Programming Essentials

(UOMU022023)

اساسيات البرمجة
2025-2024

Decision Making 2
(Switch Statement)

by
Dr Murtada Dohan
murtada.dohan@uomus.edu.iq


AL MUSTAQBAL UNIVERSITY
جامعة المستقبل
أسست عام
2010


كلية الهندسة والتقنيات
الهندسية
COLLEGE OF ENGINEERING &
TECHNOLOGY
AL MUSTAQBAL UNIVERSITY
جامعة المستقبل
2010

Recap

Syntax of if-else Statement

```
if (condition) {
   // code block if condition is true
} else {
   // code block if condition is false
}
```

1. Boolean variable

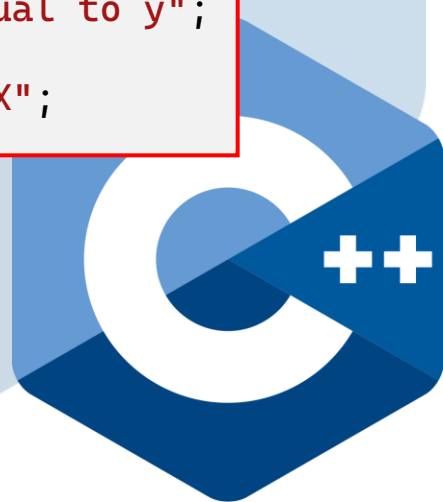
Example:

```
bool x = true;
if (x)
     cout << "Here we go!";
else
     cout << "We cannot make it";
```

2. Comparison

Example:

```
int x = 10, y = 56;
if (x>=y)
     cout << "X is greater or equal to y";
else
     cout << "Y is greater than X";
```



How if Statement Works

Condition is true

```
int number = 5;

if (number > 0) {
    // code
}

// code after if
```

Condition is false

```
int number = 5;

if (number < 0) {
    // code
}

// code after if
```

Ref: https://www.programiz.com/cpp-programming/if-else

How if...else Statement Works

1st Condition is true

```
int number = 2;
if (number > 0) {
    // code
}
else if (number == 0){
    // code
}
else {
    //code
}

//code after if
```

2nd Condition is true

```
int number = 0;
if (number > 0) {
    // code
}
else if (number == 0){
    // code
}
else {
    //code
}

//code after if
```

All Conditions are false

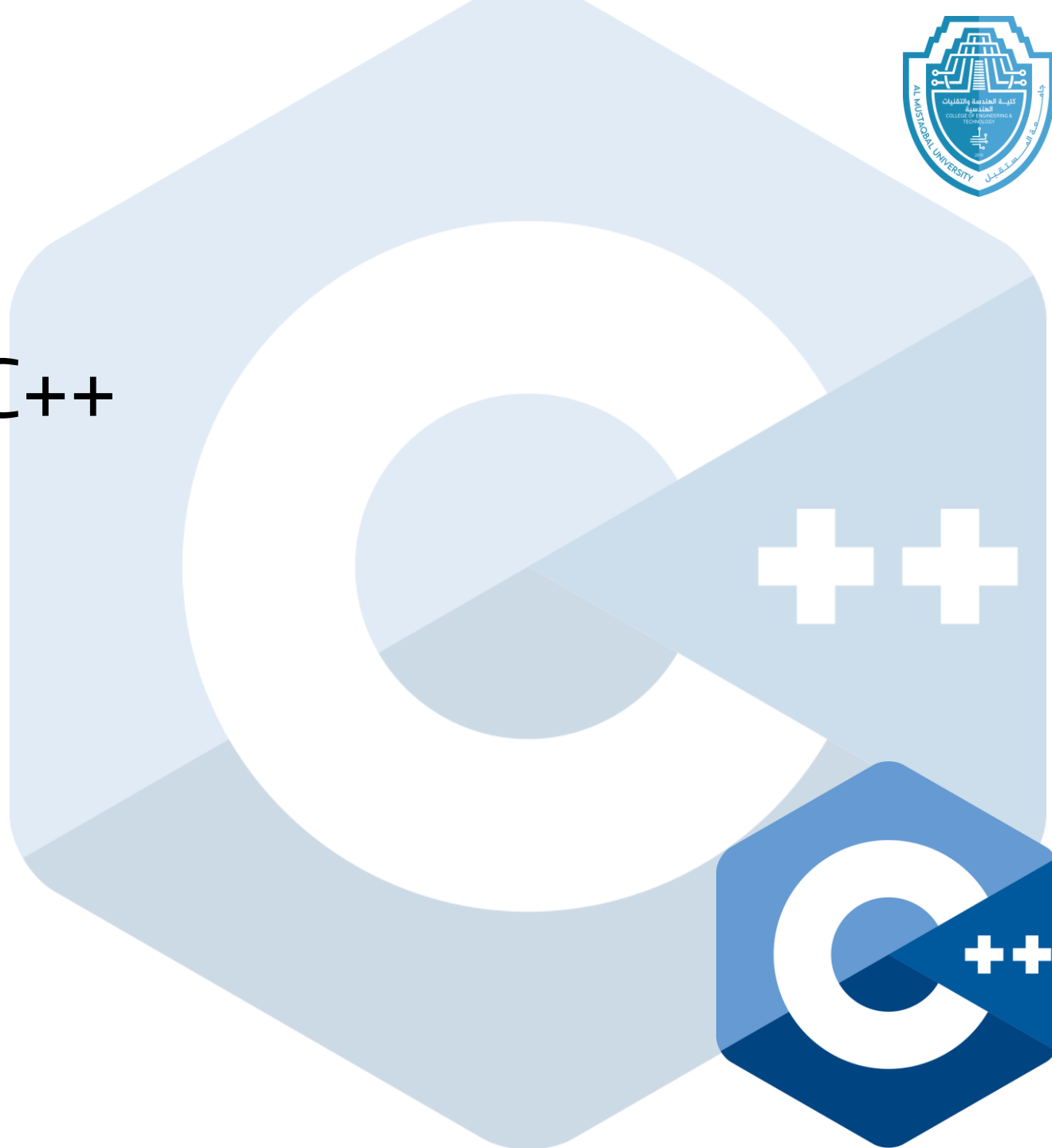
```
int number = -2;
if (number > 0) {
    // code
}
else if (number == 0){
    // code
}
else {
    //code
}

//code after if
```

Ref: https://www.programiz.com/cpp-programming/if-else

Today, Agenda

- Switch Statement in C++



Introduction to Switch Statement

- The switch statement allows us to execute a block of code among many alternatives.
- You can do the same thing with the if...else statement. However, the syntax of the switch statement is much easier to read and write.
- Why use switch?
  - Cleaner alternative to multiple if-else conditions
  - Efficient for fixed value comparisons

Syntax of switch Statement

```
switch (expression) {
case value1:
    // code block
    break;
case value2:
    // code block
    break;
    ...
default:
    // code if no case matches
```

- Key Points:
  - expression must evaluate to an integer or character
  - break exits the switch (if omitted, execution falls through)
  - default is optional (executes if no case matches)

Example 1 – Simple Grade Checker

- Using if-else:

```
char grade = 'B';
if (grade == 'A')
    cout << "Excellent!";
else if (grade == 'B')
    cout << "Good!";
else if (grade == 'C')
    cout << "Average!";
else
    cout << "Needs improvement!";
```

- Using switch (Cleaner!):

```
char grade = 'B';
 switch (grade) {
 case 'A':
     cout << "Excellent!"; break;
 case 'B':
     cout << "Good!"; break;
 case 'C':
     cout << "Average!"; break;
 default:
     cout << "Needs improvement!";
 }
```

Example 2 – Day of the Week

- Using switch:

```
int day = 3;
switch (day) {
case 1: cout << "Monday"; break;
case 2: cout << "Tuesday"; break;
case 3: cout << "Wednesday"; break;
    // ... cases 4–7 ...
default: cout << "Invalid day!";
}
```

- Using switch (without Break):

```
   int day = 2;
   switch (day) {
   case 1: cout << "Monday";
   case 2: cout << "Tuesday"; // No break → falls through!
   case 3: cout << "Wednesday";
   }
   // Output: "TuesdayWednesday"
```

switch vs if-else Comparison

Feature	switch	if-else
Readability	Better for fixed values	Better for ranges (x > 10)
Performance	Faster (jump table)	Slower (sequential checks)
Conditions	Only constants (case 5:)	Any condition (if (x > 5))
Fall-through	Possible (if no break)	Not applicable

- When to use switch?
  - Checking a variable against multiple constant values (e.g., menus, enums).
- When to use if-else?
  - Complex conditions (ranges, logical operators).

Common Pitfalls & Best Practices

- Pitfalls:
  - Forgetting break → unintended fall-through.
  - Using non-constant expressions in case.

- Best Practices:
  - ✓ Always include break unless fall-through is intentional.
  - ✓ Use default for handling unexpected values.
  - ✓ Prefer switch over if-else for readability when appropriate.

Example 1: Simple Calculator

```cpp
#include <iostream>
using namespace std;
int main()
{
    char op;
    float num1, num2;
    cout << "Enter operator (+, -, *, /): ";
    cin >> op;
    cout << "Enter two numbers: ";
    cin >> num1 >> num2;

    switch (op) {
    case '+': cout << "Result: " << num1 + num2; break;
    case '-': cout << "Result: " << num1 - num2; break;
    case '*': cout << "Result: " << num1 * num2; break;
    case '/': cout << "Result: " << num1 / num2; break;
    default: cout << "Invalid operator!";
    }
 }
```

Example 2: Traffic Light System

- Task: Simulate traffic light behavior (R = Red, Y = Yellow, G = Green).

```
#include <iostream>
using namespace std;
int main()
{
    char light = 'Y';
    switch (light) {
    case 'R': cout << "Stop!"; break;
    case 'Y': cout << "Slow down!"; break;
    case 'G': cout << "Go!"; break;
    default: cout << "Invalid light!";
    }
 }
```

Example 3: Vowel or Consonant Checker

```cpp
#include <iostream>
using namespace std;
int main()
{
    char ch;
    cout << "Enter a letter: ";
    cin >> ch;

    switch (ch) {
    case 'a': case 'e': case 'i': case 'o': case 'u':
        cout << "Vowel!"; break;
    default:
        cout << "Not Vowel!";
    }
 }
```

Example 4: Month Days Counter

```cpp
#include <iostream>
using namespace std;
int main()
{
    int month;
    cout << "Enter month (1-12): ";
    cin >> month;

    switch (month) {
    case 1: case 3: case 5: case 7: case 8: case 10: case 12:
        cout << "31 days"; break;
    case 4: case 6: case 9: case 11:
        cout << "30 days"; break;
    case 2:
        cout << "28 or 29 days (leap year)"; break;
    default:
        cout << "Invalid month!";
    }
}
```

What's the output?

```
#include <iostream>
using namespace std;
int main()
{
    int x = 2;
    switch (x) {
    case 1: cout << "One"; break;
    case 2: cout << "Two";
    case 3: cout << "Three"; break;
    default: cout << "Unknown";
  }
}
```

Summary & Key Takeaways

- switch is efficient for fixed-value comparisons.
- More readable than if-else for multiple conditions.
- Always use break (unless fall-through is needed).
- default case handles unexpected inputs.

Let’s try C++

Install Visual Studio and familiarise yourself with its interface.