Programming Essentials

(UOMU022023)

اساسيات البرمجة
2025-2024

Decision Making

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

What is a Variable?

- A variable is a named storage that holds data.
- Used to store values that can change during program execution.
- Every variable has:
  - A name
  - A type
  - A value

Syntax of Variable Definition

- Syntax of Variable Definition:
  - type variableName;

    or
  - type variableName = value;
- Example:
  - int age;
  - float salary = 5000.50;
  - char grade = 'A';

Example of creating variables

Type	Description	Example
int	Integer numbers	int x = 5;
float	Decimal numbers	float y = 3.2;
double	Larger decimal numbers	double z = 7.456;
char	Single characters: A,a,B,b.. %,$,*: 0,1,..	char c = 'A';
bool	Boolean values	bool isTrue = true;

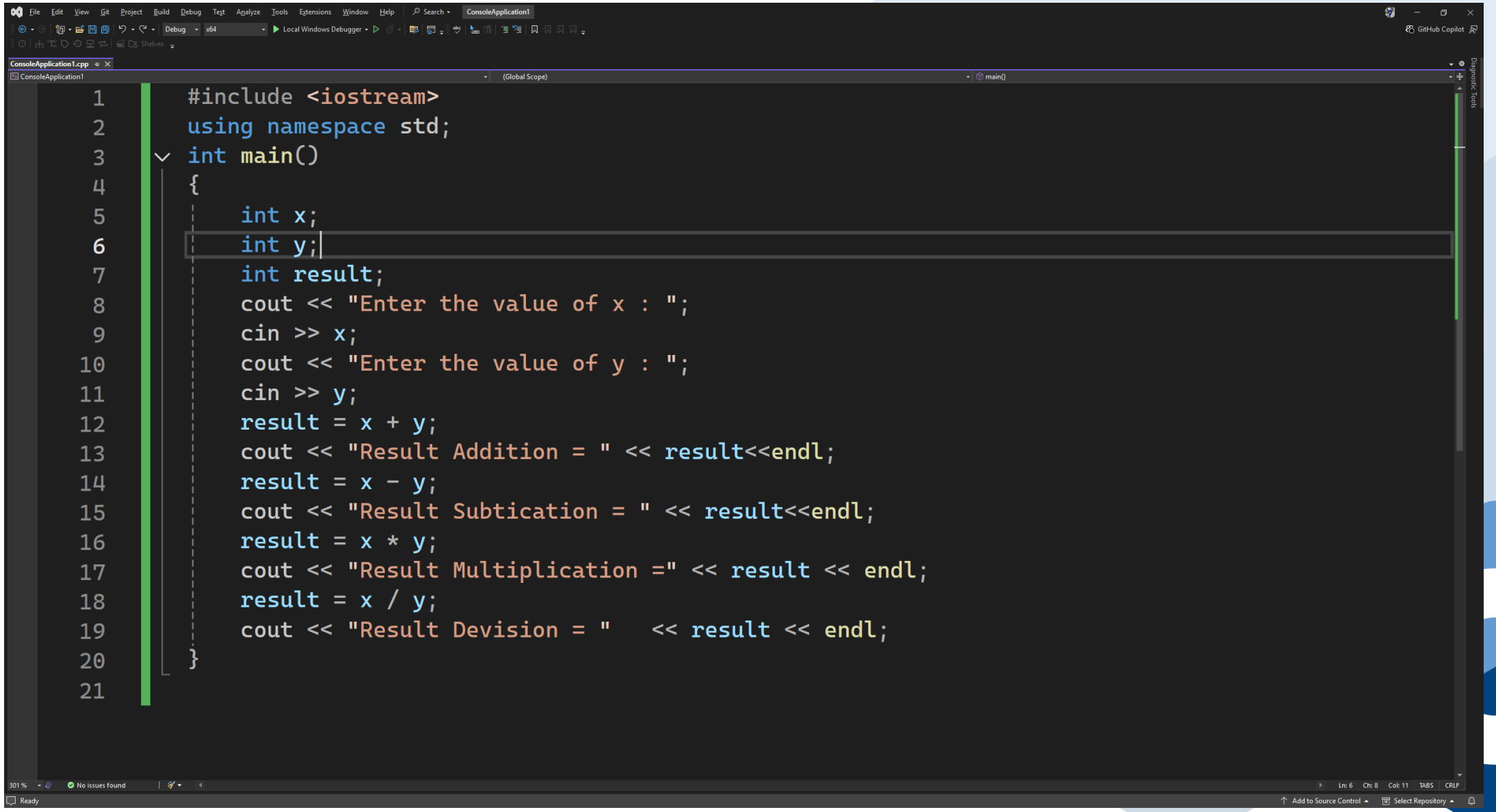
Example: Read two grades of a student and print the average of these grades

```
#include <iostream>
using namespace std;
int main()
{
int grade1, grade2, results;
cin >> grade1 >> grade2;
results = (grade1 + grade2)/2;
cout << "Results =" < results;
return 0;
}
```

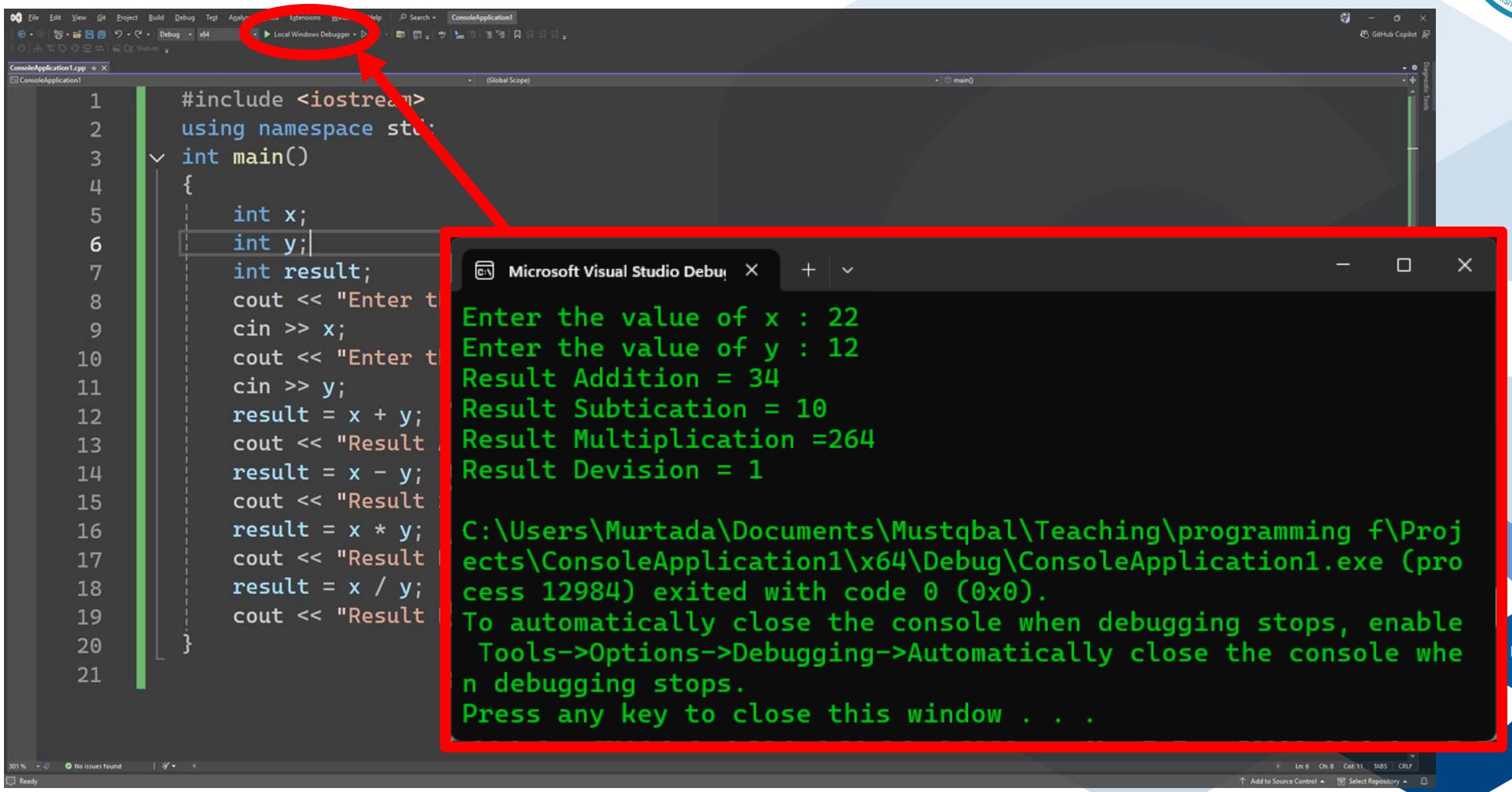
Example: Read Two numbers and print the results of the current operations (+,-,*,/,%)

```
#include <iostream>
using namespace std;
int main()
{
int x; int y; int result;
cout << "Enter the value of x : ";
cin >> x;
cout << "Enter the value of y : ";
cin >> y;
result = x + y;
cout << "Result Addition = " << result<<endl;
result = x - y;
cout << "Result Subtraction= " <<result<<endl;
result = x * y;
cout << "Result Multiplication =" << result << endl;
result = x / y;
cout << "Result Division = “ <<result << endl;
}
```

Run the example in Visual Studio 2022



Run the example in Visual Studio 2022



Today, Agenda

- Logical Operators
- Decision Making
- Flowchart

Boolean Data Type in C++

- What is a bool?
  - A data type that holds true or false.
  - Used for conditions in decision-making.
- Declaration Examples:
  - `bool isRaining = true;`
  - `bool isSunny = false;`

What is a bool?

- Boolean Values Table:

Value	Meaning	Example Usage
true	Logical true	if (isValid) { ... }
false	Logical false	while (!done) { ... }

- Key Notes:
  - Non-zero values (e.g., 1, -5) implicitly convert to true.
  - Zero (0) converts to false.

Logical Operators in C++

- Logical Operators:
  - AND (&&): True if both conditions are true.
  - OR (||): True if at least one condition is true.
  - NOT (!): Inverts the boolean value.

A	B	A && B
true	true	true
true	false	false
false	true	false
false	false	false

A	B	A || B
true	true	true
true	false	true
false	true	true
false	false	false

A	!A
true	false
false	true

Introduction to Decision Making

- In real life, decisions depend on conditions.
- Programming needs decision-making to control the flow.
- C++ uses conditional statements to make decisions.
- Main types:
  - if statement
  - if-else statement
  - if-else-if ladder
  - switch statement (introduced later)

Syntax of if Statement in C++

```
if (condition) {
    // code block to be executed if condition is true
}
```

- If the condition is true → code inside runs.
- If the condition is false → nothing happens.

Example: Basic if Statement

```
int num = 10;
if (num > 0) {
    cout << "Number is positive!";
}
```

- Key Points:
  - condition must evaluate to true or false.
  - Use relational operators (>, <, ==, etc.) or boolean variables.

Comparison Operators

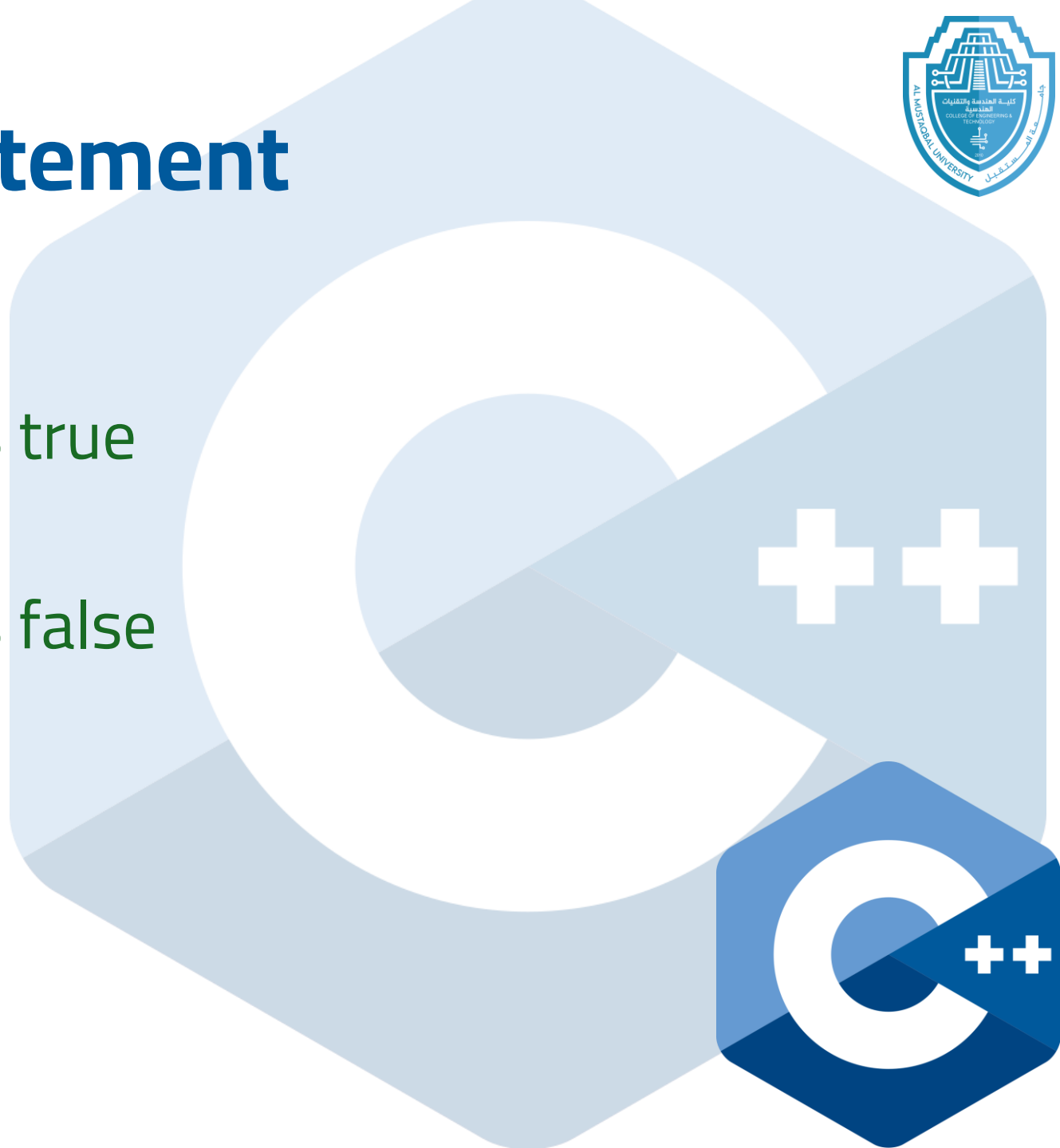
- Symbols used to compare values in conditions.
- Return true or false based on the comparison.

Operator	Name	Description	Example
==	Equal to	True if values are equal	5 == 5 → true
!=	Not equal to	True if values differ	3 != 5 → true
>	Greater than	True if left is larger	10 > 5 → true
<	Less than	True if left is smaller	3 < 2 → false
>=	Greater than or equal	True if left ≥ right	7 >= 7 → true
<=	Less than or equal	True if left ≤ right	4 <= 3 → false

Syntax of if-else Statement

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

- Two possible paths:
  - if → when true
  - else → when false



Analysis of *IF-ELSE*

- **Analysis**:
  - `if` : Checks a specific condition.
  - `else` : Executes if the condition in the IF statement is not met.

```
int age = 18;
if (age >= 18)
cout << "You are an adult.";
else
cout<<"You are a minor.";
```

Example: if-else

```
int num = 5;
if (num % 2 == 0) {
    cout << "Even";
}
else {
     cout << "Odd";
}
```

- Use Case:
  - Choose between two mutually exclusive paths.

else-if Ladder

- Syntax

```
if (condition1) { /* ... */ }
else if (condition2) { /* ... */ }
else { /* ... */ }
```

- Example:

```
int marks = 85;
if (marks >= 90) cout << "A";
else if (marks >= 80) cout << "B";
else if (marks >= 70) cout << "C";
else cout << "F";
```

Nested if Statements

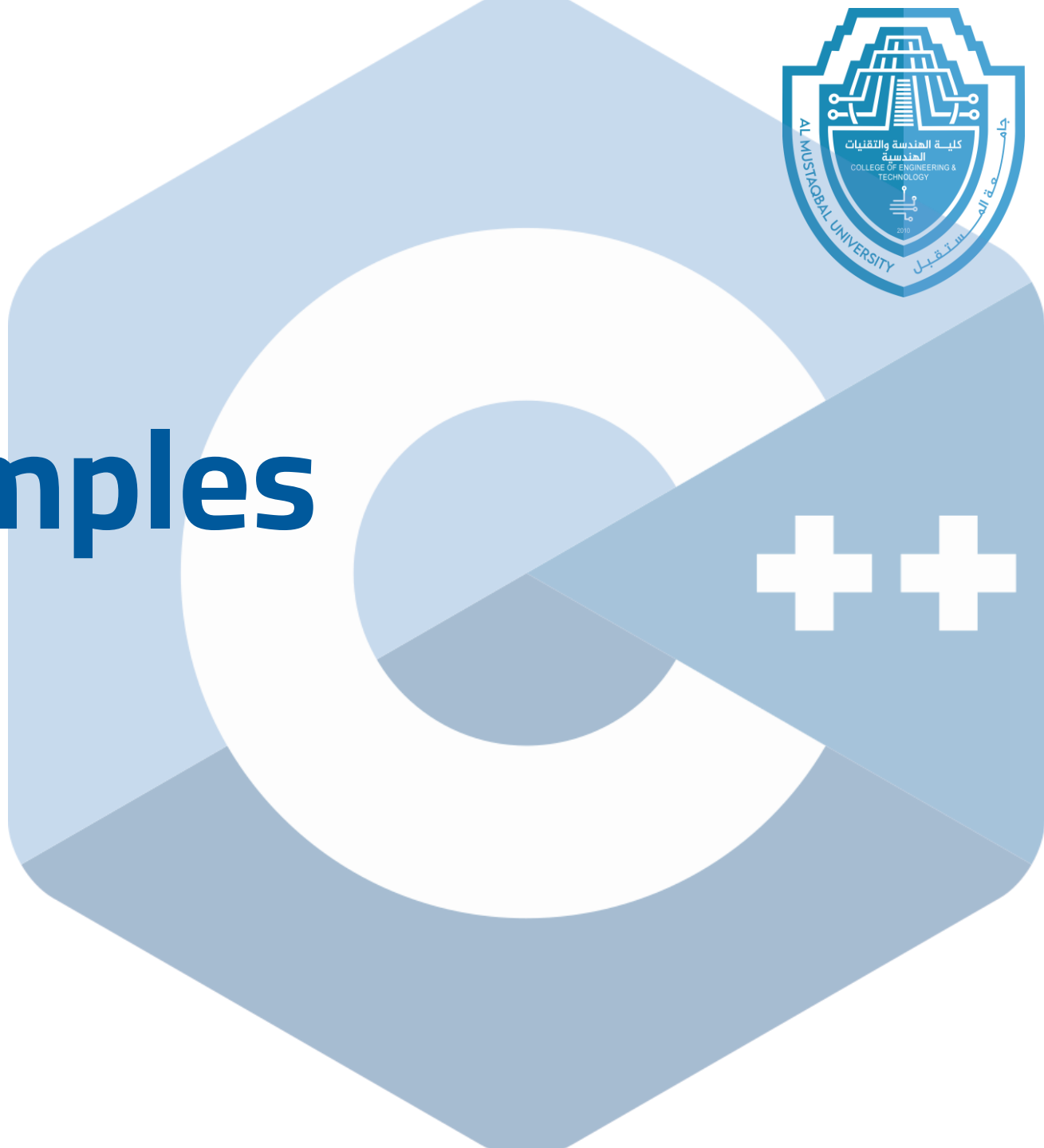
- Definition: An if inside another if/else block.
- Example

```
int age = 20;
    bool isMember = true;
    if (age >= 18) {
        if (isMember) {
            cout << "Access granted!";
        }
    }
```


AL MUSTAQBAL UNIVERSITY
جامعة المستقبل
أسست عام
2010

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

Examples



Example 1: find if two numbers are equal or which one is larger

```cpp
#include <iostream>
using namespace std;
int main()
{
    int a = 10, b = 20;
    if (a == b) {
        cout << "Equal";
    }
    else if (a > b) {
        cout << "a is larger";
    }
    else {
        cout << "b is larger";
    }
}
```

Example 2: Read a number and find if it is even

```
#include <iostream>
using namespace std;
int main() {
    int num;
    cout << "Enter an integer: ";
    cin >> num;
    if (num % 2 == 0) {
        cout << "The number is even.";
    }
}
```

Example 3: Check if a Character is Uppercase

```cpp
#include <iostream>
using namespace std;
int main() {
    char letter;
    cout << "Enter a letter: ";
    cin >> letter;
    if (letter >= 'A' && letter <= 'Z') {
        cout << "The letter is uppercase.";
    }
}
```

Example 4: Validate Age to Vote

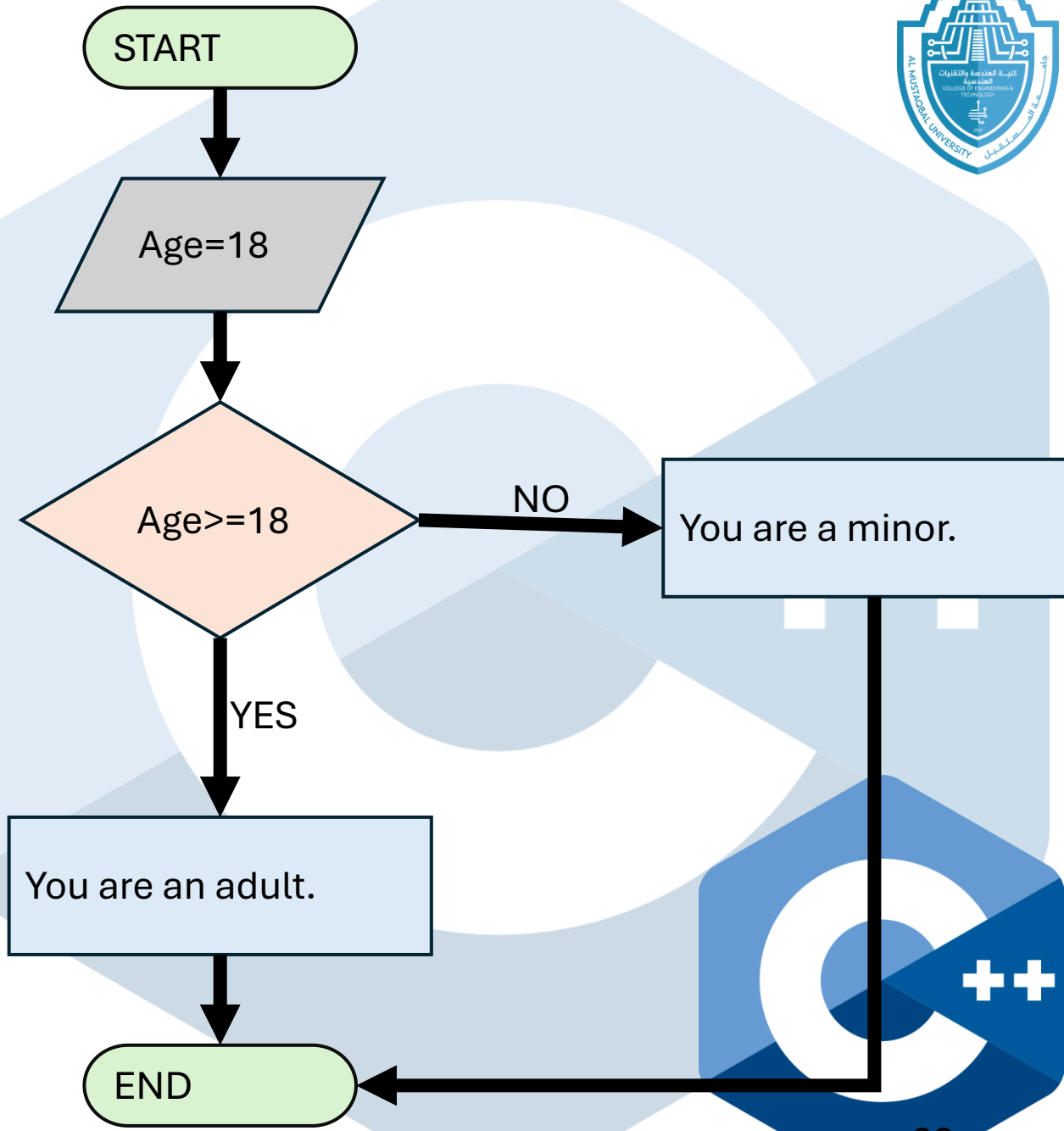
```cpp
#include <iostream>
using namespace std;
int main() {
    int age;
    cout << "Enter your age: ";
    cin >> age;
    if (age >= 18) {
        cout << "You are eligible to vote.";
    }
}
```

Example 5: Check if Number is Multiple of 5

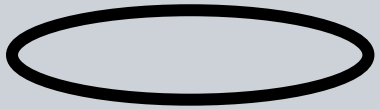
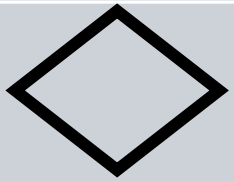
```cpp
#include <iostream>
using namespace std;
int main() {
    int num;
    cout << "Enter a number: ";
    cin >> num;
    if (num % 5 == 0) {
        cout << "The number is a multiple of 5.";
    }
    return 0;
}
```

Flowchart

- A flowchart is a type of diagram that represents a workflow or process.
- A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task.
- The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows.



Flowchart Symbols

Symbol	Name	Description
	Start/End	Oval (Terminator): Represents the start or end of a process
	Arrow	Represents the flow of the process, connecting different symbols in sequence
	Input/Output	Represents input (e.g., receiving data) or output (e.g., displaying results).
	Process	Rectangle (Process): Represents a process, task, or action to be performed.
	Decision	Represents a decision point where a condition is evaluated, and the flow branches based on the outcome (Yes/No).

Write a flowchart and program for the following scenario

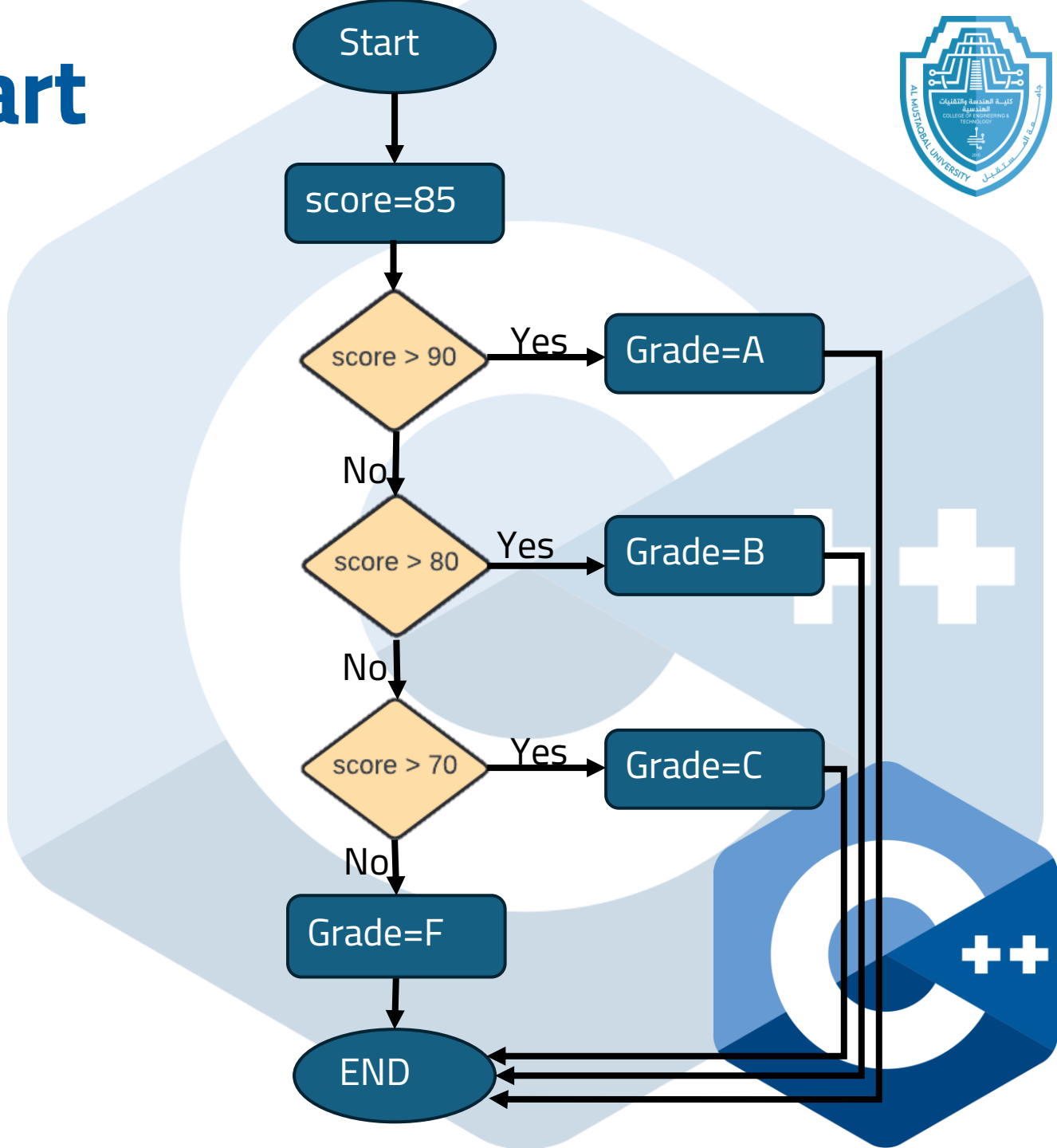
- Consider the following marking table, create a flowchart and program:

Score Details	Grade
More than 90	A
More than 80	B
More than 70	C
Otherwise	F

Example of Flowchart

Logical Analysis

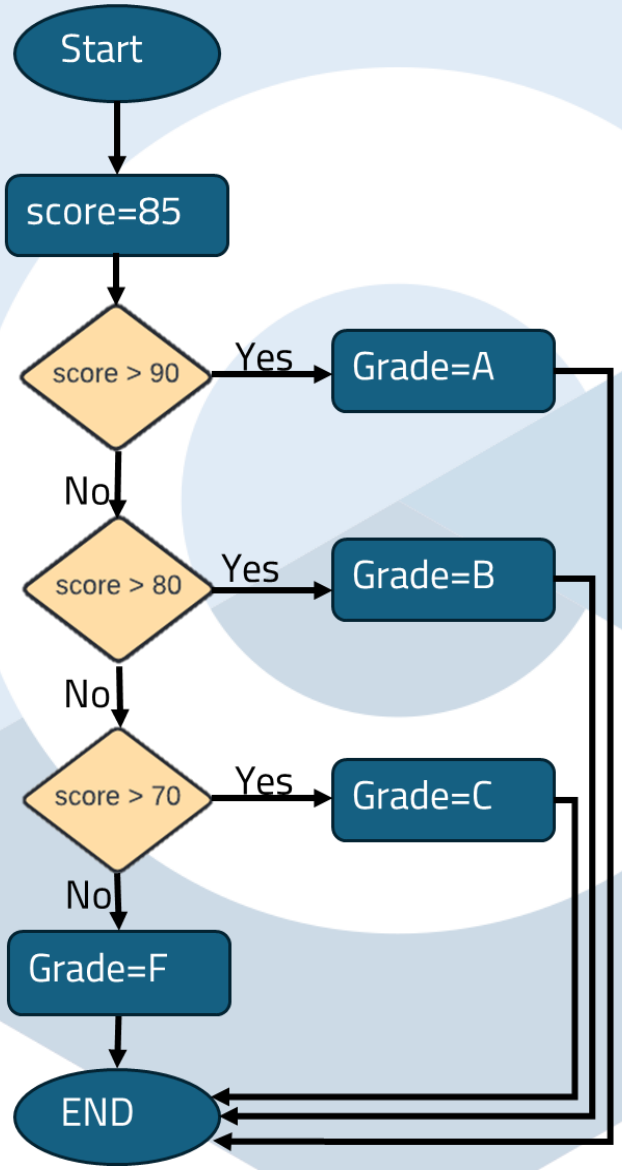
Score Details	Grade
More than 90	A
More than 80	B
More than 70	C
Otherwise	F



Convert Flowchart to program

```cpp
#include <iostream>
using namespace std;
int main() {

    int score = 85;
    char grade;
    if (score >= 90) {
        grade = 'A';
    }
    else if (score >= 80) {
        grade = 'B';
    }
    else if (score >= 70) {
        grade = 'C';
    }
    else {
        grade = 'F';
    }
    cout << "Your grade is: " << grade << endl;
}
```



Summary

- Logical operators (&&, ||, !) for complex logic.
- if-else is used for decision-making.
- Flowcharts help visualize control flow.
- Nested if-else is for more complex decisions.
- if-else-if ladder simplifies multiple conditions.
- Always write clean, readable conditions.

Let’s try C++

Install Visual Studio and familiarise yourself with its interface.