loop is a powerful construct for executing a block of code repeatedly based on a condition.
- It's particularly useful when the number of iterations is not known beforehand.
- The loop will not execute at all if the condition is initially `false`, making it different from the `do-while` loop, which guarantees at least one execution.