# **Bài 2: Cú Pháp Cơ Bản và Các Kiểu Dữ Liệu C++**

1. **Cách học và lấy tài liệu**
   * Tiếng Anh (Function, Method, Class, ERD, IDE)
   * Github
2. **Làm quen với Visual Studio Code**
   * **Mở visual studio**
   * **Thay đổi theme**
   * **Mở terminal**
   * **Auto lưu file**
3. **Giải thích cơ bản về những cú pháp**

**1. #include <iostream>**

* **Mục đích**: Dòng này yêu cầu trình biên dịch đưa vào mã nguồn của bạn các định nghĩa cần thiết từ thư viện iostream.
* **Giải thích**:
  + #include là một chỉ thị tiền xử lý (preprocessor directive). Nó thông báo cho trình tiền xử lý (preprocessor) của C++ để bao gồm nội dung của tệp tiêu đề <iostream> vào mã nguồn tại vị trí mà chỉ thị này xuất hiện.
  + Thư viện iostream chứa các định nghĩa cho các đối tượng và chức năng liên quan đến đầu vào và đầu ra (I/O), chẳng hạn như std::cout dùng để xuất dữ liệu ra màn hình và std::cin dùng để nhận dữ liệu từ người dùng.

**2. int main()**

* **Mục đích**: Đây là điểm khởi đầu của chương trình C++.
* **Giải thích**:
  + int là kiểu dữ liệu trả về của hàm main. Chương trình C++ yêu cầu hàm main phải trả về một giá trị nguyên (integer) để thông báo trạng thái kết thúc của chương trình cho hệ điều hành.
  + main là tên của hàm chính, và nó là hàm bắt buộc trong một chương trình C++. Khi bạn chạy chương trình, hệ điều hành sẽ bắt đầu thực thi từ hàm main.
  + Hàm main không nhận tham số nào trong ví dụ này, nhưng có thể nhận tham số nếu cần (chẳng hạn int main(int argc, char \*argv[])).

**3. std::cout**

* **Mục đích**: Đối tượng std::cout được sử dụng để xuất dữ liệu ra màn hình.
* **Giải thích**:
  + std::cout là một đối tượng thuộc không gian tên (namespace) std (chuẩn). Nó được định nghĩa trong thư viện iostream.
  + cout là viết tắt của "character output" và cho phép bạn in dữ liệu ra màn hình. Ví dụ này sử dụng cout để xuất chuỗi "Hello, World!" ra màn hình.

**4. std::endl**

* **Mục đích**: Đưa con trỏ ra dòng mới (newline) và xóa bộ đệm (flush the buffer).
* **Giải thích**:
  + std::endl là một đối tượng thuộc không gian tên std và được sử dụng để kết thúc dòng (new line) trong đầu ra.
  + Khi bạn sử dụng std::endl, nó không chỉ chèn ký tự dòng mới mà còn xóa bộ đệm đầu ra, điều này đảm bảo rằng tất cả dữ liệu đã được xuất ra màn hình ngay lập tức.

**5. return 0;**

* **Mục đích**: Trả về một giá trị từ hàm main để báo cáo trạng thái kết thúc của chương trình.
* **Giải thích**:
  + return 0; trong hàm main báo hiệu rằng chương trình đã kết thúc thành công. Giá trị 0 thường được dùng để chỉ rằng không có lỗi xảy ra.
  + Nếu return 0; không được chỉ định, nhiều trình biên dịch C++ sẽ mặc định trả về 0, nhưng việc chỉ định rõ ràng là một thực tiễn tốt để làm cho mã nguồn dễ hiểu hơn.

**Tóm Tắt**

* **#include <iostream>**: Bao gồm thư viện iostream để sử dụng các tính năng đầu vào và đầu ra.
* **int main()**: Hàm chính, điểm bắt đầu của chương trình.
* **std::cout**: Đối tượng để xuất dữ liệu ra màn hình.
* **std::endl**: Kết thúc dòng và xóa bộ đệm đầu ra.
* **return 0;**: Trả về giá trị 0 từ hàm main để báo hiệu kết thúc thành công của chương trình.

1. **Các kiểu dữ liệu, biến và hằng số**

**1. Các Kiểu Dữ Liệu Cơ Bản và Giới Hạn của Chúng**

* **int8\_t**, **uint8\_t**: Số nguyên có dấu và không dấu 8-bit.
  + **Kích thước**: 1 byte.
  + **Giá trị tối thiểu và tối đa**:
    - **int8\_t**: Từ -128 đến 127.
    - **uint8\_t**: Từ 0 đến 255.
  + **Ví dụ:**

cpp

Copy code

int8\_t smallNum = 127; // Giới hạn tối đa của int8\_t  
uint8\_t unsignedSmallNum = 255; // Giới hạn tối đa của uint8\_t

* **int16\_t**, **uint16\_t**: Số nguyên có dấu và không dấu 16-bit.
  + **Kích thước**: 2 bytes.
  + **Giá trị tối thiểu và tối đa**:
    - **int16\_t**: Từ -32,768 đến 32,767.
    - **uint16\_t**: Từ 0 đến 65,535.
  + **Ví dụ:**

int16\_t mediumNum = 32767; // Giới hạn tối đa của int16\_t  
uint16\_t unsignedMediumNum = 65535; // Giới hạn tối đa của uint16\_t

* **int32\_t**, **uint32\_t**: Số nguyên có dấu và không dấu 32-bit.
  + **Kích thước**: 4 bytes.
  + **Giá trị tối thiểu và tối đa**:
    - **int32\_t**: Từ 2,147,483,648 đến 2,147,483,647.
    - **uint32\_t**: Từ 0 đến 4,294,967,295.
  + **Ví dụ:**

int32\_t largeNum = 2147483647; // Giới hạn tối đa của int32\_t  
uint32\_t unsignedLargeNum = 4294967295; // Giới hạn tối đa của uint32\_t

* **int64\_t**, **uint64\_t**: Số nguyên có dấu và không dấu 64-bit.
  + **Kích thước**: 8 bytes.
  + **Giá trị tối thiểu và tối đa**:
    - **int64\_t**: Từ -9,223,372,036,854,775,808 đến 9,223,372,036,854,775,807.
    - **uint64\_t**: Từ 0 đến 18,446,744,073,709,551,615.
  + **Ví dụ:**

int64\_t veryLargeNum = 9223372036854775807LL; // Giới hạn tối đa của int64\_t  
uint64\_t unsignedVeryLargeNum = 18446744073709551615ULL; // Giới hạn tối đa của uint64\_t

* **float**: Số thực với độ chính xác đơn.
  + **Kích thước**: 4 bytes.
  + **Giá trị tối thiểu và tối đa**: Khoảng từ 1.5 × 10^−45 đến 3.4 × 10^38.
  + **Ví dụ:**

float num = 3.4e+38f; // Giới hạn tối đa của float

* **double**: Số thực với độ chính xác gấp đôi float.
  + **Kích thước**: 8 bytes.
  + **Giá trị tối thiểu và tối đa**: Khoảng từ 5.0 × 10^−324 đến 1.7 × 10^308.
  + **Ví dụ:**

double num = 1.7e+308; // Giới hạn tối đa của double

* **bool**: Giá trị đúng hoặc sai.
  + **Kích thước**: Thường là 1 byte.
  + **Giá trị**: true hoặc false.
  + **Ví dụ:**

bool flag = true;

**2. Biến và Hằng Số**

* **Biến**: Làm đại diện cho một giá trị có thể thay đổi.

int score = 50;

* **Hằng số**: Làm đại diện cho một giá trị không thay đổi.

const int MAX\_VALUE = 100;

1. **Các phép toán cơ bản**

**- Các Phép Toán Cơ Bản**

* **Phép cộng, trừ, nhân, chia**:

int a = 10;  
int b = 5;  
int sum = a + b;  
int difference = a - b;  
int product = a \* b;  
int quotient = a / b;

* **Phép toán số học nâng cao**:
  + Phép chia dư (%):

int remainder = a % b;

* **Các phép toán tăng/giảm**:

int count = 0;  
count++;  // Tăng giá trị  
count--;  // Giảm giá trị

1. **Câu lệnh nhập, xuất dữ liệu**

**Nhập dữ liệu từ bàn phím**:

int age;  
std::cout << "Enter your age: ";  
std::cin >> age;

**Xuất dữ liệu ra màn hình**:

std::cout << "You are " << age << " years old." << std::endl;