

Tin Học Cơ Sở 4



Vũ Thị Hồng Nhạn

(vthnhan@vnu.edu.vn)

Human Machine Interaction Lab.

Vietnam National Univ., Hanoi

Môn học

- ❖ Tên: Tin học cơ sở 4 (3 học trình)
- ❖ Mã: INT1006 3
- ❖ Thời gian: 8~12am, Thứ 4, 3-G3, học 7 tuần từ 15/02 ~ 30/03
- ❖ Giảng viên: TS. Vũ Thị Hồng Nhạn
- ❖ Tham khảo
 - Phạm Hồng Thái. Bài giảng Ngôn ngữ lập trình C/C++, Hà Nội, 2003.
 - B.W.Kernighan & D.M.Ritchie “The C programming language” . 2nd edition, Prentice Hall, 1988

Môn học

❖ Website môn học

- www.bbc.vnu.edu.vn

❖ Đánh giá hết môn

- 10% Dự lớp học & Kiểm giữa kỳ
- 30% Thực hành tại Lab.
- 60 % thi cuối kỳ

Mục tiêu của môn học

- ❖ Học các kỹ năng cơ bản để giải quyết bài toán
- ❖ Học cách thiết kế một chương trình
- ❖ Học và sử dụng ngôn ngữ lập trình C



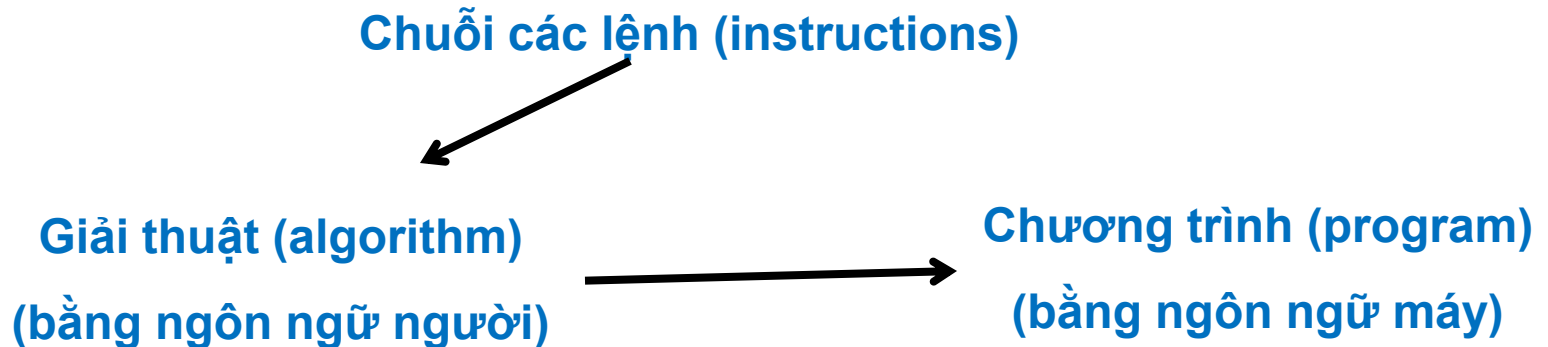
Schedule

Week	Lecture	Lab
1	Giới thiệu	
2	Các cấu trúc điều khiển (control flow)	start
3	Hàm và cấu trúc chương trình (Functions & program structure)	
4	Mảng và chuỗi (arrays & string)	
5	Midterm & Luyện tập	
6	Con trỏ (pointers)	
7	Các kiểu dữ liệu tự tạo (structures)	
8	Tập tin (files)	
9	Ôn tập hết kỳ	
10	Final	
11		
12		end

Ngôn ngữ lập trình & Phương pháp lập trình

1. **Ngôn ngữ lập trình (programming language)**
 - Giải thuật (algorithm)
 - Chương trình (program)
 - Programming language
2. **Các bước lập trình**
3. **Kỹ thuật lập trình**
 - Input-Process-Output cycle
 - Sử dụng flowchart
4. **Phát triển phần mềm**
5. **Môi trường phát triển tích hợp (IDE)**

1. Ngôn ngữ (lập trình) là gì?



- ❖ Một chương trình cần được viết bằng một **ngôn ngữ**
- ❖ Có rất nhiều ngôn ngữ lập trình
 - Bậc thấp (Low-level): phụ thuộc vào máy tính → máy có thể hiểu
 - Bậc cao (High-level): độc lập với máy → cần được dịch (translator)
- ❖ C là một **ngôn ngữ lập trình bậc cao**

1. Ngôn ngữ lập trình là gì?

Ví dụ

Machine binary language

```
00001001001011100110011001101001011011000110010100001001001000
0101011100100110010100110001001011100110001100100010000010100:
0110001101101111011011010111000001101001011011000110010101100:
00110110010101100011011101000110100101101111011011100000100100
01110100001000100000101000001001001011100110000101101100011010
1010000010010010111001100111011011000110111101100010011000010:
0110111000001010000010010010111001110100011110010111000001100:
1001011011100010110000100011011001100111010101101110011000110:
00001001001011100111000001110010011011110110001100001001001100
1001011011100011101000001010000010010010000100100011010100000:
01010101010001010010001100100000001100000000101000001001011100
01010111001101110000001011000010110100110001001100100011100000
0000101000001001001000010010001101010000010100100100111101001:
00110010000000110001000010100000100101101101011011110111011000
0011000000001010000010010111001101110100001000000010010101101:
0110011100000010110100110010001100000101110100001010000010010:
0010110000100101011011110011000000001010000010010111001101110:
1100010110110010010101100110011100000010110100110010001101000:
00100000010110110010010101100110011100000010110100110010001100
0000000010100000100101101100011001000010000001011011001001010:
01011101001011000010010101101111001100010000101000001001011000
1111001100000010110000100101011011110011000100101100001001010:
011101000010000000100101011100110000001011000101101100100:
10000101110100001010000010010110110101101111011101100010000000
00001010000010010110001000100000001011100100110001001100001100
00000000101000101110010011000100110000110001001110100000101000
0000100101110010011001010111001101110100011011110111001001100:
0110011001010011000100111010000010100000100100101110011100110:
0110110101100001011010010110111000101100001011100100110001000110001100110010100110001001011010110
1101011000010110100101101110000010100000100100101110011010010110010001100101011011100111010000001001
0010001001000111010000110100001100111010001000000010100001000111010011100101010100101001001000000011
0010001011100011100000101110001100010010001000001010
```

Low-level assembly

```
main:
    !#PROLOGUE# 0
    save %sp,-128,%sp

    !#PROLOGUE# 1
    mov 1,%o0
    st %o0,[%fp-20]
    mov 2,%o0
    st %o0,[%fp-24]
    ld [%fp-20],%o0
    ld [%fp-24],%o1
    add %o0,%o1,%o0
    st %o0,[%fp-28]
    mov 0,%i0
    nop
```

High-level

```
int main()
{
    int x, y, z;

    x = 1;
    y = 2;
    z = x+y;

    return 0;
}
```

1. Ngôn ngữ lập trình là gì?

Dịch như thế nào?

Một chương trình được viết bằng ngôn ngữ bậc cao

(vd. C program)

↓
COMPILER
(ví dụ, Visual C++, gcc)

Chương trình dạng bậc thấp (ngôn ngữ máy) mà
máy tính có thể hiểu

1. Ngôn ngữ lập trình là gì?

Giải thuật (algorithm)

- ❖ Là một **chuỗi các thao tác** xác định trên một đối tượng, sao cho sau khi thực hiện một số **hữu hạn các bước** thì đạt được **mục tiêu** nào đó
- ❖ Bản chất của giải thuật: **giải thuật = logic + điều khiển** [R.A.Kowalski]
 - Logic: “thuật toán làm gì & giải quyết vấn đề gì?”
 - Điều khiển: “thuật toán làm thế nào?”

1. Ngôn ngữ lập trình là gì?

Chương trình (program)

- ❖ Là một **tập các mô tả, các phát biểu**, nằm trong một hệ thống quy ước về **ý nghĩa** và **thứ tự thực hiện**, nhằm điều khiển máy tính làm việc
 - **Chương trình = thuật toán + cấu trúc dữ liệu** [Niklaus Wirth]
- ❖ Dựa **trên ba cấu trúc điều khiển**
 - **Tuần tự**: các bước thực hiện tuần tự một cách chính xác từ trên xuống, mỗi bước thực hiện đúng 1 lần
 - **Chọn lọc**: chọn 1 trong nhiều thao tác để thực hiện
 - **Lặp lại**: một hay nhiều bước được thực hiện lại lại một số lần

1. Ngôn ngữ lập trình là gì?

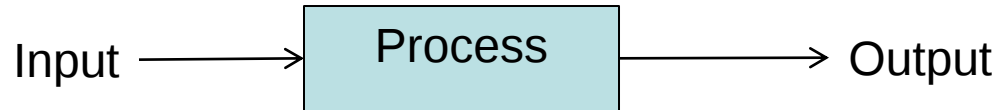
Programming language

- ❖ Là một hệ thống **các ký hiệu** tuân theo các quy ước về **ngữ pháp** và **ngữ nghĩa**, dùng để xây dựng chương trình cho máy tính
- ❖ Một chương trình được viết bằng ngôn ngữ lập trình cụ thể (vd. C/C++, Java) gọi là **source program**

2. Các bước lập trình

1. Phân tích vấn đề và xác định các đặc điểm (I-P-O)
2. Đưa ra giải thuật
3. Viết chương trình (implementation)
4. Chạy thử chương trình (dịch chương trình)
5. Kiểm chứng và hoàn thiện

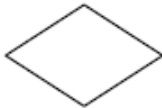
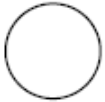
I-P-O cycle



- ❖ Quy trình xử lý cơ bản của máy tính: **nhập – xử lý – xuất**
- ❖ Ví dụ: xác định Input, Process, Output của chương trình tính “tiền lương công” nhận tháng 10/2016, biết $lương = lương\ căn\ bản * ngày\ công$
 - Input: *lương căn bản, ngày công*
 - Process: *nhân lương căn bản với ngày công*
 - Output: *lương*

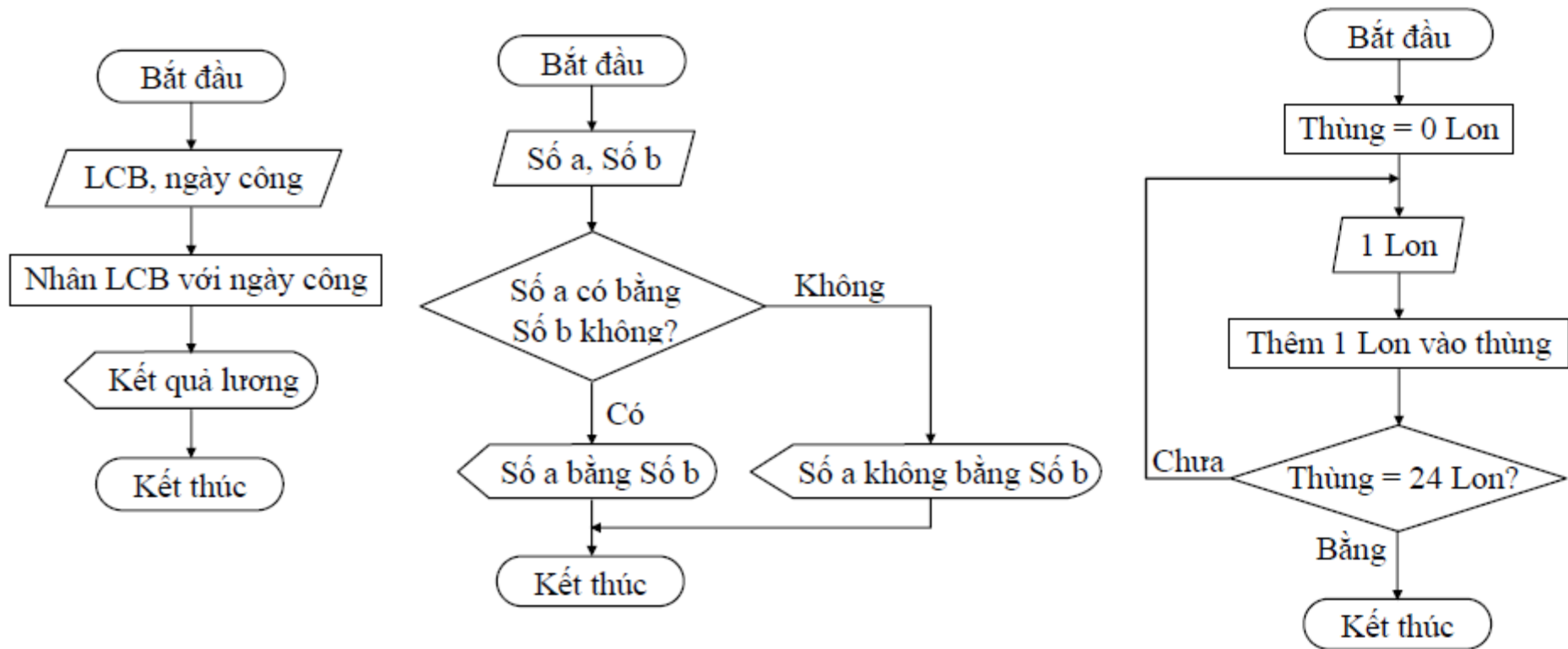
3. Kỹ thuật lập trình

Flowchart

Hình dạng (symbol)	Hành động (Activity)
	Dữ liệu vào (Input)
	Xử lý (Process)
	Dữ liệu ra (Output)
	Quyết định (Decision), sử dụng điều kiện
	Luồng xử lý (Flow lines)
	Gọi CT con, hàm... (Procedure, Function...)
	Bắt đầu, kết thúc (Begin, End)
	Điểm ghép nối (Connector)

3. Kỹ thuật lập trình

Flowchart



❖ Tính lương

❖ So sánh hai số

❖ Xếp lon vào thùng

4. Phát triển phần mềm

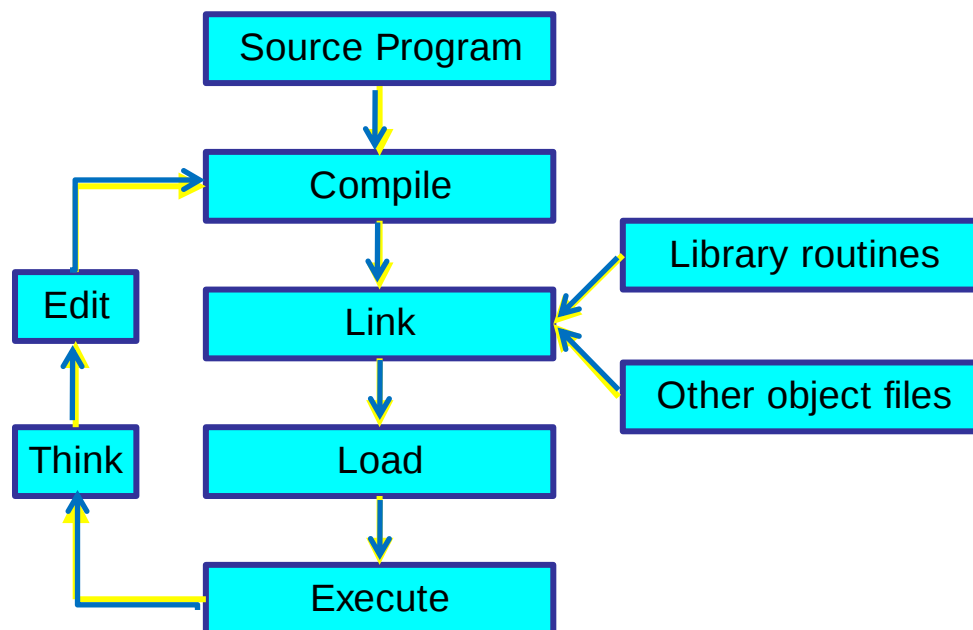
❖ Hoạt động chính

- Soạn thảo (viết chương trình)
- Dịch (compiling tạo ra .obj file)
- Liên kết các files đã dịch (tạo .exe file)

- ❖ Object files

- ❖ Library modules

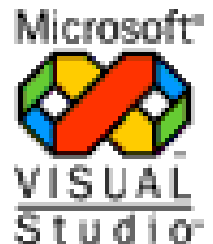
- Nạp và xử lý
- Chạy kiểm tra & sửa chữa



5. Môi trường tích hợp (IDE)

❖ Kết hợp tất cả các khả năng mà một lập trình viên muốn khi phát triển một phần mềm (vd. VC++, Dev-C++)

- Editor
- Compiler
- Linker
- Loader
- Debugger
- Viewer



6. Môi trường thực hành (environment) tại Lab

❖ Compiler

- GCC (GNU compiler collections) in Linux/Cygwin



❖ Editor: Dev-C++

- Môi trường phát triển tích hợp (IDE) hỗ trợ việc lập trình C/C++
- Hỗ trợ trình biên dịch nền tảng GCC
- <http://www.bloodshed.net/devcpp.html> (version 5)

7. Ví dụ chương trình đầu tiên

```
// a simple program  
#include <stdio.h>
```

Chú thích, bình luận

```
int main() {  
    printf("Hello world!");  
    return 0;  
}
```

Hàm có tên
main() chỉ ra chỗ
bắt đầu một
chương trình

Kết thúc xử lý của main() và
kết thúc chương trình

Lệnh in ra màn hình dòng chữ
trong ngoặc