More About C++ Loops

By: Mir Faraz Ali

Programming Fundamentals

What is a Loop?

- What is a Loop?
- A loop is a way to repeat a set of instructions many times until a condition is met.
- Instead of writing the same code again and again, we use loops.

- Real life example:
- A teacher checks all students' papers one by one (repetition).
- A washing machine spins clothes until the timer ends.

The While Statement

- Definition:

  - A while loop repeats code as long as a condition is true.
  - Condition is checked before each repetition

Example:

```
#include <iostream>
using namespace std;

int main() {
    int count = 1;
    while (count <= 5) {
        cout << "Student " << count << " present"
<< endl;
        count++;  // increment
    }
    return 0;
}
```

Increment and Decrement Operators

- Definition:
- Increment (++) increases a value by 1.
- Decrement (--) decreases a value by 1.

```
int x = 5;
x++;   // now x = 6
x--;   // now x = 5 again
```

Real life example:

- Adding one chocolate to the box (increment).
- Eating one chocolate from the box (decrement).

The For Statement

- Definition:
  - A for loop is best when we know how many times to repeat.
  - It has 3 parts: initialization, condition, update.

Example:

```
for (int i = 1; i <= 5; i++) {
    cout << "Hello " << i << endl;
}
```

What Kind of Loop to Use?

- Definition:

- for loop → When number of repetitions is known.
- while loop → When we don't know how many times, but condition controls it.
- do-while loop → When we want the loop to run at least once.

- Real life examples:
- for: Counting 1 to 100.
- while: Keep reading until end of book.
- do-while: Show menu once, then repeat if user wants.

The Break Statement

- Definition:

Used to stop a loop early, even if the condition is still true.

Example:

```
for (int i = 1; i <= 10; i++) {
    if (i == 5) {
        break;   // stop loop when i = 5
    }
    cout << i << " ";
}
```

Designing Loops in C++

- Definition:

Designing a loop means planning carefully before writing the loop, so it works correctly and doesn't run forever.

- Decide what to repeat.
- Decide how many times (condition).
- Decide how to update the counter.

Loops for Sums and Products

- Sum: Add numbers together.

Example:

```
int sum = 0;
for (int i = 1; i <= 5; i++) {
    sum += i;  // same as sum = sum + i
}
cout << "Sum = " << sum;  // Output: 15
```

Loops for Sums and Products

- Product: Multiply numbers together.

Example:

```
int product = 1;
for (int i = 1; i <= 5; i++) {
    product *= i;  // same as product = product * i
}
cout << "Product = " << product;  // Output: 120
```

Ending a Loop

- A loop ends when:

The condition becomes false (for/while).

Or we use a break statement.

Example:

```
int x = 1;
while (x <= 3) {
    cout << "Running..." << endl;
    x++;
}
cout << "Loop ended.";
```

End Of Class