

Tin học cơ sở 4

Lập trình căn bản C++ (INT 1006 7)

Học kì I năm học 2018-2019

UET – G2-107

Giảng viên: TS. Nguyễn Thị Ngọc Diệp
Email: ngocdiep@vnu.edu.vn
Tuần học: 1

Thông tin môn học

- Tên môn học: Tin học cơ sở 4
- Số tín chỉ: 3
- Lý thuyết: 10+1 tuần (20 tiết); Thực hành: 46 tiết (?)
- Đánh giá kết quả học tập:
 - Thi cuối kì: 60%
 - Chủ yếu về kĩ năng lập trình
 - Kiểm tra giữa kì: 40%
 - Tổng hợp từ bài tập thực hành hàng tuần (20%)
 - Một bài viết (20%)
 - Chuyên cần: khuyến khích (tối đa +20%)

Mục tiêu khoá học

- **Kiến thức:** Giới thiệu cái khái niệm cơ bản trong lập trình và ngôn ngữ lập trình
 - Dùng cú pháp của ngôn ngữ C++
 - Tiền đề để học tiếp các môn: *Lập trình nâng cao, Lập trình hướng đối tượng, Nguyên lý lập trình...*
- **Kĩ năng:** Người học bắt đầu viết được các chương trình đơn giản bằng C++
- Khuyến khích:
 - Thảo luận trong và ngoài giờ
 - Tự học
- Bắt buộc:
 - Ghi chú rõ ràng về nguồn gốc của các trích dẫn từ sách hoặc của người khác
 - **Không sao chép** khi thi

Bí quyết của giới lập trình

80% thời gian là tự học, tự code nhiều

20% thời gian là học cách code (đẹp) của người khác

Tài liệu tham khảo

- Website của bộ môn: <https://uetcodehub.xyz/course/view.php?id=34>
 - Đăng nhập: usr/pwd = mã sv/mã sv
- Sách: "Programming Principles and Practice Using C++," Bjarne Stroustrup, 4th edition, 2013.
 - Cha đẻ của C++
 - Cách phát âm: Bjar-ne Strou-strup ☺
- Website miễn phí: <http://www.learncpp.com/>

Khuyến khích: - đọc tài liệu bằng tiếng Anh để có thêm thuật ngữ + hữu ích khi gỡ lỗi
 - tự học trên các forum
 - tự tìm kiếm tài liệu dùng Google

Lập trình căn bản?

Từ cơ bản:



Đến phức tạp



Đến kiểu dáng + chức năng



Kiểu dữ liệu
int, float, char, ...

Gán giá trị
 $x = 10$

Phép toán
+, -, *, /, %, ...

Nhập, xuất

Cấu trúc điều khiển
if ... else, for, while, ...

Mảng
int [5], ...

Hàm, mô-đun, lớp

Nhập/xuất tệp văn bản

Cấu trúc dữ liệu

Quản lý bộ nhớ

Thư viện

Mẫu thiết kế

Quản lý đồ họa

...



Image credits: from Google Image search

Lập trình căn bản ở INT 1006

Từ cơ bản:



Đến phức tạp



Đến kiểu dáng + chức năng



Kiểu dữ liệu
int, float, char, ...

Gán giá trị
 $x = 10$

Phép toán
+, -, *, /, %, ...

Nhập, xuất

Cấu trúc điều khiển
if ... else, for, while, ...

Mảng
int [5], ...

Hàm, mô-đun, lớp

Nhập/xuất tệp văn bản

Cấu trúc dữ liệu

Quản lý bộ nhớ

Thư viện

Mẫu thiết kế

Quản lý đồ họa

...



Image credits: from Google Image search

Nội dung các tuần

- Tuần 1: Giới thiệu chương trình và lập trình
- Tuần 2: Biến và các phép toán (kiểu và giá trị) ← Đơn vị cơ bản của tính toán
- Tuần 3, 4: Cấu trúc điều khiển
 - Chia nhánh: if...else, switch ← Lúc nào làm gì
 - Lặp: for, while ← Làm các việc lặp lại
- Tuần 5: Mảng ← Nhóm các thành phần giống nhau
- Tuần 6: Nhập xuất với tệp ← Làm việc với bộ nhớ ngoài
- Tuần 7: Hàm, mô-đun chương trình ← Tạo nhóm chức năng tính toán
- Tuần 8: Con trỏ ← Làm việc với địa chỉ của biến
- Tuần 9: Cấu trúc, lớp ← Tạo nhóm mẫu của đơn vị và chức năng tính toán
- Tuần 10: Mẫu hàm, mẫu lớp, thư viện STL
- Tuần 11 : Tóm tắt và ôn tập

Questions?

Nội dung buổi học này

1. Khái niệm chương trình và lập trình
2. Môi trường lập trình
3. Cấu trúc một chương trình
4. Biên dịch và chạy chương trình

Chương trình (a program)

- Máy tính không thể tự vận hành
 - Hoạt động theo các chương trình được lập trình sẵn
- Một chương trình là một dãy các lệnh (instructions) cho máy tính
- Máy tính chạy các lệnh của chương trình tại bộ xử lý trung tâm (CPU)

Python:

Chương trình

```
In [1]:
x = 1
y = 2
print "Sum of x + y is: ", x + y
```

Kết quả

```
Sum of x + y is: 3
```

C++:

Chương trình

```
1 #include <iostream>
2
3 int main()
4 {
5     int x = 1;
6     int y = 2;
7     std::cout << "Sum of x + y is: " << x + y << "\n";
8     return 0;
9 }
10
```

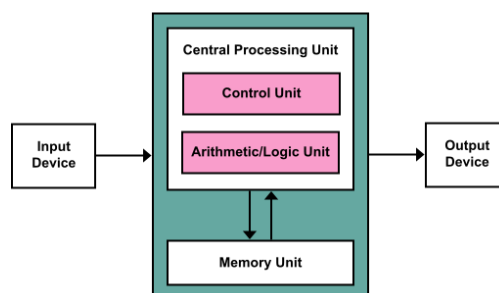
Kết quả

```
Sum of x + y is: 3
Exit code: 0 (normal program termination)
```

Q: Có nhận xét về sự khác biệt giữa cú pháp (cách viết) của hai chương trình?

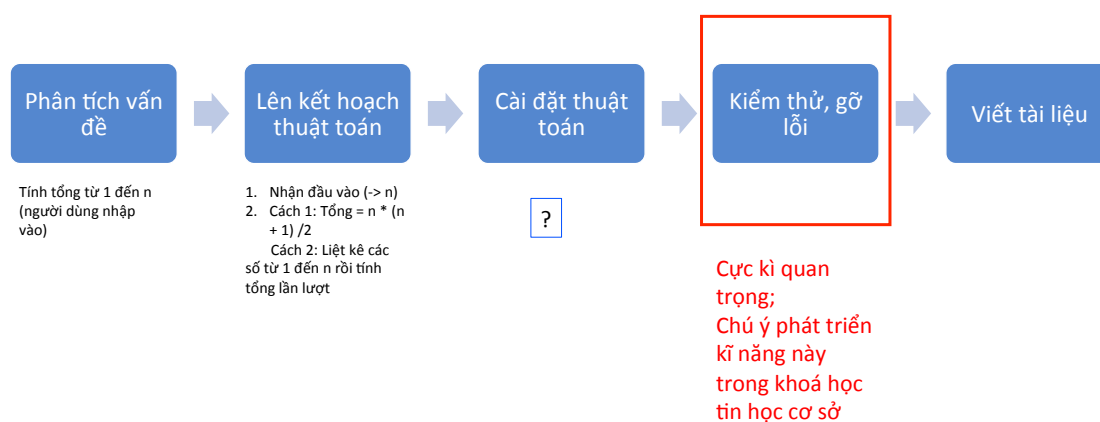
Tìm hiểu thêm

- Kiến trúc Von Neumann
- Lập trình
 - “Programming” hay “coding”?
- Máy Turing



Source: Wikipedia

5 bước cơ bản trong lập trình



Môi trường lập trình

- Các công cụ giúp xây dựng chương trình
 - Soạn thảo
 - Biên dịch
 - Chạy chương trình
 - Kiểm thử
 - Gỡ lỗi
- Sinh viên có thể phối hợp các công cụ riêng lẻ
 - Ví dụ: Notepad (hay Vim hay Emacs) + dòng lệnh
- Hoặc dùng môi trường phát triển tích hợp (IDE) ← Dùng trong các tiết thực hành
 - Ví dụ: Code::Blocks, Dev-C++, Visual Studio, ...
- Sinh viên có thể thực hiện từng bước, hoặc cả gói
 - Biên dịch rồi chạy luôn
 - Biên dịch xong chạy kiểm thử

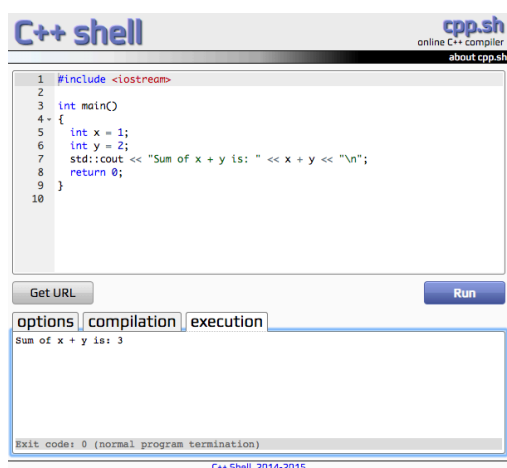
Ví dụ môi trường lập trình (1)

Dùng Vim để soạn thảo

Dùng Atom để soạn thảo

Dùng dòng lệnh để mở tệp chương trình, biên dịch và chạy

Ví dụ môi trường lập trình (2)



Dùng môi trường online để soạn thảo và chạy (bao gồm biên dịch)

Chương trình đầu tiên

Yêu cầu: máy tính in ra Hello, World!!!

```

helloworld.cpp
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     cout << "Hello, World!!!" << endl;
8     return 0;
9 }
10

```


Phân tích chương trình

```

helloworld.cpp
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      cout << "Hello, World!!!" << endl;
8      return 0;
9  }
10

```

Khai báo thư viện

Khai báo không gian tên

Khai báo hàm main()

Mở đầu khối lệnh

Câu lệnh in và xuống dòng

Kết thúc khối lệnh

★ Có thể viết
#include "iostream"
nhưng nghĩa sẽ khác!

Khai báo thư viện

- Cú pháp: `#include <tên_thư_viện>`
- Thư viện = tập hợp các cấu trúc lập trình đã được viết sẵn
 - Dữ liệu, mô-đun, hàm, mã lệnh, tài liệu
 - Ai viết?
- Thư viện **iostream**
 - Nhập/xuất dữ liệu chuẩn
 - In ra màn hình: `cout`
 - Đọc từ bàn phím: `cin`
 - Xuất lỗi: `cerr`
 - Xuất chuẩn cho log: `clog`
 - Tham khảo: <http://www.cplusplus.com/reference/iostream/>

Khai báo không gian tên

- Cú pháp: `using namespace không_gian_tên;`
- Không gian tên (namespace): tập hợp các tên
 - Tên hàm, biến, lớp, ...
- Giúp tổ chức chương trình
 - Phân biệt các tên trùng nhau
- Không gian tên `std`
 - Chứa các tên `cout`, `endl`
 - Nếu không khai báo, cần viết: `std::cout`, `std::endl`

★ Khi lập trình nâng cao hơn, không khuyến khích dùng khai báo không gian tên toàn cục như thế này!
(vì khả năng xảy ra xung đột tên)
Xem thêm: <https://www.learncpp.com/cpp-tutorial/4-3c-using-statements/>

Khai báo không gian tên – cách khác

```

1  #include <iostream>
2
3  int main()
4  {
5      using std::cout; // this using declaration tells the compiler that cout
6      // should resolve to std::cout
7      cout << "Hello world!\n"; // so no std:: prefix is needed here!
8      return 0;
9  }

```

```

1  #include <iostream>
2
3  int main()
4  {
5      using namespace std; // this using directive tells the compiler that we're
6      // using everything in the std namespace!
7      cout << "Hello world!\n"; // so no std:: prefix is needed here!
8      return 0;
9  }

```

Hàm main()

- Mỗi chương trình cần một hàm `main()`
- Mã lệnh trong hàm `main()` được chạy trước tiên
- Khai báo hàm `main()`

Kết quả là số nguyên

```
int main()
{
    ...
    return 0;
}
```

`()`: hàm không có tham số

Giá trị trả về cho hệ điều hành
0: chương trình không có lỗi
1, -1, ...: chương trình có lỗi

- Sẽ tìm hiểu kĩ hơn ở tuần 7

Câu lệnh in ra màn hình

```
cout << "Hello, World!!!" << endl ;
```

Đối tượng
đại diện cho
luồng xuất
chuẩn

Toán tử đẩy
trái: ghi ra
`cout`

Dữ liệu cần ghi là một xâu kí tự
(trong 2 dấu nháy kép)

Toán tử đẩy
trái: ghi ra
`cout`

Đối tượng
đại diện cho
một dấu
xuống dòng

Dấu chấm
phẩy báo
hiệu kết thúc
câu lệnh

★ Trong C++, mỗi câu lệnh phải kết thúc bằng dấu chấm phẩy!

In nhiều dòng liên tục

```
cout << "Hello, World!!!" << endl;
cout << "Introduction to Programming" << endl;
```

```
cout << "Hello, World!!!" << endl
     << "Introduction to Programming" << endl;
```

Một câu lệnh duy nhất, không có dấu chấm phẩy

Biên dịch dùng câu lệnh

- Soạn thảo file chương trình
- Ghi xuống đĩa thành file có đuôi **.cpp**
- Biên dịch:

```
$ g++ -c main.cpp
```

Dịch mã nguồn sang mã máy
(object file .o)

```
$ g++ main.o -o main
```

Liên kết mã máy để tạo ra 1 file
thực thi duy nhất có thể chạy

Hoặc là

```
$ g++ main.cpp -o main
```

```
$ ./main
```

Chạy chương trình

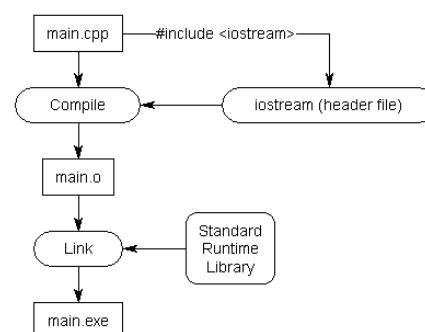


Diagram source: <https://www.learncpp.com/cpp-tutorial/19-header-files/>

Thử gỡ lỗi

- Thiếu dấu chấm phẩy sau câu lệnh
- Sai từ khoá
- Không khai báo thư viện
- Không khai báo không gian tên
- Thiếu dấu báo khối lệnh
- ...

Ví dụ lập trình dùng hàm

```
1  #include <iostream>
2
3  using namespace std;
4
5  void printHelloWorld()    Khai báo hàm
6  {
7      cout << "Hello, World !!!" << endl;    Chuyển lệnh vào trong hàm
8  }
9
10
11 int main()
12 {
13     printHelloWorld();    Gọi hàm
14     return 0;
15 }
16
```

Lập trình dùng hàm và nhập từ bàn phím

```
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  void printName(string name)  Khai báo hàm với tham số là chuỗi ký tự
7  {
8      cout << "Hello, " << name << endl;
9  }
10
11 int main()
12 {
13     string name;  Khai báo biến
14     cout << "What is your name? \n";
15     getline(cin, name);  Nhập từ bàn phím
16     printName(name);
17 }
18
19
```

★ Cách này khá là “đắt đỏ” về bộ nhớ
Sẽ học kĩ hơn khi làm việc với con trỏ (tuần 8)

Questions?