



Designing Loops

By: Mir Faraz Ali
Programming Fundamentals



The Break Statement

- The break statement is used to end a loop immediately, even if the loop condition is still true.
- It “jumps out” of the loop and continues with the next part of the program.

- Real-life example:

Leaving a cinema early if you don't like the movie.

Example:

```
#include <iostream>
using namespace std;
int main() {
    for (int i = 1; i <= 10; i++) {
        if (i == 5) {
            break; // stop when i = 5
        }
        cout << i << " ";
    }
    return 0;
}
```

The Continue Statement

- **Definition:**
 - The continue statement is used to skip one repetition of the loop, but not end the loop.
 - It moves to the next iteration immediately.

Real-life example:

A teacher checks students' homework but skips absent students.

Example:

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

```
int main() {
    for (int i = 1; i <= 5; i++) {
        if (i == 3) {
            continue; // skip when i = 3
        }
        cout << i << " ";
    }
    return 0;
}
```

Debugging a Loop

- Definition:
 - Debugging means finding and fixing errors in a program.
 - When debugging loops, check:
 1. Initialization – Did the loop start correctly?
 2. Condition – Will the loop stop? Or is it infinite?
 3. Update – Is the counter increasing/decreasing properly?
 4. Logic – Is the loop body doing what you expect?

Example of Infinite Loop (bug):

```
int i = 1;  
while (i <= 5) {  
    cout << i << endl;  
    // forgot i++ → infinite loop  
}
```

Nested Loops

- Definition:
 - A loop inside another loop is called a nested loop.
 - The inner loop runs completely every time the outer loop runs once.
 - Useful for tables, patterns, and multi-level tasks.

Example:

```
#include <iostream>
using namespace std;
int main() {
    for (int i = 1; i <= 3; i++) {        // outer loop
        for (int j = 1; j <= 3; j++) {    // inner loop
            cout << i << " x " << j << " = " << i * j << endl;
        }
        cout << endl;
    }
    return 0;
}
```



End Of Class

