

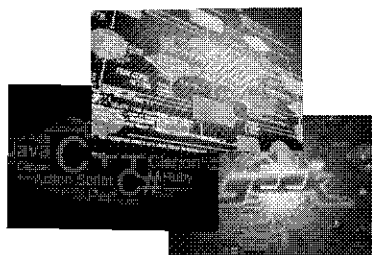


ĐẠI HỌC QUỐC GIA TP HCM  
TRƯỜNG ĐẠI HỌC  
CÔNG NGHỆ THÔNG TIN

# **NHẬP MÔN LẬP TRÌNH**

## **CHƯƠNG III**

### **KIỂU DỮ LIỆU CƠ BẢN**



Nguyễn Trọng Chính  
chinhnt@uit.edu.vn

### **KIỂU DỮ LIỆU CƠ BẢN**

- ❖ CẤU TRÚC CHƯƠNG TRÌNH
- ❖ CHƯƠNG TRÌNH ĐƠN GIẢN
- ❖ CÁC KIỂU DỮ LIỆU CƠ SỞ VÀ PHÉP TOÁN
- ❖ KÝ TỰ VÀ CHUỖI
- ❖ CÁC HÀM THÔNG DỤNG

## CẤU TRÚC CHƯƠNG TRÌNH

### ❖ CÁC THÀNH PHẦN

- \* **Từ khóa (keyword)**: là các từ dành riêng trong một ngôn ngữ lập trình. Các từ khóa không được dùng để đặt tên cho biến, hằng, hàm. Một số từ khóa thông dụng gồm:
  - const, enum, signed, struct, typedef, unsigned...
  - char, double, float, int, long, short, void
  - case, default, else, if, switch
  - do, for, while
  - break, continue, goto, return

## CẤU TRÚC CHƯƠNG TRÌNH

### ❖ CÁC THÀNH PHẦN

- \* **Chỉ thị tiền xử lý (preprocessor directive)**: là các hướng dẫn để trình biên dịch liên kết các file hoặc xác định các thông số cần thiết trước khi biên dịch chương trình. Trong ngôn ngữ C/C++, Các chỉ thị tiền xử lý thường bắt đầu bằng dấu #.  
Chỉ thị tiền xử lý thường gặp là:
  - **#include <file> hoặc #include "file"**: liên kết nội dung file trước khi biên dịch.
  - **#define <abc> <xyz>**: xác định giá trị cho ký hiệu <abc>. Mỗi khi gặp ký hiệu <abc> trong file chương trình, trình biên dịch sẽ thay bằng giá trị <xyz>.

## CẤU TRÚC CHƯƠNG TRÌNH

### ❖ CÁC THÀNH PHẦN

Ví dụ 1 : hai đoạn chương trình sau là giống nhau

```
#define S_HELLO "Xin chào"
```

```
void main() {  
    printf("%s", S_HELLO);  
}
```

Và

```
void main() {  
    printf("%s", "Xin chào");  
}
```

## CẤU TRÚC CHƯƠNG TRÌNH

### ❖ CÁC THÀNH PHẦN

\* Tên/Định danh (Identifier): là một dãy các ký tự theo quy định để chỉ tên biến, hằng số, hằng ký tự, kiểu dữ liệu hay một hàm.

Quy định đối với định danh

- Không trùng với từ khóa, chỉ chứa các ký tự chữ cái, ký số và dấu\_. Đồng thời phải bắt đầu bằng ký tự chữ cái hoặc dấu \_
- Số ký tự tối đa cho một định danh là 255.
- Phân biệt chữ in hoa (upper case) và chữ thường (lower case)

## CẤU TRÚC CHƯƠNG TRÌNH

### ❖ CÁC THÀNH PHẦN

- \* **Dấu chấm phẩy ";"** dùng để phân cách các câu lệnh trong chương trình viết bằng ngôn ngữ C/C++.
- \* **Chú thích (Note)** bắt đầu bằng cặp dấu "//" cho mỗi dòng chú thích hoặc bắt đầu bằng cặp dấu "/\*" cho đầu khối chú thích và kết thúc khối chú thích bằng cặp dấu "\*/".
- \* **Hằng ký tự** là một ký tự được đặt trong cặp dấu "'" và có phân biệt chữ hoa và chữ thường. Ví dụ: 'A', 'a'.
- \* **Hằng chuỗi** là các ký tự được đặt trong cặp dấu "\"" và có phân biệt chữ hoa và chữ thường. Ví dụ: "chao". Chú ý: "A" khác với 'A'.

## CẤU TRÚC CHƯƠNG TRÌNH

### ❖ CÁC THÀNH PHẦN

- \* **Câu lệnh (Statement)**: có thể là một khai báo kiểu dữ liệu, một khai báo biến hoặc hằng, một phép gán, một lời gọi hàm, một cấu trúc lặp, một cấu trúc rẽ nhánh hoặc trả kết quả của hàm với từ khóa return.

## CẤU TRÚC CHƯƠNG TRÌNH

### ❖ CÁC THÀNH PHẦN

\* Ví dụ 2: Tìm chỗ sai trong đoạn chương trình sau.

```
#include <stdio.h>
```

```
void main() {
```

```
    int height, long;
```

```
    printf("nhap chieu dai va chieu rong cua HCN\n");
```

```
    scanf("%d ", &long);
```

```
    scanf("%d ", &height);
```

```
    int p = (height + long) * 2;
```

```
    printf("chu vi hinh chu nhat: %d\n", &p);
```

```
}
```

## CẤU TRÚC CHƯƠNG TRÌNH

### ❖ CÁC THÀNH PHẦN

Các chỗ sai:

```
#include <stdio.h>
```

```
void main() {
```

```
    int height, long;
```

```
    printf("nhap chieu dai va chieu rong cua HCN\n");
```

```
    scanf("%d ", &long);
```

```
    scanf("%d ", &height);
```

```
    int p = (height + long) * 2;
```

```
    printf("chu vi hinh chu nhat: %d\n", p);
```

```
}
```

Tên biến  
trùng từ khóa

Thiếu dấu ;  
cuối một lệnh

## CẤU TRÚC CHƯƠNG TRÌNH

### ❖ CÁC THÀNH PHẦN

\* Ví dụ 3: Tìm chỗ sai trong đoạn chương trình sau.

```
#include <stdio.h>
int main(int n, char** arg) {
    float a, b, c;
    float 1_2cv, d.t;
    printf("Nhập 3 cạnh tam giác\n")
    scanf("%f%f%f", &a, &b, &c);
    1_2Cv = (a + b + c) / 2;
    d.t = sqrt(1_2cv*(1_2cv-a)*(1_2cv-b)*(1_2cv-c));
    printf("Diện tích tam giác %.2f", d.t);
    return 0;
}
```

## CẤU TRÚC CHƯƠNG TRÌNH

### ❖ CÁC THÀNH PHẦN

```
#include <stdio.h>
#include <math.h>
int main(int n, char** arg) {
    float a, b, c;
    float 1_2cv, d.t;
    printf("Nhập 3 cạnh tam giác\n");
    scanf("%f%f%f", &a, &b, &c);
    1_2Cv = (a + b + c) / 2;
    d.t = sqrt(1_2cv*(1_2cv-a)*(1_2cv-b)*(1_2cv-c));
    printf("Diện tích tam giác %.2f", d.t);
    return 0;
}
```

Bắt đầu với ký hiệu khác chữ và \_

Sử dụng ký hiệu khác chữ, số và \_

Phân biệt in hoa và thường

Include file chứa khai báo hàm

## CẤU TRÚC CHƯƠNG TRÌNH

### ❖ TỔ CHỨC CHƯƠNG TRÌNH

Chương trình trong ngôn ngữ C/C++ được tổ chức thành nhiều file, mỗi file có cấu trúc như sau:

Các chỉ thị tiền xử lý

Câu lệnh khai báo kiểu dữ liệu;

Câu lệnh khai báo biến;

Câu lệnh khai báo hàm;

Kiểu TênHàm(ThamSố) {

    Các câu lệnh;

}

Kiểu main(ThamSố) {

    Các câu lệnh;

}

## CẤU TRÚC CHƯƠNG TRÌNH

### ❖ TỔ CHỨC CHƯƠNG TRÌNH

Trong các file chương trình của C/C++, chỉ có một file chứa hàm được đặt tên **main**. Hàm main có thể có hoặc không có tham số cũng như kết quả trả về.

Hàm main sẽ được gọi khi hệ điều hành thực thi chương trình máy tính được biên dịch bởi các file chương trình này.

Các file chương trình của C/C++ được phân chia làm 2 loại. File header (.h) thường chứa các khai báo hàm và kiểu dữ liệu. File source (.cpp hoặc .c) chứa các hàm đã được viết các câu lệnh xử lý (cài đặt).

## CHƯƠNG TRÌNH ĐƠN GIẢN

- ❖CHỈ THỊ LIÊN KẾT FILE
- ❖HÀM MAIN
- ❖KHAI BÁO BIẾN VÀ HẲNG
- ❖CÁC HÀM NHẬP XUẤT ĐƠN GIẢN

## CHƯƠNG TRÌNH ĐƠN GIẢN

### ❖CHỈ THỊ LIÊN KẾT FILE

#### **#include <file>**

- Liên kết các khai báo hàm đã được xây dựng sẵn để sử dụng trong chương trình.
- Các file thông dụng:
  - + stdio.h chứa các khai báo hằng và hàm về nhập và xuất cơ bản;
  - + math.h chứa các khai báo hàm tính toán trong toán học;



## CHƯƠNG TRÌNH ĐƠN GIẢN

### ❖CHỈ THỊ LIÊN KẾT FILE

- + `stdlib.h` chứa các khai báo hàm quản lý bộ nhớ, chuyển đổi giá trị giữa một số kiểu dữ liệu và thực hiện một số lệnh của hệ điều hành.
- + Với C++, có thể sử dụng file `iostream.h` chứa các hàm nhập xuất cơ bản để sử dụng hơn `stdio.h`

## CHƯƠNG TRÌNH ĐƠN GIẢN

### ❖HÀM MAIN

```
#include <stdlib.h>
int main(int n, char **arg) {
    Các câu lệnh;
    return exit_code;
}
```

Trạng thái xử lý `exit_code` có thể là một trong hai hằng số được định nghĩa trong file **`stdlib.h`** là:

**EXIT\_SUCCESS**: quá trình thực hiện không có lỗi;

**EXIT\_FAILURE**: quá trình thực hiện có lỗi.

## CHƯƠNG TRÌNH ĐƠN GIẢN

### ❖HÀM MAIN

#### Ví dụ 4:

```
#include <stdlib.h>
#include <stdio.h>
int main(int n, char** arg) {
    int a = atoi(arg[1]);
    int b = atoi(arg[2]);
    printf("Tong cua %d và %d la ", a, b);
    a = a + b; printf("%d", a);
    return EXIT_SUCCESS;
}
```

## CHƯƠNG TRÌNH ĐƠN GIẢN

### ❖HÀM MAIN

- Hàm main cũng có thể không xử lý tham số

```
int main() {
    Các câu lệnh;
    return exit_code;
}
```

- Hoặc không trả về trạng thái xử lý

```
void main() {
    Các câu lệnh;
}
```

## CHƯƠNG TRÌNH ĐƠN GIẢN

### ❖ KHAI BÁO BIẾN VÀ HẲNG

- Cú pháp để khai báo một biến trong C/C++ như sau:

+ khai báo biến thông thường

**kiểu tênbiến;**

+ khai báo biến giả (bí danh – alias)

**kiểu &tênbiếngiả = tênbiến;**

Biến giả thực chất là biến đã được khai báo trước đó nhưng dùng tên khác để thuận tiện cho người lập trình.

## CHƯƠNG TRÌNH ĐƠN GIẢN

### ❖ KHAI BÁO BIẾN VÀ HẲNG

#### Ví dụ 5

int namsinh;

float diemtrungbinh;

char gioitinh;

double chieudai, chieurong;

## CHƯƠNG TRÌNH ĐƠN GIẢN

### ❖ KHAI BÁO BIẾN VÀ HẲNG

- Cú pháp để khai báo một biến trong C/C++ như sau:

**const kiểu tên\_hằng = giá trị;**

Hằng là một dạng biến nhưng không thể thay đổi giá trị của nó trong quá trình thực hiện chương trình.

#### Ví dụ 6

```
const int MAX_AGE = 100;
```

```
const float PI = 3.14;
```

## CHƯƠNG TRÌNH ĐƠN GIẢN

### ❖ CÁC HÀM NHẬP XUẤT ĐƠN GIẢN

- Các hàm nhập dữ liệu từ bàn phím được khai báo trong file `stdio.h`, gồm:

+ **`scanf("chuỗi định dạng", &biến1, &biến2,...);`**

Hàm này dùng để nhập giá trị số hoặc ký tự. Chuỗi định dạng gồm các ký hiệu định dạng liên tiếp nhau, mỗi ký hiệu bắt đầu bằng dấu % và tương ứng với 1 biến theo thứ tự xuất hiện. Một số ký hiệu định dạng thường dùng:

## CHƯƠNG TRÌNH ĐƠN GIẢN

### ❖ CÁC HÀM NHẬP XUẤT ĐƠN GIẢN

Ký hiệu	Ý nghĩa
c	nhập một ký tự
d	nhập số nguyên có dấu
f	nhập số thực
o	nhập số bát phân
s	nhập chuỗi không chứa khoảng trắng
u	nhập số nguyên không dấu
x, X	nhập số thập lục phân

## CHƯƠNG TRÌNH ĐƠN GIẢN

### ❖ CÁC HÀM NHẬP XUẤT ĐƠN GIẢN

**Ví dụ 7** nhập tên và năm sinh của một người

```
int namsinh;  
char ten[10];  
scanf("%s%d", &ten[0], &namsinh);
```

Đoạn chương trình tương đương

```
int namsinh;  
char ten[10];  
scanf("%s%d", ten, &namsinh);
```

## CHƯƠNG TRÌNH ĐƠN GIẢN

### ❖ CÁC HÀM NHẬP XUẤT ĐƠN GIẢN

+ **printf("chuỗi định dạng", giá trị1, giá trị2, ...);**

Hàm này dùng để in ra màn hình giá trị số hoặc ký tự. Chuỗi định dạng có thể chứa thông báo cùng với các ký hiệu định dạng liên tiếp nhau, mỗi ký hiệu định dạng bắt đầu bằng dấu % và tương ứng với một giá trị là một biến hoặc một hằng theo thứ tự xuất hiện. Một số ký hiệu định dạng thường dùng:

## CHƯƠNG TRÌNH ĐƠN GIẢN

Ký hiệu	Ý nghĩa
Nc	in một ký tự với độ rộng N cột
Nd	in số nguyên có dấu với độ rộng N cột
N.Mf	in số thực với độ rộng N cột trong đó có M chữ số thập phân
No	in số bát phân với độ rộng N cột
s	in chuỗi ký tự
Nu	in số nguyên không dấu với độ rộng N cột
Nx, NX	in số thập lục phân với độ rộng N cột

## CHƯƠNG TRÌNH ĐƠN GIẢN

### ❖ CÁC HÀM NHẬP XUẤT ĐƠN GIẢN

Ví dụ 8 in ra màn hình tên và năm sinh của một người đã nhập trong ví dụ 7

```
printf("ong/ba %s sinh nam %d", ten, namsinh);
```

Ví dụ 9 nhập bán kính hình tròn và in ra diện tích với 2 chữ số phần thập phân.

```
float r, s;
```

```
scanf("%f", &r); s = r * r * 3.14;
```

```
printf("Dien tich hinh tron la: %.2f", s);
```

## CHƯƠNG TRÌNH ĐƠN GIẢN

### ❖ CÁC HÀM NHẬP XUẤT ĐƠN GIẢN

Với C++, sử dụng các toán tử nhập xuất được khai báo trong file `iostream.h` như sau:

+ **cin >> biến1 >> biến2 >> .. >> biếnN;**

toán tử >> gọi là toán tử nhập sẽ gán các giá trị tương ứng từ luồng dữ liệu nhập (in-stream) cin cho các biến.

Ví dụ 10 nhập tên và năm sinh như ví dụ 7

```
cin >> ten >> namsinh;
```

## CHƯƠNG TRÌNH ĐƠN GIẢN

### ❖ CÁC HÀM NHẬP XUẤT ĐƠN GIẢN

+ **cout << giá trị1 << giá trị2 << ... << giá trịN;**

toán tử << gọi là toán tử xuất sẽ ghi các giá trị là biến hoặc hằng tương ứng vào luồng dữ liệu xuất (out-stream) cout.

**Ví dụ 11** in ra màn hình tên và năm sinh của một người như trong ví dụ 8

```
cout << "ong/ba " << ten << " sinh nam " << namsinh;
```

## CHƯƠNG TRÌNH ĐƠN GIẢN

### ❖ CÁC HÀM NHẬP XUẤT ĐƠN GIẢN

Để định dạng kết quả xuất cho cout, sử dụng các hàm định dạng như sau:

- + **cout.setprecision(n):** xác định số chữ số phân thập phân của số thực sẽ được in, có tác dụng đến lần gọi kế tiếp.
- + **cout.setf(n):** chọn kiểu in số thực, n có thể là hằng **ios::fixed** xác định số thực có phân thập phân có kích thước cố định hoặc **ios::scientific** xác định số thực được viết dưới dạng lũy thừa với cơ số 10, có tác dụng đến lần gọi kế tiếp.



## CHƯƠNG TRÌNH ĐƠN GIẢN

### ❖ CÁC HÀM NHẬP XUẤT ĐƠN GIẢN

- + **cout.unsetf(n)**: bỏ chọn kiểu in số thực. n là kiểu như trong hàm setf.
- + **cout.width(n)**: xác định độ rộng cột cho một kết quả in ngay sau hàm được gọi.

#### Một số hằng số định dạng:

- + **oct**: các số được in dưới dạng bát phân
- + **hex**: các số được in dưới dạng thập lục phân
- + **dec**: các số được in dưới dạng thập phân
- + **endl**: xuống dòng.

## CHƯƠNG TRÌNH ĐƠN GIẢN

### ❖ CÁC HÀM NHẬP XUẤT ĐƠN GIẢN

Ví dụ 12 nhập bán kính hình tròn và in ra diện tích với 2 chữ số phân thập phân.

```
float r, s;  
cin >> r;  
s = r * r * 3.14;  
cout.setprecision(2);  
cout << "Diện tích hình tròn là: " << s;
```

## CHƯƠNG TRÌNH ĐƠN GIẢN

### ❖ BÀI TẬP

**Bài 1:** nhập vào tên, số báo danh, điểm các môn toán, lý, anh văn của 3 học sinh và in bảng điểm của 3 thí sinh.

**Bài 2:** nhập vào chiều dài 3 cạnh của 1 tam giác, in ra chu vi và diện tích của tam giác.

**Bài 3:** nhập vào một số ở hệ thập phân và in giá trị của nó trong hệ bát phân và thập lục phân.

**Bài 4:** nhập vào một số nguyên dương n, tính

$$s1 = 1 + 2 + \dots + n;$$

$$s2 = 1^2 + 2^2 + \dots + n^2;$$