

Bài giảng môn NHẬP MÔN LẬP TRÌNH

(45 tiết Lý thuyết + 30 tiết Thực hành)

■ GV: Thái Hùng Văn

Giới thiệu tổng quan về **Lập Trình**

1. Các khái niệm cơ bản

* Lập trình (Programming)

Là Lập ra / tạo ra / làm ra chương trình máy tính

(ngày trước còn gọi là Thảo chương – cũng là Thảo ra /soạn ra /làm ra chương trình)

→ Người chuyên lập trình gọi là Lập trình viên /Thảo chương viên (Programmer).

* Máy Tính (Computer)

Là các loại máy có hệ thống xử lý, biết tính toán & xử lý thông minh
(*máy tính xách tay /đề bàn, máy tính bảng, điện thoại di động, TV, máy ATM, đầu đĩa DVD, máy ảnh số, ...*)

1. Các khái niệm cơ bản

* Chương trình Máy tính

- CTMT (Computer Program, thường gọi tắt là CT) là một chuỗi các chỉ thị (lệnh - instruction), được viết ra để thực hiện một nhiệm vụ nào đó bằng MT.
- Nhờ có CT, MT mới xử lý được các công việc. Nói cách khác: MT chỉ hoạt động được khi có CT, và hoạt động của MT chính là thực thi (chạy - execute) CT.

2. Tầm quan trọng của Lập Trình

* MT mà không có CT thì sao?

Thì “đơ” hoàn toàn, không xử lý được gì cả!

→ MT mà không có CT thì giống như cái xác không hồn!

* CT mà không có MT thì sao?

Thì cũng không thể hoạt động được, dù chỉ là một công việc nhỏ.

=> CT bị đưa ra khỏi MT thì cũng giống như hồn lìa khỏi xác!

2. Tầm quan trọng của Lập Trình

* MT (và CT) có quan trọng không?

Cực kỳ quan trọng, là thứ không thể thiếu trong cuộc sống.

(Chỉ cần không có điện thoại, mạng Internet là đã khó sống:)

* MT (&CT) có thể làm được những gì?

Nhờ sự hỗ trợ của MT mà con người làm được đủ thứ.

(nhiều thứ ko làm vì ko cần hoặc ko nên chứ ko phải là ko thể:)

** Trong vòng 10 năm tới có thể ~50% thanh niên Mỹ sẽ bị Robot & AI cướp mất việc. Các nơi khác thì sẽ còn thê thảm hơn!!!*

(<http://thanhnien.vn/gioi-tre/gioi-tre-my-canh-tranh-viec-lam-voi-robot-836270.html>)

3. Học Lập Trình

* HSSV có thể /có nên học LT không?

Hoàn toàn **Có thể** và **Rất nên**!

(Nhiều nước phát triển đã bắt phải học từ đầu cấp 2, thậm chí từ 5 tuổi – vì thấy phù hợp và rất cần thiết cho tương lai, để có thể “sống sót” trong thời đại Cách mạng Công nghiệp 4.0)

* Học LT có vất vả & mất nhiều thời gian?

Có những Ngôn ngữ LT được thiết kế để dạy cho các HS nhỏ tuổi (5 tuổi mà còn học được), chúng không chỉ dễ nắm bắt trong thời gian ngắn mà còn cảm thấy rất vui vẻ và hứng thú !

3. Học Lập Trình

* Học LT được gì?

- Các HV khá thì có thể làm được đủ thứ chương trình & thực hiện được rất nhiều việc, làm được nhiều điều có ích trên thực tế.
- Có khả năng tư duy và suy nghĩ độc lập hơn, tốt hơn. Có khả năng tìm ra giải pháp tốt cho các vấn đề có nhiều cách giải quyết (*là những vấn đề hay gặp trên thực tế*)
- Không chỉ tăng khả năng tư duy, sáng tạo; HV còn có khả năng nắm bắt, làm chủ các công nghệ mới – một điều rất cần thiết trong vài thập niên tới.
- Có thể tìm được các công việc tốt ở nhiều công ty lớn với mức đãi ngộ cao cùng điều kiện phát triển nghề nghiệp tốt.

- ...

3. Học Lập Trình

* LT giỏi được gì?

Những nghề liên quan đến khoa học máy tính chiếm hầu hết các thứ hạng cao trong bảng xếp hạng các nghề nghiệp tốt nhất của Mỹ, 7 trên 10 hạng đầu là kỹ sư phần mềm và lập trình viên.

(<http://aptech.vn/tin-tuc-su-kien/lap-trinh-vien-full-stack-vuot-qua-chuyen-gia-nghien-cuu-khoa-hoc-du-lieu-de-tro-thanh-nghe-nghiep-tot-nhat-gioi.html>)

4. Máy Tính

* Khái niệm về Máy Tính & Computer

- Máy tính không phải là máy dùng để tính toán, đó là các thiết bị biết "xử lý", có "thông minh".
- “Computer”: Người tính toán →
 - Máy giải toán ("máy tính")
 - Thiết bị xử lý.

4. Máy Tính

* Khái niệm về Máy Tính & Computer

- Computer là các thiết bị điện tử có thể giải quyết các vấn đề khác nhau, có thể xử lý dữ liệu, lưu trữ và truy xuất dữ liệu, thực hiện tính toán nhanh chóng hiệu quả hơn con người.
- Ví dụ:
 - PC, Smartphone, Tablet, TV, ATM,...
 - hơi đặc biệt: SIM card, Flash card, USB disk,...

4. Máy Tính

* Các loại máy tính (theo góc độ khả năng xử lý):

- Siêu máy tính (supercomputers)
- Siêu máy tính cỡ nhỏ (minisupercomputers)
- Máy tính lớn (mainframe) / (Máy tính mini)
- Máy chủ (server)
- Máy trạm (workstation)
- Máy tính cá nhân (PC – personal computer)
- Máy chuyên biệt.

5. Chương Trình

- CT Nguồn (Source Code Program): do LTV viết ra bằng một Ngôn ngữ Lập trình nào đó, các chỉ thị trong CT ở dạng văn bản gọi nhớ gọi nghĩa.
 - CT Mã máy (machine code program): là loại CT duy nhất mà máy hiểu và thực thi được, mỗi chỉ thị là một dãy bit ứng với một lệnh thuộc tập lệnh của CPU, cấu trúc tổ chức của CT phải tương hợp với OS (Hệ điều hành).
- ⇒ Mỗi CT mã máy chỉ chạy được trên một họ CPU và OS cụ thể, và LTV hầu như không thể viết ra nổi!
- ⇒ LTV thường sẽ viết CT nguồn bằng NNLT (cấp càng cao càng dễ) và dùng trình dịch để chuyển CT nguồn sang CT mã máy.

Chương trình dịch & Phần mềm

- Chương trình dịch (source code => executable /binary /machine code)
 - . Trình hợp dịch (assembler)
 - . Trình thông dịch (interpreter)
 - . Trình biên dịch (compiler)
- Phần mềm (Software)
 - . Phần mềm hệ thống
 - . Phần mềm ứng dụng

6. Phần mềm - Phần cứng (Software-Hardware)

- Nghĩa rộng
- Nghĩa hẹp
- Phần “dẻo” (firmware)

LẬP TRÌNH

- Lập Trình (Programming) là làm ra chương trình.
- Lập Trình Viên (Programmer) là người làm ra chương trình.
- Để tạo ra một Phần Mềm (Software), lập trình viên dùng 1 ngôn ngữ lập trình và 1 công cụ lập trình tương ứng để viết ra 1 chương trình nguồn, rồi dịch chương trình mã nguồn đó sang mã thực thi.
- Cuộc Cách mạng Công nghiệp 4.0
- Mối quan hệ giữa CMCN 4.0 và Lập Trình

Ngôn ngữ Lập trình

- Khái niệm về Ngôn Ngữ Lập Trình (Programming Language)
- Ngôn ngữ lập trình cấp thấp (low level programming language)
- Ngôn ngữ lập trình cấp cao (high level programming language)
- Một số ngôn ngữ lập trình hay gặp: Assembly, C, C++, C#, Java, Pascal, Basic, PHP, Scratch, ...

Cách tạo ra CT giải quyết vấn đề

* CÁCH GIẢI QUYẾT MỘT BÀI TOÁN THỰC TẾ:

- Xác định yêu cầu của bài toán.
- Nghĩ ra một phương cách giải quyết (algorithm).
- Thực hiện phương cách đó.
- Kiểm tra kết quả xem có đúng không.

Cách tạo ra CT giải quyết vấn đề

* CÁCH TẠO CT GIẢI QUYẾT VẤN ĐỀ CŨNG TƯƠNG TỰ:

- Xác định yêu cầu của bài toán /mục đích của CT.
- Nghĩ ra cách giải quyết /thuật toán (*và viết lại dưới dạng mã giả /lưu đồ*)
- Tạo CT nguồn từ thuật toán (*cài đặt thành mã thật của NNLT*)
- Dịch + chạy CT và kiểm tra kết quả (*nếu lỗi thì tiến hành dò lỗi (debug) + sửa lỗi (fixbug) và tiếp tục kiểm tra đến khi CT chạy đúng*)

Môi trường Lập trình

- Qui trình tạo phần mềm hiện nay được hỗ trợ rất hiệu quả bởi các IDE (Integrated Development Environment)
- Các IDE tích hợp hầu hết các chức năng quan trọng: tạo giao diện, soạn thảo chương trình nguồn, quản lý các tập tin mã nguồn & các phiên bản, kiểm tra lỗi cú pháp, dịch sang mã thực thi, chạy thử & cho phép dò bắt lỗi (debug),...
- Một số IDE thông dụng: Visual Studio (C++, C#, VB,...), Eclipse, Visual Café (Java), ...

Một số chương trình ví dụ

- Chương trình tính chu vi đường tròn khi biết chiều dài đường kính:

// chương trình được viết bằng ngôn ngữ lập trình C

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

```
void main (void)
```

```
{
```

```
float duong_kinh, chu_vi;
```

```
printf ( " Nhập chiều dài của đường kính: " );
```

```
scanf ( " %f ", &duong_kinh );
```

```
chu_vi = 3.14 * duong_kinh;
```

```
printf ( " Chu vi của đường tròn tương ứng là: %f \n ", chu_vi );
```

```
}
```

Một số chương trình ví dụ

- Chương trình Giải phương trình bậc 1 ($aX+b=0$):

Thuật toán:

- . Nhập hai hệ số a, b
- . Nếu $a=0$:

Nếu $b=0$: thông báo Ph/Tr có vô số nghiệm – nghiệm là mọi giá trị thực

Ngược lại: thông báo Ph/Tr vô nghiệm.

Ngược lại (tức a khác 0):

$$X = -b/a$$

Thông báo phương trình có 1 nghiệm là X

Một số chương trình ví dụ

- Mã nguồn cài đặt bằng ngôn ngữ C++ của thuật toán Giải phương trình bậc 1:

```
#include <iostream>
using namespace std;
int main() {
    float a, b, x;
    cout << "Nhap gia tri A va B: " ;
    cin >> a >> b;
    if ( a == 0) {
        if ( b == 0 ) cout << "PT co vo so nghiem";
        else cout << "PT vo nghiem " ;
    }else {
        x = -b / a ;
        cout << "PT co 1 nghiem la: " << x;
    }
    return 0;
}
```

Môi trường thông dụng của lập trình viên

- Hệ thống máy tính: PC (laptop, desktop), Smartphone, Tablet, ..
- Hệ điều hành: Windows, Android, iOS, Linux, ..
- Ngôn ngữ lập trình: C, C++, C#, Objective-C, Java, PHP, VB, ..

Lập trình với Scratch

- **Giới thiệu** - <https://scratch.mit.edu/>
(có thể download bản offline ở mục <Support /OfflineEditor> để dùng khi ko có Internet)
- **Tạo account nếu muốn share /public**
- **Khuyến khích học / làm việc nhóm (mỗi nhóm khoảng 4 SV)**
- **Tích cực tiến hành khảo sát /thí nghiệm /làm nhiều bài tập.**