



*Adapted from C++ How To Program edited for our own purposes*

## break and continue statements

---

C++ also provides statements break and continue to alter the flow of control. The preceding section showed how break can be used to terminate a switch statement's execution. This section discusses how to use break in a repetition statement.

### break Statement

The break statement, when executed in a while, for, do...while or switch statement, causes immediate exit from that statement. Program execution continues with the next statement. Common uses of the break statement are to escape early from a loop or to skip the remainder of a switch statement.

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

int main()
{
    int count; // control variable also used after loop terminates

    for ( count = 1; count <= 10; ++count ) // loop 10 times
    {
        if ( count == 5 )
            break; // break loop only if count is 5

        cout << count << " ";
    } // end for

    cout << "\nBroke out of loop at count = " << count << endl;
} // end main
```

```
1 2 3 4
Broke out of loop at count = 5
```

### continue Statement

The continue statement, when executed in a while, for or do...while statement, skips the remaining statements in the body of that statement and proceeds with the next iteration of the loop. In while and do...while statements, the loop-continuation test evaluates immediately after the continue statement executes.

In the for statement, the increment expression executes, then the loop-continuation test evaluates.

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

int main()
{
    for ( int count = 1; count <= 10; ++count ) // loop 10 times
    {
        if ( count == 5 ) // if count is 5,
            continue;    // skip remaining code in loop

        cout << count << " ";
    } // end for

    cout << "\nUsed continue to skip printing 5" << endl;
} // end main
```

```
1 2 3 4 6 7 8 9 10
Used continue to skip printing 5
```