DAY 1:

Variables and Fundamental Data Types

variables: biến
Fundamental Data Types: các kiểu dữ liệu cơ bản

Section 1

C++ Basic

C++ Basic

- Statements and the structure of a program
- Comments
- Variables
- literals and operators

Statements and the structure of a program

- **Statements:** câu lệnh
  - ✓ A computer program is a sequence of instructions that tell the computer what to do. A **statement** is a type of instruction that causes the program to *perform some action*
- **Functions and the main function** hàm và hàm main
  - ✓ In C++, statements are typically grouped into units called functions. A **function** is a collection of statements that executes sequentially

trong C++, các câu lệnh thường được nhóm lại trong 1 đơn vị được gọi là hàm
hàm là mọt tập hợp các câu lệnh được kết nối và được thực thi tuần tự

Comments

- **Single-line comments**

```
// std::cout lives in the iostream library
std::cout << "Hello world!\n";
```

- **Multi-line comments**

```
/* This is a multi-line comment.
   This line will be ignored.
   So will this one. */
```

Variables

- **The name of a variable:**
  - ✓ Starts with an underscore “_” or a letter, lowercase or uppercase, such as a letter from a to z or from A to Z. Examples are Name, gender, _Students, pRice
  - ✓ Can include letters, underscore, or digits. Examples are: keyboard, Master, Junction, Player1, total_grade, _Score_Side1
  - ✓ Cannot include special characters such as !, %,], or $
  - ✓ Cannot include an empty space
  - ✓ Cannot be any of the reserved words
  - ✓ Should not be longer than 32 characters (although allowed)
- **C++ is case sensitive**

literals and operators

- **Literals** hằng số
  - ✓ A **literal** is a fixed value that has been inserted directly into the source code. hằng số là 1 giá trị cố định được chèn trực tiếp vào trong mã nguồn

```
std::cout << "Hello world!"; or int x = 5;
```

- **Operators** toán tử
  - ✓ The specific operation to be performed is denoted by a construct (typically a symbol or pair of symbols) called an **operator**

```
std::cout << 3 + 2;
```

phép toán cụ thể cần thực hiện được biểu thị bằng một cấu trúc (thường là một ký hiệu hoặc một cặp ký hiệu) gọi là toán tử

Section 2

Functions, Preprocessor and Files

Function (1)

- **User-Define a function**

```
return-type identifier()
{
        // Your code here
}
```

- **Calling functions**

```
void doPrint()
{
    std::cout << "In doPrint()\n";
}

// Definition of function main()
int main()
{
    doPrint(); // doPrint() called for the first time
    return 0;
}
```

các function phải trả về đúng kiểu trả về đã được khai báo ở đầu function (ngoại trừ function main vì đây là 1 function đặc biệt)

Function (2)

- **Function return values**

```
#include <iostream>

int returnFive()
{
    // the return statement indicates the specific value that will be returned
    return 5; // return the specific value 5 back to the caller
}

int main()
{
    std::cout << returnFive() << '\n';
    std::cout << returnFive() + 2 << '\n';
    return 0;
}
```

quá trình biên dịch một chương trình C++:
1. Tiền xử lý: trình biên dịch sẽ bắt đầu bằng việc đọc mã nguồn và thực hiện các chỉ thị tiền xử lý. Trình biên dịch sẽ thực hiện thay thế, định nghĩa các macro và điều khiển biên dịch có điều khinrj dựa trên các diều kiện đưiọc xác định trước khi mã C++ được biên dịch thực sự. quá trình này giúp chuẩn bị mã nguồn cho các bước biên dịch tiếp theo
2. biên dịch: mã C++ sau khi được tiền xử lý sẽ được biên dịch thành mã máy tương ứng cho các nền tảng mục tiêu
3. liên kết: nếu chương trình cảu bạn sử dụng các tập tin như thư viện bên ngoài, trình biên dịch sẽ liên kết các mã máy đã được biên dịch với các thư viện này để tạo ra một chương trình hoàn chỉnh có thể chạy
4. khởi chạy: thực thi các tệp thực thi trên máy tính

Preprocessor

bắt đầu bằng dấu #

- **Includes**

  #include <iostream>
- **Macro defines**

  #define identifier

  #define identifier substitution_text
- **Conditional compilation**

  #ifdef PRINT_JOE

  #ifdef PRINT_BOB
- **#if 0**

```
#if 0 // Don't compile anything starting here
      std::cout << "Bob\n";
      std::cout << "Steve\n";
#endif // until this point
```

phân biệt define và const , typedef:

+ #define PI 3.12159 : không phải là 1 phần của ngôn ngữ C++, nó sử dụng để định nghĩa các hằng số và macro, thuộc giai đoạn tiền xử lý, không được kiểm tra kiểu dữ liệu

+ const double PI = 3.14159; : là 1 phần của C++, được sử dụng để khai báo 1 biến có giá trị không thay đổi. Được kiểm tra kiểu dữ liệu bởi trình biên dịch.

+typedef int* IntPtr; : cũng là 1 phần của C++, sử dụng để đặt bí danh cho kiểu dữ liệu, tương tự const

conditionnal compilation: điều khiển biên dịch có điều kiện trong C++:

+có chức năng bao gồm hoặc loại trừ 1 phần của mã nguồn dựa trên 1 số điều kiện cụ thể;

```
#define DEBUG 1

#if DEBUG
std::cout << "Debug mode is on." << std::endl;
#else
std::cout << "Debug mode is off." << std::endl;
#endif
```

```
#define FEATURE_X

#ifdef FEATURE_X
std::cout << "Feature X is enabled." << std::endl;
#endif

#ifndef FEATURE_Y
std::cout << "Feature Y is not enabled." << std::endl;
#endif
```

Files (1)

- **Headers** (.h)

phần mở rộng

thông thường

đôi khi

mục đích chính

✓ C++ code files (with a .cpp extension) are not the only files commonly seen in C++ programs. The other type of file is called a **header file**. Header files usually have a .h extension, but you will occasionally see them with a .hpp extension or no extension at all. The primary purpose of a header file is to propagate declarations to code files.

truyền tải

các khai báo

conditional compilation

```
#ifndef REQUIRED_MACRO
#error "REQUIRED_MACRO must be defined."
#endif
```

```
#include <iostream>
// Định nghĩa macro nền tảng
#if defined(_WIN32) || defined(_WIN64)
#define PLATFORM_WINDOWS
#elif defined(__linux__)
#define PLATFORM_LINUX
#else
#error "Unsupported platform!"
#endif

int main() {
   #ifdef PLATFORM_WINDOWS
   std::cout << "Running on Windows." << std::endl;
   #endif
   #ifdef PLATFORM_LINUX
   std::cout << "Running on Linux." << std::endl;
   #endif
```

Files (2)

- **Using standard library header files**

```
#include <iostream>

int main()
{
    std::cout << "Hello, world!";
    return 0;
}
```

Files (3)

- **Header guards** bảo vệ tiêu đề

```
#ifndef SOME_UNIQUE_NAME_HERE
#define SOME_UNIQUE_NAME_HERE

// your declarations (and certain types of definitions) here
```

khai báo của bạn và 1 số loại định nghĩa ở đây

```
#endif
```

Section 3

Basic Data Types

Basic Data Type

nguyên thủy

- **Primitive Data Types:**

  Int, char, bool, float, double, void, wide character

- **Derived Data Types:** kiểu dữ liệu dẫn xuất

  Function, array, pointer, reference

- **User Defined Data Types:**

  Class, structure, Enum, Typedef

References

- https://www.learncpp.com/

Lesson Summary

- **C++ Basic**
  - ✓ Statements and the structure of a program
  - ✓ Comments
  - ✓ Variables
  - ✓ literals and operators
- **Functions, Preprocessor and Files**
- **Basic Data Types**
  - ✓ Primitive Data Types
  - ✓ Derived Data Types
  - ✓ User Defined Data Types

Thank you