

Basic Structure

```
#include <iostream> //Header file for input-output
operations
using namespace std; //Allow using standard
namespace
int main()
{
cout << " Hel lo, Wor ld! " << endl; //Prints
output to console
return 0;
}
```

Selection Statements

| Statement | Description | Syntax |
|-------------------|---|--|
| If Statement | Executes a statement if a condition is true | <code>if (condition) {statement(s)};</code> |
| If-else Statement | Executes one block if the condition is true, another if false | <code>if (condition) {statement(s);} else{statement(s)};</code> |
| Ternary Operator | A shorthand for if-else | <code>variable = (condition) ? value if true : value if false;</code> |
| Switch Statement | Selects a case to execute based on a variable's value | <code>switch (expression) {case value: statement; break; default: statement;}</code> |

Loop Control Statements

| Statement | Description |
|-----------|---|
| break | Exits a loop or switch statement. |
| continue | Skips the rest of the loop iteration and continues to the next. |
| exit() | Terminates the program immediately. |

IF_ELSE-SWITCH

| if-else Statement Example | switch Statement Example |
|--|--|
| <pre>#include <iostream> using namespace std; int main() { int num; cout << "Enter a number: "; cin >> num; if (num % 2 == 0) { cout << "The number is even.\n"; } else { cout << "The number is odd.\n"; } return 0; }</pre> | <pre>#include <iostream> using namespace std; int main() { int day; cout << "Enter a day number (1-7): "; cin >> day; switch (day) { case 1: cout << "Monday\n"; break; case 2: cout << "Tuesday\n"; break; case 3: cout << "Wednesday\n"; break; case 4: cout << "Thursday\n"; break; case 5: cout << "Friday\n"; break; case 6: cout << "Saturday\n"; break; case 7: cout << "Sunday\n"; break; default: cout << "Invalid input\n"; } return 0; }</pre> |

WHILE LOOP

| while Loop Example | do-while Loop Example |
|--|---|
| <pre>#include <iostream> using namespace std; int main() { int count = 0; while (count < 5) { cout << "Count: " << count << endl; count++; } return 0; }</pre> | <pre>#include <iostream> using namespace std; int main() { int num; do { cout << "Enter a positive number (1 to exit): "; cin >> num; } while (num != -1); cout << "Program terminated.\n"; return 0; }</pre> |

Input and Output

```
int num;
cout << " Enter a number: ";
cin >> num;
cout << "You entered: " << num << endl;
```

Repetition Statements (Loop)

| Loop Type | Description | Syntax |
|---------------|---|---|
| while loop | Repeats a block of code while a condition is true. | <code>while (condition) {statement(s)};</code> |
| do-while loop | Executes at least once, then repeats while a condition is true. | <code>do {statement(s);} while (condition);</code> |
| for Loop | A compact loop with initialization, condition, and increment. | <code>for (int; condition; update) { statement(s); }</code> |



By **lukenelson**

cheatography.com/lukenelson/

Published 2nd March, 2025.

Last updated 2nd March, 2025.

Page 1 of 2.

Sponsored by **ApolloPad.com**

Everyone has a novel in them. Finish

Yours!

<https://apollopad.com>

Types of Loop

| Types | Description |
|--------------------------|--|
| Counter-Controlled Loop | Executes a set number of times. |
| Sentinel-Controlled Loop | Runs until a special value is entered. |
| Flag-Controlled Loop | Uses a bool flag to control execution. |
| Nested Loops | A loop inside another loop. |

Compound Statement

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

int main()
{
    float fPrice;

    cout << "Enter the price:";
    cin >> fPrice;
    if (fPrice >= 1000)
    {
        cout << " Expensive!\n";
        cout << "Can't Afford!\n";
    }
    return 0;
}
```

FOR-BREAK

| for Loop Example | break Statement Example |
|---|--|
| <pre>#include <iostream> using namespace std; int main() { for (int i = 1; i <= 10; i++) { cout << i << " "; } cout << "\nLoop finished.\n"; return 0; }</pre> | <pre>#include <iostream> using namespace std; int main() { for (int i = 0; i < 10; i++) { if (i == 5) { cout << "Breaking loop at i = " << i << endl; break; } cout << i << " "; } return 0; }</pre> |

CONTINUE-NESTED

| continue Statement Example | Nested Loop Example |
|---|--|
| <pre>#include <iostream> using namespace std; int main() { for (int i = 0; i < 5; i++) { if (i == 2) continue; cout << "Iteration " << i << endl; } return 0; }</pre> | <pre>#include <iostream> using namespace std; int main() { for (int i = 1; i <= 3; i++) { for (int j = 1; j <= 5; j++) { cout << " "; } cout << endl; } return 0; }</pre> |

