 **PHƯƠNG PHÁP LẬP TRÌNH**

SỐ TIẾT 45 LT- 60TH  
GV: Nguyễn Văn Thắng

Giảng Viên: Nguyễn Văn Thắng 1

---

---

---

---

---

---

---

---

 **CHƯƠNG I: MỞ ĐẦU**

Giảng Viên: Nguyễn Văn Thắng 2

---

---

---


---

---

---

---

---

 **NỘI DUNG**

1. Các khái niệm cơ bản
2. Các thành phần cơ bản trong ngôn ngữ C++
3. Các kiểu dữ liệu trong C++
4. Biến, hằng, cách khai báo
5. Lệnh xuất nhập
6. Định dạng kết quả xuất

Giảng Viên: Nguyễn Văn Thắng 3

---

---

---

---

---

---

---

---

 **CÁC KHÁI NIỆM CƠ BẢN**

Giảng Viên: Nguyễn Văn Thắng 4

---

---

---


---

---

---

---

---

 **Lập trình**

- Lập trình máy tính (*programming*) gọi tắt là **lập trình** là kỹ thuật cài đặt một hoặc nhiều **thuật toán** trừu tượng có liên quan với nhau bằng một hoặc nhiều **ngôn ngữ lập trình** để tạo ra một **chương trình máy tính**.

Giảng Viên: Nguyễn Văn Thắng 5

---

---

---


---

---

---

---

---

 **Thuật toán (giải thuật)**

- Thuật toán là một bộ các qui tắc hay qui trình cụ thể nhằm giải quyết một vấn đề trong một số bước hữu hạn

Giảng Viên: Nguyễn Văn Thắng 6

---

---

---

---

---

---

---

---

**Ví dụ:**

Thuật toán để giải phương trình bậc nhất  $P(x): ax + b = c$ , ( $a, b, c$  là các số thực), có thể là một bộ các bước sau đây:

- Nếu  $a = 0$ 
  - $b = c$  thì  $P(x)$  có nghiệm bất kì
  - $b \neq c$  thì  $P(x)$  vô nghiệm
- Nếu  $a \neq 0$ 
  - $P(x)$  có duy nhất một nghiệm  $x = (c - b)/a$

Giảng Viên: Nguyễn Văn Thắng 7

---

---

---

---

---

---

---

---

**Ngôn ngữ lập trình**  
(*programming language*):

- Là một tập con của ngôn ngữ máy tính, là một hệ thống được ký hiệu hóa để miêu tả những tính toán (qua máy tính) trong một dạng mà cả con người và máy đều có thể đọc và hiểu được.
- Một tập hợp các chỉ thị được biểu thị nhờ ngôn ngữ lập trình để thực hiện các thao tác máy tính nào đó thông qua một **chương trình**

Giảng Viên: Nguyễn Văn Thắng 8

---

---

---

---

---

---

---

---

**Ngôn ngữ lập trình**  
(*programming language*):

- Ngôn ngữ lập trình có thể phân theo 2 cách:
  - Ngôn ngữ cấp cao – ngôn ngữ cấp thấp
  - Ngôn ngữ hướng thủ tục – ngôn ngữ hướng đối tượng
- Ngôn ngữ c++ là ngôn ngữ bậc cao có chứa các đặc trưng của cả 2 loại ngôn ngữ: hướng thủ tục và hướng đối tượng

Giảng Viên: Nguyễn Văn Thắng 9

---

---

---

---

---

---

---

---

**Lớp và đối tượng**

- **Lớp** được phân loại dựa vào hai đặc điểm:
  - Dữ liệu
  - Phương pháp có thể áp dụng cho dữ liệu này
- Một **đối tượng** là một trường hợp cụ thể của lớp

Giảng Viên: Nguyễn Văn Thắng 10

---

---

---

---

---

---

---

---

**CÁC THÀNH PHẦN CƠ BẢN TRONG CHƯƠNG TRÌNH C++**

Giảng Viên: Nguyễn Văn Thắng 11

---

---

---

---

---

---

---

---

**Bộ ký tự dùng trong ngôn ngữ C++**

- Các chữ cái : A, B, C ..., Z, a,b,c,...z
- Các chữ số : 0,1,..., 9.
- Ký tự gạch nối \_ (underscore, chú ý phân biệt dấu -).
- Dấu cách ( space ) : dùng để phân biệt các từ

Ví dụ :  
lop Hoc ( 7 ký tự ) còn LopHoc( 6 ký tự).

Giảng Viên: Nguyễn Văn Thắng 12

---

---

---

---

---

---

---

---

**Tên(định danh identifier)**

- Tên ( định danh ) : là 1 dãy kí tự bắt đầu bằng chữ hoặc ký tự gạch dưới, theo sau là chữ cái, chữ số hoặc ký tự gạch nối underscore (\_).
- Tên : dùng làm tên hằng, tên biến, nhân, tên hàm....

Ví dụ :

- Tên đúng : \_abc, Delta\_1, BETA.
- Tên sai : 1xyz ( vì bắt đầu là 1 chữ số )  
A#B ( vì có dấu # )  
X-1 ( vì sử dụng dấu gạch ngang).

Giảng Viên: Nguyễn Văn Thắng 13

---

---

---

---

---

---

---

---

**Tên(định danh identifier)**

- Chú ý :
- + Tên : chữ hoa và chữ thường được xem là khác nhau
- + Thông thường :
  - . Đặt chữ hoa cho các hằng, chữ thường cho các đại lượng còn lại (biến, hàm..).
  - . Nên đặt 1 cách gọi nhớ ( 8 kí tự đầu là có nghĩa và tùy thuộc chương trình ).

Giảng Viên: Nguyễn Văn Thắng 14

---

---

---

---

---

---

---

---

**Tên(định danh identifier)**

Ví dụ:

```
int x, y;
int HeA, HeB, HeC;
float u, v;
...
```

Giảng Viên: Nguyễn Văn Thắng 15

---

---

---

---

---

---

---

---



**Từ khóa (keyword)**

- Một số từ được giữ bởi C++ cho một số mục đích riêng và không thể được dùng cho các định danh gọi là từ khóa

Giảng Viên: Nguyễn Văn Thắng 16

---

---

---

---

---

---

---

---

**Từ khóa (keyword)**

asm continue float new signed try  
 auto default for operator sizeof  
 typedef  
 break delete friend private static  
 union  
 case do goto protected struct unsigned  
 catch double if public switch  
 virtual  
 char else inline register template void  
 class enum int return this volatile  
 const extern long short throw while

Giảng Viên: Nguyễn Văn Thắng 17

---

---

---

---

---

---

---

---

**Chú thích trong chương trình**

- Chú thích được dùng để giải thích một vài khía cạnh của chương trình. Trình biên dịch bỏ qua hoàn toàn các chú thích trong chương trình.
- C++ cung cấp hai loại chú thích:
- Chú thích trên 1 dòng: //
- Chú thích trên nhiều dòng: /\* và \*/

Giảng Viên: Nguyễn Văn Thắng 18

---

---

---

---

---

---

---

---

**Các ký tự đặc biệt**

- \' : dấu nháy đơn
- \" : dấu nháy kép
- \\ : dấu chéo ngược
- \n : ký tự xuống dòng
- \0 : ký tự rỗng ( null)
- \t : phím tab
- \f : trang kế tiếp

Giảng Viên: Nguyễn Văn Thắng 19

---

---

---

---

---

---

---

---

**Kiểu dữ liệu trong C++**

Giảng Viên: Nguyễn Văn Thắng 20

---

---

---

---

---

---

---

---

**Kiểu số nguyên**

Name	Description	Size	Range*
char	Character or small integer.	1byte	signed: -128 to 127 unsigned: 0 to 255
short int (short)	Short Integer.	2bytes	signed: -32768 to 32767 unsigned: 0 to 65535
int	Integer.	4bytes	signed: -2147483648 to 2147483647 unsigned: 0 to 4294967295
long int (long)	Long integer.	4bytes	signed: -2147483648 to 2147483647 unsigned: 0 to 4294967295

Giảng Viên: Nguyễn Văn Thắng 21

---

---

---

---

---

---

---

---

**Kiểu số thực**

Name	Description	Size	Range*
float	Floating point number.	4bytes	3.4e +/- 38 (7 digits)
double	Double precision floating point number.	8bytes	1.7e +/- 308 (15 digits)
long double	Long double precision floating point number.	10bytes	1.2e +/- 4932 (19 digits)

Giảng Viên: Nguyễn Văn Thắng 22

---

---

---

---

---

---

---

---

---

---

**Kiểu bool**

bool	Boolean value. It can take one of two values: true or false.	1byte	true or false
------	--	-------	---------------

Giảng Viên: Nguyễn Văn Thắng 23

---

---

---

---

---

---

---

---

---

---

**BIÊN-HẲNG – CÁCH KHAI BÁO**

Giảng Viên: Nguyễn Văn Thắng 24

---

---

---

---

---

---

---

---

---

---



**BIÊN-CÁCH KHAI BÁO BIÊN**

Biến đại lượng thay đổi, mỗi biến có 1 tên và địa chỉ vùng nhớ danh riêng cho nó.

- CÁCH KHAI BÁO BIÊN:**  
**< Kiểu dữ liệu > < Danh sách các biến >;**

Ví dụ:

```
int a;
float mynumber;
```

Giảng Viên: Nguyễn Văn Thắng 25

---

---

---

---

---

---

---

---

**BIÊN-CÁCH KHAI BÁO BIÊN**

- Nếu các biến có cùng kiểu dữ liệu thì có thể khai báo:  
**<kiểu dữ liệu><danh sách các biến cách nhau bởi dấu phẩy>;**

Ví dụ:

```
int a, b, c;
```

Giảng Viên: Nguyễn Văn Thắng 26

---

---

---

---

---

---

---

---

**BIÊN-CÁCH KHAI BÁO BIÊN**

- PHẠM VI CỦA BIÊN**

```
#include <iostream>
using namespace std;

int Integer;
char aCharacter;
char string [20];
unsigned int NumberOfSons;

int main ()
{
    unsigned short Age;
    float Number, AnotherOne;

    cout << "Enter your age:"
    cin >> Age;
    ...
}
```

Global variables

Local variables

Instructions

ng 27

---

---

---

---

---

---

---

---

**BIÊN-CÁCH KHAI BÁO BIÊN**

- KHỞI TẠO GIÁ TRỊ CHO BIÊN  
`type identifier = initial_value ;`

Ví dụ:

```

int a = 0;
  
```

Diagram illustrating the components of the variable declaration `int a = 0;`:

- `int` is labeled as **type**.
- `a` is labeled as **identifier**.
- `= 0` is labeled as **initial\_value**.

Giảng Viên: Nguyễn Văn Thắng 28

---

---

---

---

---

---

---

---

---

---

**VÍ DỤ**

- `#include <iostream>`
- `using namespace std;`
- `int main ()`
- `{`
- `int a=5; // initial value = 5`
- `int b(2); // initial value = 2`
- `int result; // initial value undetermined`
- `a = a + 3;`
- `result = a - b;`
- `cout << result;`
- `return 0;`
- `}`

Giảng Viên: Nguyễn Văn Thắng 29

---

---

---

---

---

---

---

---

---

---

**HẲNG – CÁCH KHAI BÁO HẲNG**

**HẲNG (Constants):** Là một đại lượng mà giá trị của nó không đổi

Giảng Viên: Nguyễn Văn Thắng 30

---

---

---

---

---

---

---

---

---

---

**Hằng nguyên ( Int )**

- q Có giá trị từ -32768 đến 32767
- q Có thể viết theo hệ 16 bằng cách thêm tiền tố Ox
- q Hoặc theo cơ số 8 bằng cách thêm tiền tố O

**Ví dụ :**

$O306_8 = 6 * 8^0 + 0 * 8^1 + 3 * 8^2 = 198$  trong hệ 10

$O345_8 = 5 * 8^0 + 4 * 8^1 + 3 * 8^2 = 29$  trong hệ 10

$Ox147 = 7 * 16^0 + 4 * 16^1 + 1 * 16^2 = 327$  trong hệ 10

Giảng Viên: Nguyễn Văn Thắng 31

---

---

---

---

---

---

---

---

**Hằng long Int**

- Giống như hằng nguyên, chỉ khác thêm L hoặc l ở cuối.
- Ví dụ:

75u // unsigned int

75l // long

75ul // unsigned long

Giảng Viên: Nguyễn Văn Thắng 32

---

---

---

---

---

---

---

---

**Hằng thực ( float và double )**

Có 2 cách viết

- Cách 1 : dạng thập phân: phần nguyên, dấu chấm thập phân và phần phân.
- \* Ví dụ : 214.35 , - 234.34.
- Cách 2 : Viết theo dạng khoa học
- \* Ví dụ :

$1.543e7 = 15430000$

$123.456E-4 = 0.123456 ( 123.456/105)$

Giảng Viên: Nguyễn Văn Thắng 33

---

---

---

---

---

---

---

---

**Hằng ký tự**

- Viết trong 2 dấu nháy kép. Giá trị của hằng chính là mã ASCII của chữ.
- Ví dụ :  
'A' = 65;  
'd' = 100,  
'9' - '0' = 57 - 48 = 9

Giảng Viên: Nguyễn Văn Thắng 34

---

---

---

---

---

---

---

---

**Định nghĩa hằng**

Cú pháp:

```
#define identifier value
```

Ví dụ:

```
#define PI 3.14159265  
#define NEWLINE '\n'
```

Giảng Viên: Nguyễn Văn Thắng 35

---

---

---

---

---

---

---

---

**Ví dụ**

```
// defined constants: calculate circumference  
#include <iostream>  
using namespace std;  
#define PI 3.14159  
#define NEWLINE '\n'  
int main ()  
{  
    double r=5.0;           // radius
```

Giảng Viên: Nguyễn Văn Thắng 36

---

---

---

---

---

---

---

---

**Ví dụ**

```
double circle;
circle = 2 * PI * r;
cout << circle;
cout << NEWLINE;
getch();
return 0;
}
```

Giảng Viên: Nguyễn Văn Thắng 37

---

---

---

---

---

---

---

---

**Khai báo hằng (const)**

Cú pháp:  
**const** <kiểu dữ liệu> tên hằng = giá trị

Ví dụ:  
**const** int pathwidth = 100;  
**const** char tabulator = '\t';

Giảng Viên: Nguyễn Văn Thắng 38

---

---

---

---

---

---

---

---

**CÁC LỆNH XUẤT NHẬP CHUẨN**

Giảng Viên: Nguyễn Văn Thắng 39

---

---

---

---

---

---

---

---



**Lệnh xuất chuẩn (cout)**

- Cú pháp:  
`cout<<biểu thức;`

Trong đó :

Biểu thức có thể là:

- Biến
- Hằng
- Chuỗi

Giảng Viên: Nguyễn Văn Thắng 40

---

---

---

---

---

---

---

---

**Lệnh xuất chuẩn (cout)**

- Ví dụ:  
`cout << "This is a sentence.";`  
`cout << "This is another sentence.";`  
`cout << "Second sentence.\nThird sentence.";`

Giảng Viên: Nguyễn Văn Thắng 41

---

---

---

---

---

---

---

---

**Lệnh nhập chuẩn (cin)**

- Cú pháp:  
`cin>>tên biến;`

ví dụ:

```
int a;  
cout<<"nhap gia tri cho a: ";  
cin>>a;  
cout<<"gia tri cua a la :"<<a;
```

Giảng Viên: Nguyễn Văn Thắng 42

---

---

---

---

---

---

---

---

**Định dạng kết quả xuất**

- setw(n): ấn định độ rộng để in một số là n
- setprecision(n): ấn định số chữ số thập phân sẽ hiển thị.
- dec: hiển thị giá trị số ở hệ thập phân
- hex: hiển thị giá trị số ở hệ thập lục phân
- oct : hiển thị giá trị số ở hệ bát phân

Giảng Viên: Nguyễn Văn Thắng 43

---

---

---

---

---

---

---

---

**Định dạng kết quả xuất**

- Lưu ý: muốn sử dụng các định dạng thì trong chương trình ta phải khai báo :  
`#include <iomanip.h>`

Giảng Viên: Nguyễn Văn Thắng 44

---

---

---

---

---

---

---

---

**Ví dụ**

- #include <iostream.h>
- #include <iomanip.h>
- void main()
- {
- cout <<setw(5) <<12; //kết quả "...12"
- cout <<setw(10); cout <<hex <<15 // kết quả f
- cout <<oct <<10; // kết quả 12
- cout <<dec <<0x10; // kết quả 16
- return;
- }

Giảng Viên: Nguyễn Văn Thắng 45

---

---

---

---

---

---

---

---

**Cấu trúc của một chương trình c++**

- Một chương trình gồm nhiều đoạn được sắp xếp theo một trình tự logic, mỗi đoạn thực hiện một chức năng trong chương trình gọi là modules
- Trong C++ modules có thể là lớp hoặc hàm
- Mỗi hàm có một tên riêng
- Các lệnh của 1 hàm phải đặt trong { }
- Mỗi lệnh được kết thúc bằng dấu ;

Giảng Viên: Nguyễn Văn Thắng 46

---

---

---

---

---

---

---

---

**Cấu trúc của một chương trình c++**

- Những khai báo, những chỉ thị tiền xử lý.  
#include < Thuvien.h>  
#define ....
- Khai báo các biến toàn cục
- Hàm main ()  
{  
    Các lệnh trong hàm main;  
}

Giảng Viên: Nguyễn Văn Thắng 47

---

---

---

---

---

---

---

---

**Hàm main()**

- Mỗi chương trình trong C++ phải có duy nhất một hàm main(), dùng để điều khiển chương trình
- **Cấu trúc của hàm main**  
<data type> main()  
{  
    **program statement;**  
    **return value;**  
}

Giảng Viên: Nguyễn Văn Thắng 48

---

---

---

---

---

---

---

---

**Ví dụ: chương trình C++ đơn giản**

```
#include <iostream.h>
#include <conio.h>
int main ()
{
    cout << "Hello World!";
    getch();
    return 0;
}
```

Giảng Viên: Nguyễn Văn Thắng 49

---

---

---

---

---

---

---

---

**Biên dịch một chương trình C++**

- Bộ tiền xử lý C++ xem qua mã trong chương trình và thực hiện các chỉ thị được chỉ định bởi các chỉ thị tiền xử lý (ví dụ, #include). Kết quả là một mã chương trình đã sửa đổi mà không còn chứa bất kỳ một chỉ thị tiền xử lý nào cả.
- Sau đó, trình biên dịch C++ dịch các mã của chương trình. Trình biên dịch có thể là một trình biên dịch C++ thật sự phát ra mã assembly

Giảng Viên: Nguyễn Văn Thắng 50

---

---

---

---

---

---

---

---

**Biên dịch một chương trình C++**

- Cuối cùng, trình liên kết hoàn tất mã đối tượng bằng cách liên kết nó với mã đối tượng của bất kỳ các module thư viện mà chương trình đã tham khảo tới. Kết quả cuối cùng là một tập tin thực thi
- Trợ giúp: Dùng tổ hợp phím ctrl+f1

Giảng Viên: Nguyễn Văn Thắng 51

---

---

---

---

---

---

---

---

```
#include <iostream.h>
int main (void)
{ cout << "Hello World\n";}
```

Giảng Viên: Nguyễn Văn Thắng

52

---

---

---

---

---

---

---

---