



IT001 - NHẬP MÔN LẬP TRÌNH

CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN VỀ MÁY TÍNH VÀ LẬP TRÌNH

Chương này sẽ giới thiệu một số khái niệm cơ bản về máy tính, chương trình máy tính và ngôn ngữ lập trình, ... giúp người học hiểu rõ hơn về máy tính và có thể phát triển chương trình một cách hiệu quả.

Khoa Khoa học Máy tính



Nội dung

- 1.1 Tổng quan về máy tính
- 1.2 Các khái niệm cơ bản về lập trình
- 1.3 Giới thiệu sơ lược về ngôn ngữ lập trình
- 1.4 Giới thiệu ngôn ngữ lập trình C++
- Tài liệu tham khảo



1.1 Tổng quan về máy tính



1.1 Tổng quan về máy tính

1.1.1 Máy tính là gì?

1.1.2 Xử lý thông tin trên máy tính

1.1.3 Phần cứng máy tính

1.1.4 Phần mềm máy tính

1.1.5 Biểu diễn và đo lường thông tin



1.1.1 Máy tính là gì ?

- Máy tính (computer) là một thiết bị điện tử có khả năng thực hiện các lệnh. Các lệnh cơ bản mà máy tính thực hiện là nhập (lấy dữ liệu), xuất (hiển thị kết quả), lưu trữ và thực hiện các phép toán số học và logic.



laptop



*Siêu máy chủ
AI NVIDIA DGX A100*



destop

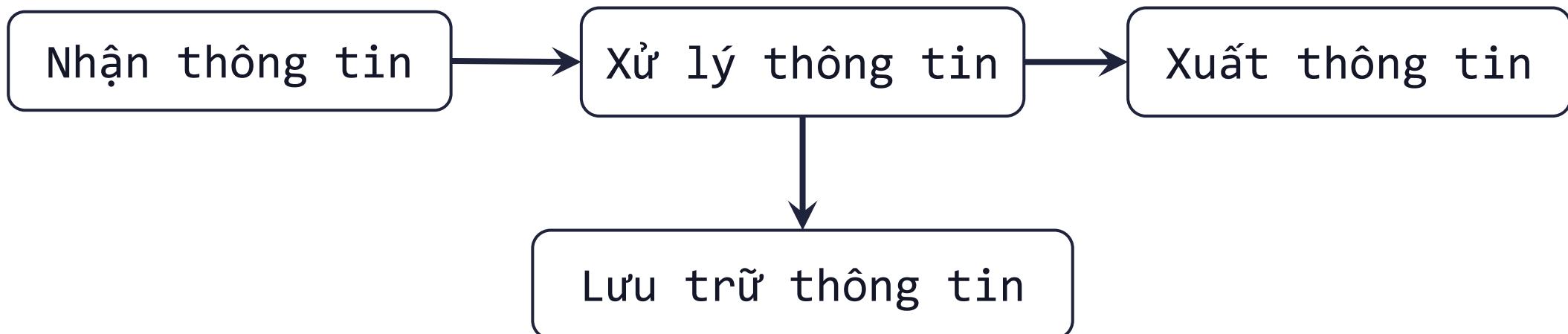


Siêu máy tính Frontier (Mỹ)



1.1.2 Xử lý thông tin trên máy tính

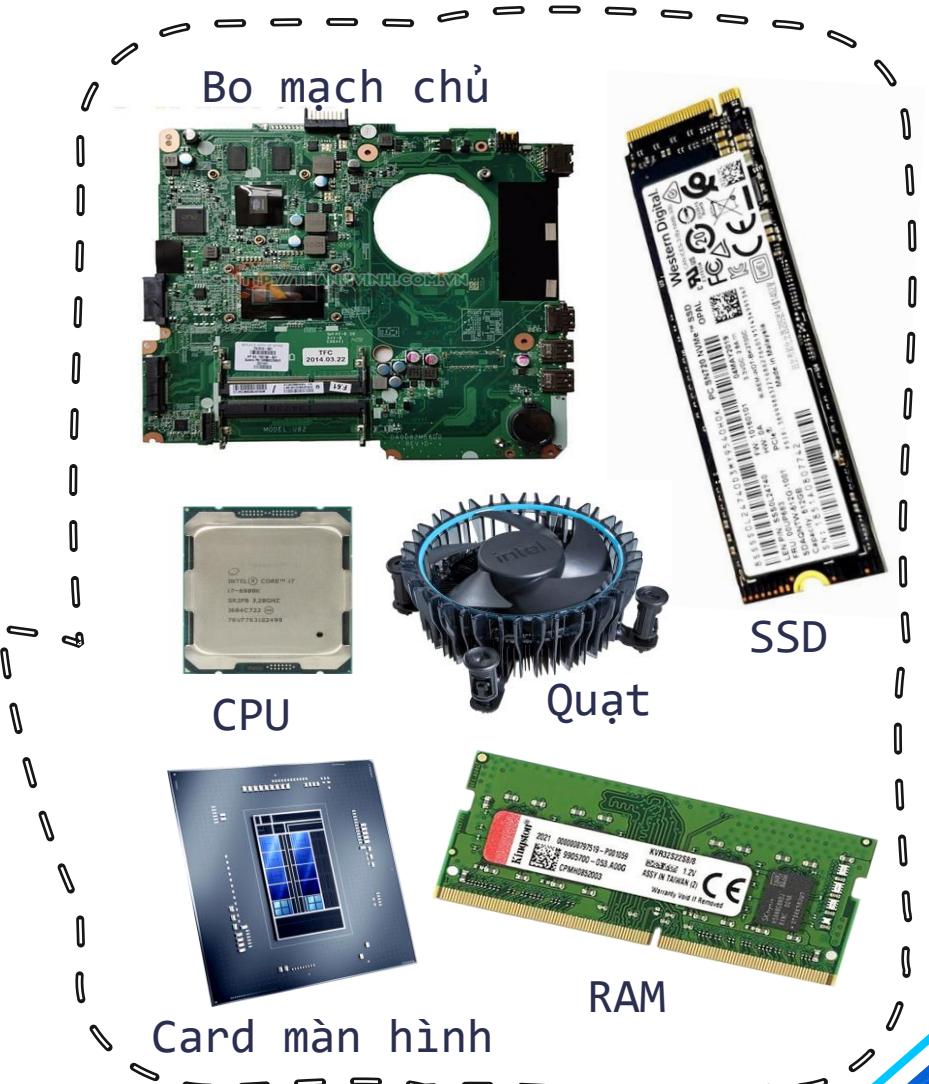
- Thông tin là sự hiểu biết của con người về một thực thể nào đó, có thể thu thập, lưu trữ, xử lý được.
- Dữ liệu: Dữ liệu là thông tin dưới dạng ký hiệu, chữ viết, chữ số, hình ảnh, âm thanh hoặc dạng tương tự → dữ liệu là thông tin được mã hóa trong máy tính.
- Quá trình xử lý thông tin trên máy tính minh họa qua sơ đồ sau:





1.1.3 Phân cứng máy tính

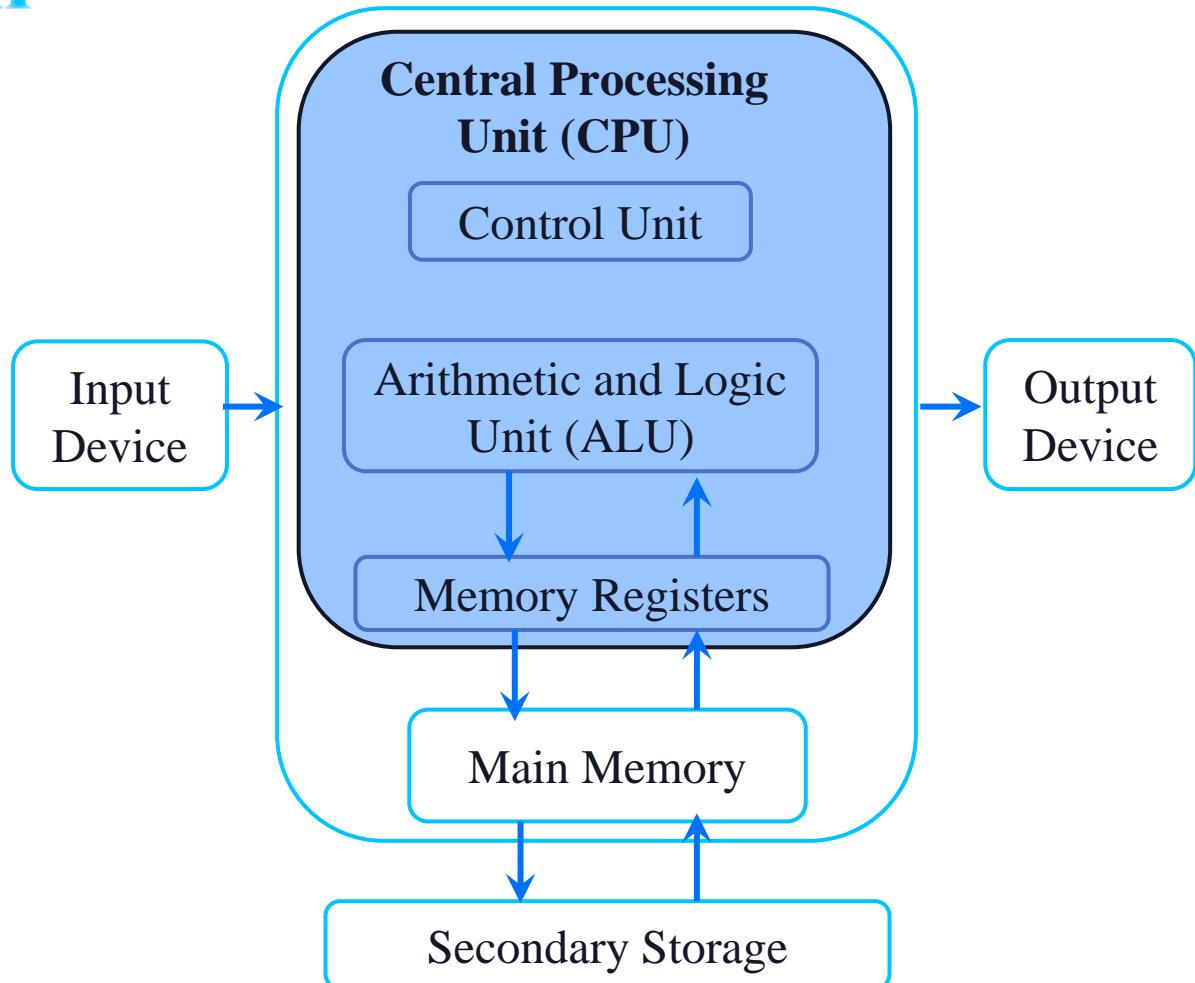
- Ví dụ một số phần cứng máy tính:





1.1.3 Phần cứng máy tính

- Các thành phần phần cứng (Hardware) bao gồm:
 - **Bộ xử lý trung tâm** (Central Processing Unit: CPU)
 - **Bộ nhớ chính** (Main Memory: MM)
 - **Thiết bị đầu vào/đầu ra** (Input / Output Devices)
 - **Thiết bị lưu trữ thứ cấp** (Secondary Storage)



Sơ đồ Cấu trúc máy tính:
Các thành phần phần cứng của máy tính và bộ nhớ chính



Bộ xử lý trung tâm - Central Processing Unit (CPU)

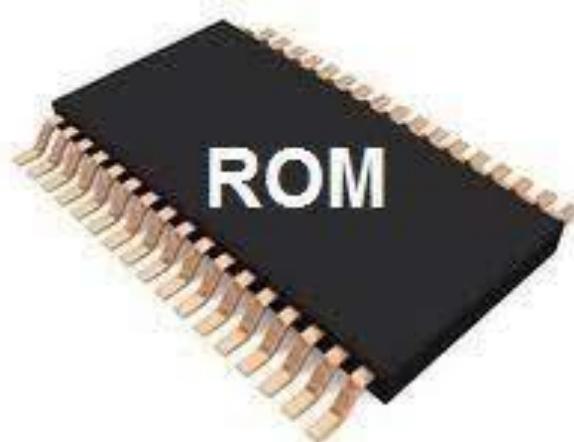
- Bộ xử lý trung tâm - Central Processing Unit (CPU) - là “bộ não” của máy tính.
- CPU càng mạnh thì máy tính càng nhanh.
- Bộ số học luận lý (Arithmetic and Logic Unit): Tính toán các phép số học và luận lý.





Main memory

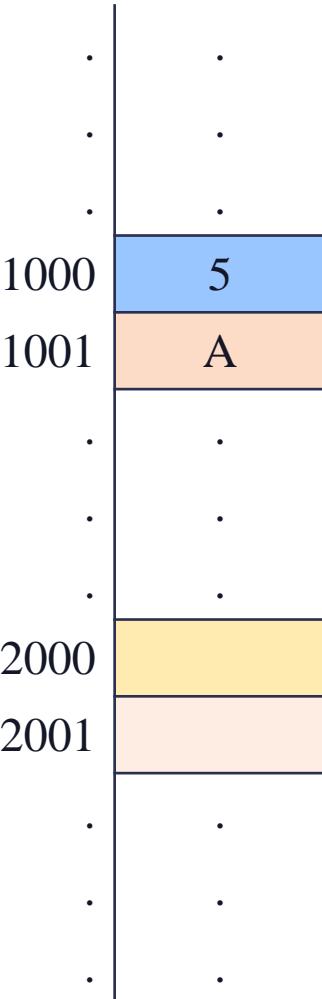
- Bộ nhớ chính - Main memory
 - Lưu trữ thông tin dùng trong quá trình tính toán của máy tính
- Bộ nhớ chính gồm:
 - ROM (Read Only Memory):
 - RAM (Random Access Memory)





Random access memory - RAM

- RAM là bộ nhớ cho phép **đọc, ghi** ngẫu nhiên từ bất kỳ vị trí nào trên bộ nhớ dựa theo **địa chỉ ô nhớ**.
 - Mỗi ô nhớ có một **địa chỉ duy nhất** và địa chỉ này giúp truy cập thông tin được lưu trữ trong nó.
 - RAM **kết nối trực tiếp** với CPU.
 - Các chương trình được tải lên RAM trước khi thực thi.
 - Tất cả dữ liệu phải được đưa vào RAM trước khi chương trình có thể thao tác.
 - Khi máy tính tắt, mọi thứ trong bộ nhớ chính sẽ **bị mất**.



Main Memory with some data



Thiết bị lưu trữ thứ cấp - Secondary storage

- Thiết bị lưu trữ thứ cấp là thiết bị lưu trữ dữ liệu lâu dài, dữ liệu không bị mất đi sau khi tắt máy.
- Ví dụ: đĩa cứng, ổ đĩa flash, đĩa mềm, đĩa ZIP, CD-ROM, DVD, băng từ, lưu trữ cloud, ...



Đĩa mềm



Đĩa zip



USB



Memory card



DVD



CD



Blue Ray



Băng từ



External HDD



HDD



SSD





Thiết bị đầu vào - Input Device

- Các thiết bị đưa dữ liệu và chương trình vào máy tính được gọi là **thiết bị đầu vào**.
- Ví dụ: Bàn phím, chuột, màn hình cảm ứng, mic, webcam, thiết bị lưu trữ thứ cấp, ...



Màn hình cảm ứng



Chuột



Cần điều khiển



Bàn phím



Máy đọc thẻ



Máy scan



mic



Máy đọc mã vạch



webcam

June 2024



Bút cảm ứng



touch pad



Thiết bị đầu ra - Output Devices

- Các thiết bị mà máy tính sử dụng để hiển thị kết quả được gọi là thiết bị đầu ra.
- Ví dụ: Màn hình, máy in, thiết bị lưu trữ thứ cấp, ...





Input - Output Devices

- Một số thiết bị vừa là thiết thi đầu vào vừa là thiết bị đầu ra.
- Ví dụ: màn hình cảm ứng, tai nghe kèm mic, fax, card mạng, Modems, ...



Modem



Màn hình cảm ứng



Máy fax



Tai nghe kèm



Card mạng

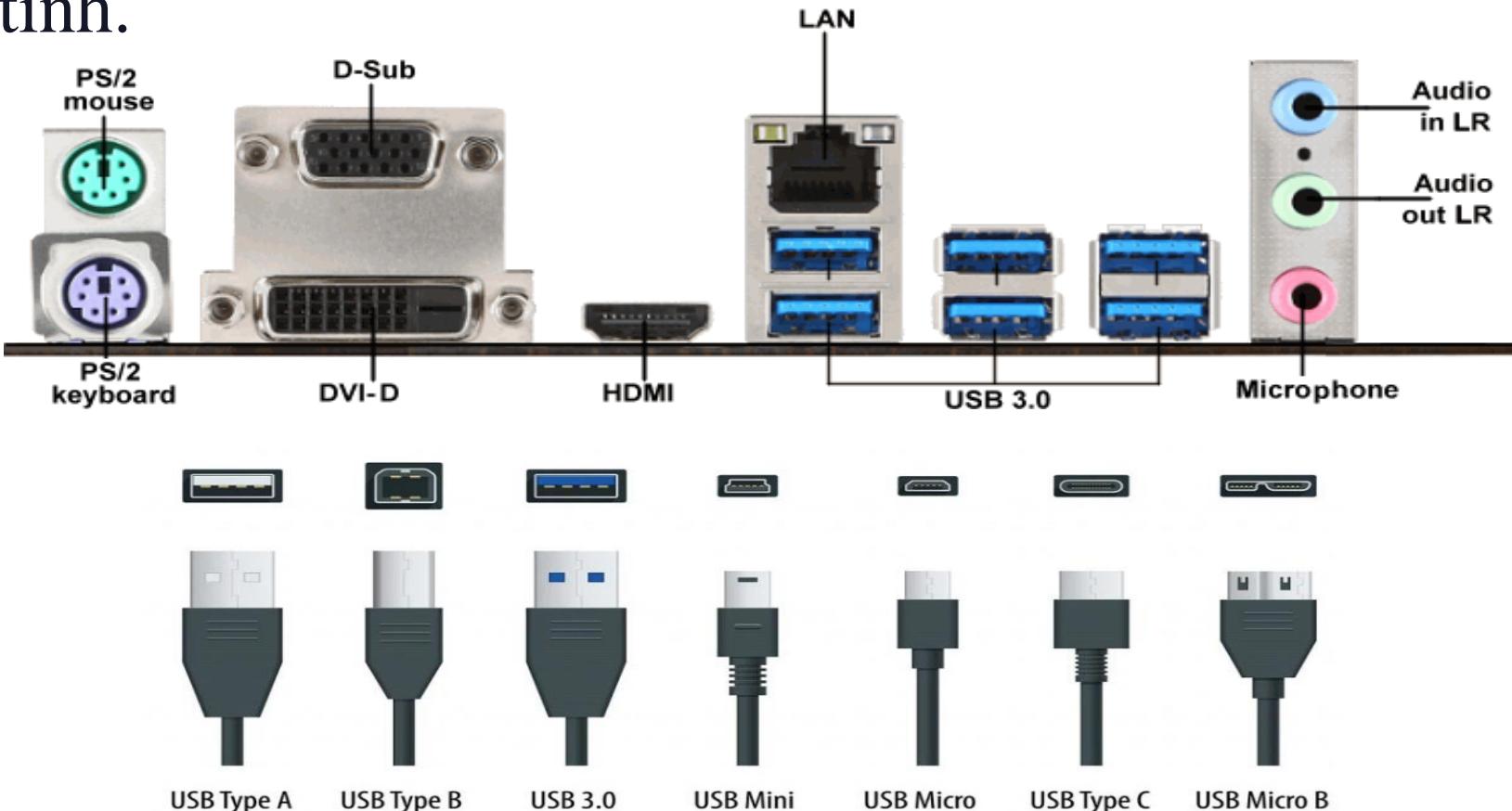


CD-RW



Cổng giao tiếp máy tính

- Các cổng giao tiếp của máy tính được thiết kế trên bản mạch chính có nhiệm vụ kết nối các thiết bị ngoại vi nhằm tương tác với máy tính.

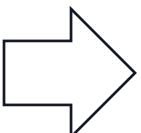




1.1.4 Phần mềm máy tính

- **Phần mềm máy tính** (Computer Software): là một tập hợp những câu lệnh hoặc chỉ thị (Instruction) được viết bằng một hoặc nhiều ngôn ngữ lập trình theo một trật tự xác định nhằm tự động thực hiện một số nhiệm vụ hay chức năng hoặc giải quyết một vấn đề cụ thể nào đó.

```
    string4replace = string4replace.replace("czData", str(key))
    value = float(value) tmpValue = str(row['value'])
    tmpFormat = 14 #Replace string by value's QA temporary format
    tempString = tempString.replace("czDataTyp
    str(key)) tempString = tempString.replace("czFieldID",str(key)) tempString =
    str(value*pow(10,14-tmpFormat))) tempString = tempString.replace("czData", str(key))
    if(typeOffID == "BUFFER"): s = value dataCal =
    tempString.replace("czFieldID",str(key)) tempString =
    str(value) if(typeOffID == "ASCII_STRING"): s = value dataCal =
    tempString = tempString.replace("czData", str(key))
    if("name value=" in line and flagCheckRicnam):
        if "<Message>" in line: myEvent =
        "RT_CHANGE"
        if onlyFilename+"\n" if typeOfFile ==
        "RT_CHANGE": os.path.exists(path): os.makedirs(path)
        if content: searchObj = re.search("filename in ", content)
        if searchObj:
            filename = searchObj.group(1)
```





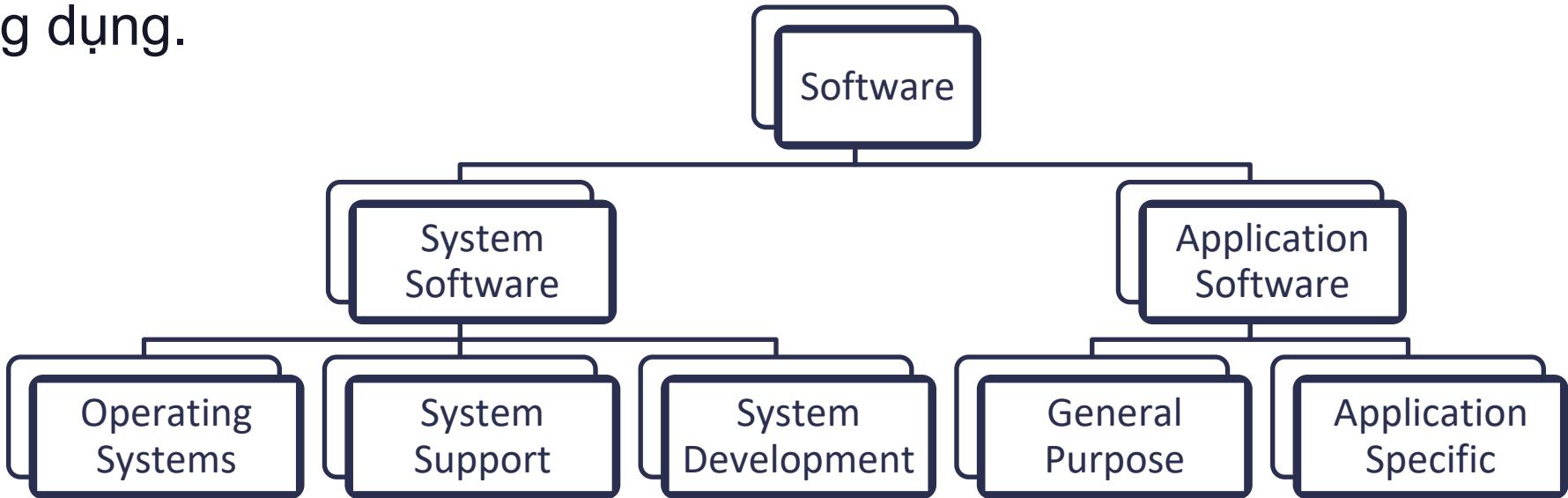
1.1.4 Phần mềm máy tính (tt)

- **Phần mềm** thực hiện các chức năng của nó bằng cách gửi các chỉ thị trực tiếp đến phần cứng (hay phần cứng máy tính, Computer Hardware) hoặc bằng cách cung cấp dữ liệu để phục vụ các chương trình hay phần mềm khác.
- Phần mềm là một khái niệm trừu tượng, nó khác với phần cứng ở chỗ là "phần mềm không thể sờ hay đụng vào", và nó cần phải có phần cứng mới có thể thực thi được.



1.1.4 Phần mềm máy tính (tt)

- Phân loại phần mềm máy tính:
 - Dựa trên tính chất của phần mềm: phần mềm hệ thống, phần mềm ứng dụng.

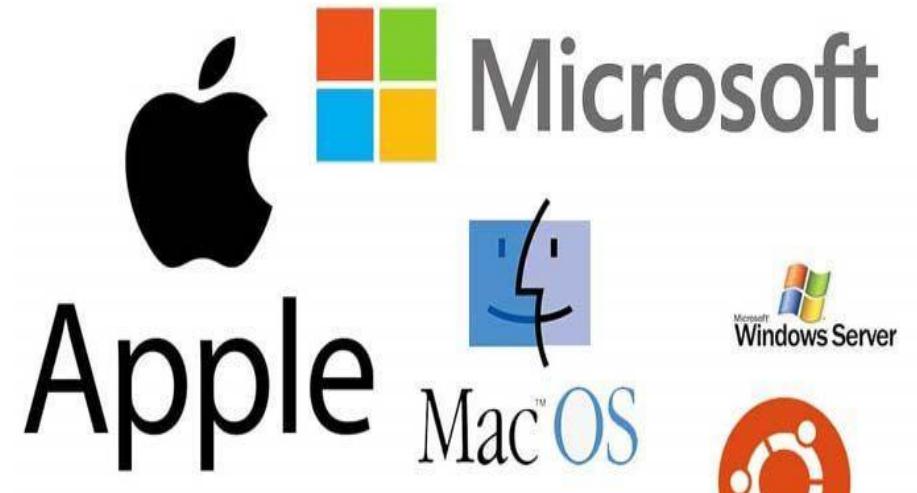


- Dựa trên dạng cung cấp phần mềm: phần mềm mã mở, phần mềm mã nguồn đóng.
- Dựa trên chi phí sử dụng phần mềm: phần mềm thương mại, phần mềm dùng thử, phần mềm miễn phí.



1.1.4 Phần mềm máy tính (tt)

- **Phần mềm hệ thống** (System Software): là phần mềm máy tính thiết kế cho việc **vận hành và điều khiển phần cứng máy tính** và cung cấp một **kiến trúc cho việc chạy phần mềm ứng dụng**. Phần mềm hệ thống có thể được chia thành 3 loại:
 - Hệ điều hành (operating system)
 - Phần mềm tiện ích (utility software or system support program)
 - Các công cụ phát triển phần mềm (software development tools).





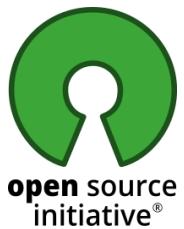
1.1.4 Phần mềm máy tính (tt)

- **Phần mềm ứng dụng** (Application Software): thực hiện một nhiệm vụ cụ thể của người dùng. Ví dụ tiêu biểu cho phần mềm ứng dụng là chương trình xử lý văn bản, bảng tính, chương trình giải trí.





1.1.4 Phần mềm máy tính (tt)



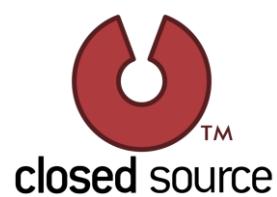
Phần mềm nguồn mở (Open Source Software):

- Là phần mềm với mã nguồn **được công bố** và sử dụng một **giấy phép nguồn mở**.
- Giấy phép này cho phép bất cứ ai cũng có thể nghiên cứu, thay đổi, cải tiến phần mềm và phân phối phần mềm ở dạng chưa thay đổi hoặc đã thay đổi.





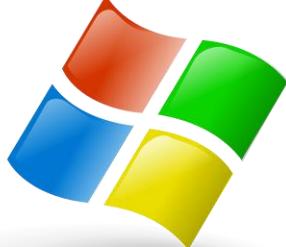
1.1.4 Phần mềm máy tính (tt)



Phần mềm nguồn đóng

(Closed Source Software):

- Mã nguồn không được công bố
- Thông thường sẽ có tính phí sử dụng
- Các hình thức sao chép, sửa đổi, xóa các phần mã nguồn bị xem như là không hợp pháp.



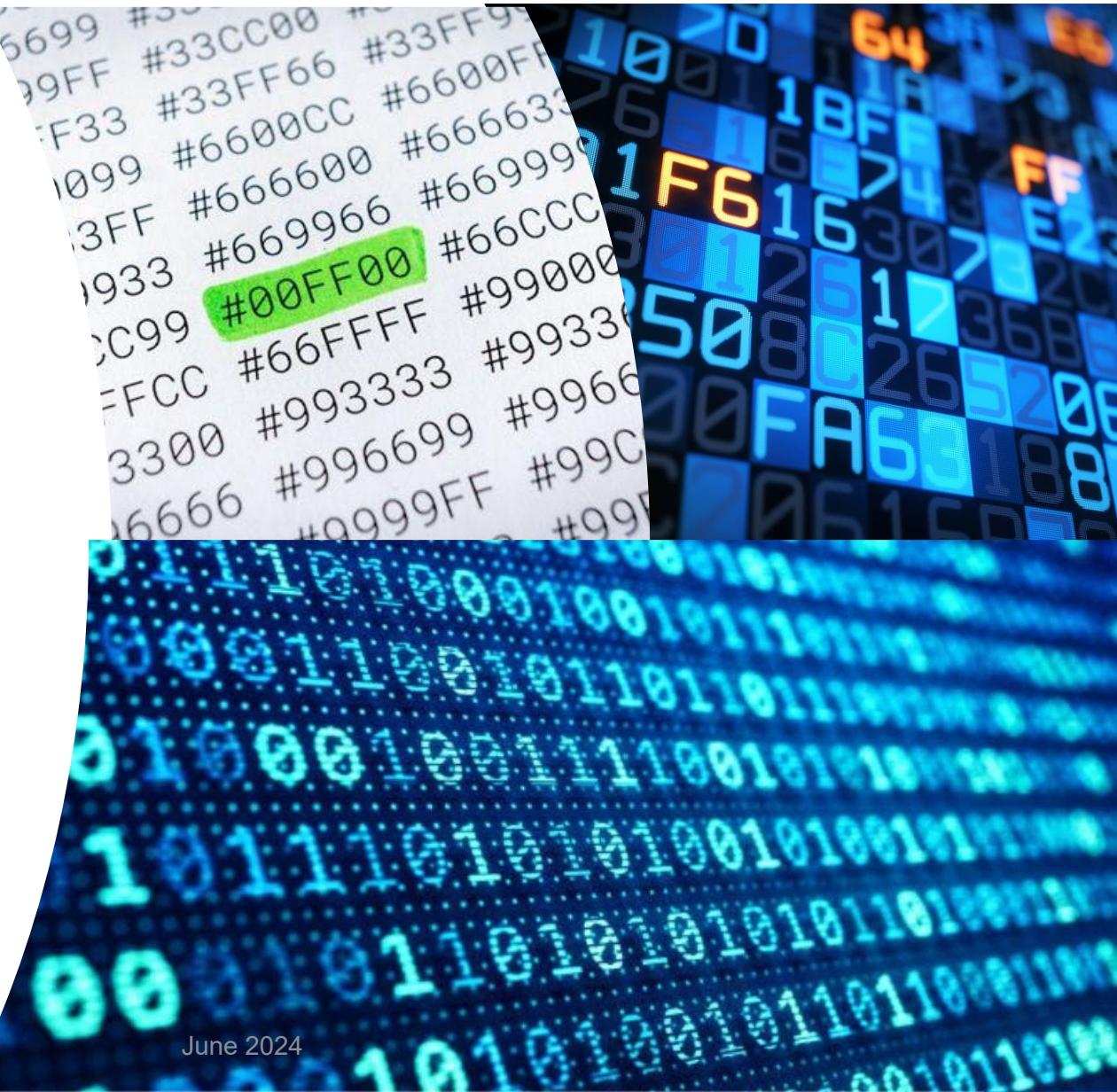
ADOBE
FLASH PLAYER





1.1.5 Biểu diễn và đo lường thông tin

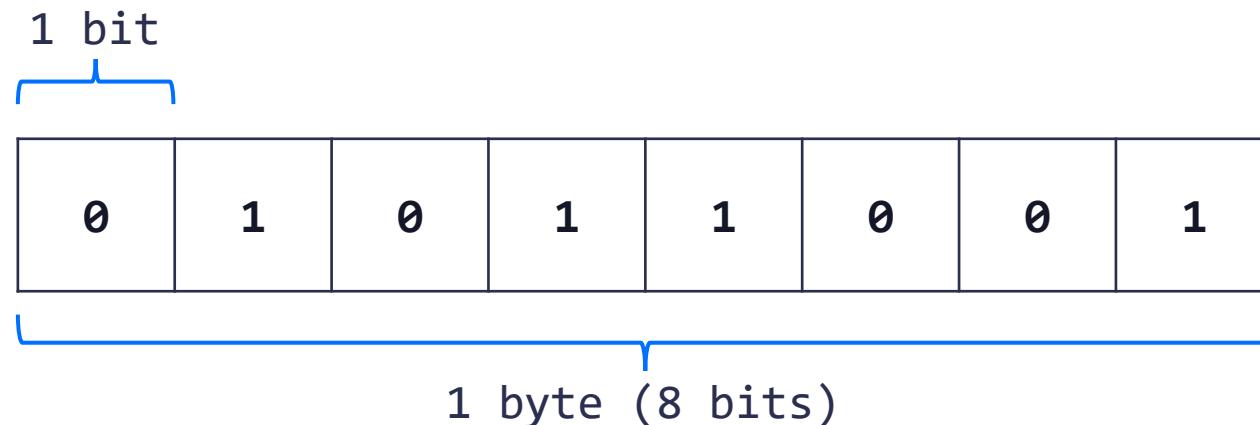
- **Thông tin được biểu diễn bằng hệ đếm nhị phân còn gọi là bit** (chữ viết tắt của **binary digit**), có 2 trạng thái 0 hoặc 1.
- **Có 4 hệ đếm:**
 - Hệ nhị phân (Binary System): 0, 1
 - Hệ thập phân (Decimal System): 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
 - Hệ bát phân (Octal System): 0, 1, 2, 3, 4, 5, 6, 7
 - Hệ thập lục phân (Hexa System): 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F





1.1.5 Biểu diễn và đo lường thông tin (tt)

- **Bit (b)**: trạng thái lưu trữ máy tính hai giá trị 0 và 1.
- **Byte (B)**: tập hợp 8 bit, biểu đạt một ký tự trong ngôn ngữ tự nhiên. Các bội số của Byte còn có KiloByte, MegaByte, GigaByte, TeraByte, PetaByte, ... với độ chênh đơn vị là một bội số của $2^{10} = 1024$.





1.1.5 Biểu diễn và đo lường thông tin (tt)

- Bảng chuyển đổi các đơn vị đo thông tin và hệ đếm:

Tên gọi	Ký hiệu	Giá trị
Bit	b	Binary Digit
Byte	B	$1 \text{ B} = 8 \text{ bit}$
Kilobyte	KB	$1 \text{ KB} = 2^{10} \text{ B} = 1024 \text{ B}$
Megabyte	MB	$1 \text{ MB} = 2^{20} \text{ B} = 1024 \text{ KB}$
Gigabyte	GB	$1 \text{ GB} = 2^{30} \text{ B} = 1024 \text{ MB}$
Terabyte	TB	$1 \text{ TB} = 2^{40} \text{ B} = 1024 \text{ GB}$
Petabyte	PB	$1 \text{ PB} = 2^{50} \text{ B} = 1024 \text{ TB}$
Exabyte	EB	$1 \text{ EB} = 2^{60} \text{ B} = 1024 \text{ PB}$
Zettabyte	ZB	$1 \text{ ZB} = 2^{70} \text{ B} = 1024 \text{ EB}$
Yottabyte	YB	$1 \text{ YB} = 2^{80} \text{ B} = 1024 \text{ ZB}$
Brontobyte	BB	$1 \text{ BB} = 2^{90} \text{ B} = 1024 \text{ YB}$
Geopbyte	GeB	$1 \text{ GeB} = 2^{100} \text{ B} = 1024 \text{ BB}$

Dec	Hex	Oct	Bin
0	0	0	0
1	1	1	1
2	2	2	10
3	3	3	11
4	4	4	100
5	5	5	101
6	6	6	110
7	7	7	111
8	8	10	1000
9	9	11	1001
10	A	12	1010
11	B	13	1011
12	C	14	1100
13	D	15	1101
14	E	16	1110
15	F	17	1111



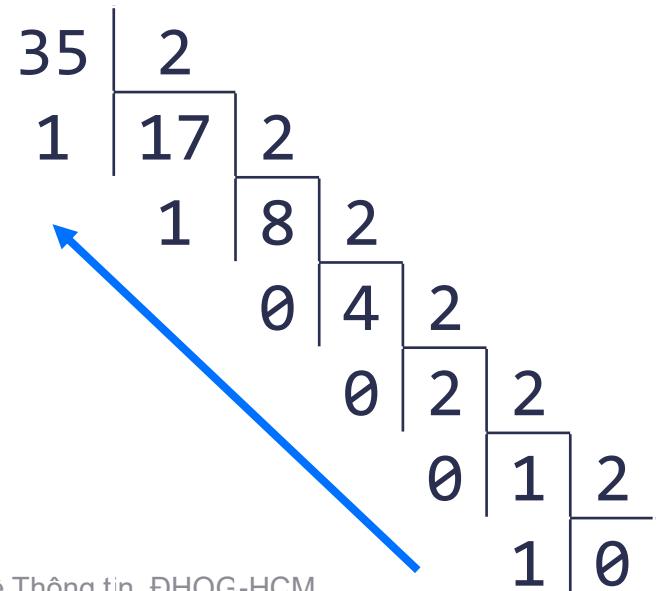
Cách chuyển đổi qua lại giữa các hệ đếm

Quy tắc 1: Chuyển số từ hệ thập phân sang hệ có cơ số b

Lấy số thập phân chia cho cơ số b cho đến khi phần thương của phép chia bằng 0, số đổi được chính là phần dư của phép chia theo thứ tự ngược lại.

Ví dụ: $35_{10} = ?_2$

Ta có:



Kết luận: $35_{10} = 100011_2$



Cách chuyển đổi qua lại giữa các hệ đếm

Quy tắc 2: Chuyển số từ hệ có cơ số b sang hệ thập phân

Công thức: $X_b = a_n a_{n-1} \dots a_1 a_0 = a_n b^n + a_{n-1} b^{n-1} + \dots + a_1 b + a_0$

Trong đó:

b: hệ cơ số đếm

$a_n, a_{n-1}, \dots, a_1, a_0$: các chữ số cơ bản

X: số ở hệ đếm cơ số b

Ví dụ: $1BF_{16} = ?_{10}$

Ta có: $1BF_{16} = 1 * 16^2 + B * 16^1 + F * 16^0 = 1 * 256 + 11 * 16 + 15 * 1 = 447_{10}$



Cách chuyển đổi qua lại giữa các hệ đếm

Quy tắc 3: Chuyển số từ hệ nhị phân sang hệ thập lục phân

Nhóm lần lượt 4 bit từ phải qua trái, thay thế các nhóm 4 bit bằng giá trị tương ứng với hệ 16 (tra theo bảng)

Ví dụ: $1111011_2 = ?_{16}$

Nhóm lần lượt 4 bit từ phải sang trái, ta được:

$$0111'1011_2$$

$$(0111)_2 = (7)_{16}$$

$$(1011)_2 = (B)_{16}$$

Kết luận: $1111011_2 = 7B_{16}$

Bin	Hex
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F



Cách chuyển đổi qua lại giữa các hệ đếm

Quy tắc 4: Chuyển số từ hệ nhị phân sang hệ bát phân

Nhóm lần lượt 3 bit từ phải qua trái, thay thế các nhóm 3 bit bằng giá trị tương ứng với hệ cơ số 8 (tra theo bảng):

$$\text{Ví dụ: } 1111011_2 = ?_8$$

Nhóm lần lượt 3 bit từ phải sang trái, ta được:

$$001'111'011_2$$

$$(001)_2 = (1)_8$$

$$(111)_2 = (7)_8$$

$$(011)_2 = (3)_8$$

$$\text{Kết luận: } 1111011_2 = 173_8$$

Binary	Oct
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7



Cách chuyển đổi qua lại giữa các hệ đếm

Quy tắc 5: Chuyển số từ hệ thập lục phân sang hệ nhị phân

Ứng với mỗi chữ số sẽ được biểu diễn dưới dạng 4 bit nhị phân (tra theo bảng)

Ví dụ: $4AE_{16} = ?_2$

Ta có:

$$(4)_{16} = (0100)_2$$

$$(A)_{16} = (1010)_2$$

$$(E)_{16} = (1110)_2$$

Kết luận: $4AE_{16} = 100'1010'1110_2 = 10010101110_2$

Bin	Hex
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F



Cách chuyển đổi qua lại giữa các hệ đếm

Quy tắc 6: Chuyển số từ hệ bát phân sang hệ nhị phân

Üng với mỗi chữ số sẽ được biểu diễn dưới dạng 3 bit nhị phân (tra theo bảng):

Ví dụ: $276_8 = ?_2$

Ta có:

$$(2)_8 = (010)_2$$

$$(7)_8 = (111)_2$$

$$(6)_8 = (110)_2$$

Kết luận: $276_8 = 010'111'110_2 = 01011110_2$

Binary	Oct
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7



Cách chuyển đổi qua lại giữa các hệ đếm

Quy tắc 7: Chuyển đổi khác

Khi chuyển đổi 2 cơ số bất kỳ khác 10, ta có thể chọn hệ đếm thập phân làm hệ trung gian:

$$X_b \xrightarrow{} Y_{10} \xrightarrow{} Z_a$$



1.2 Các khái niệm cơ bản về lập trình



1.2 Các khái niệm cơ bản về lập trình

1.2.1 Chương trình máy tính

1.2.2 Lập trình máy tính, lập trình viên

1.2.3 Mã nguồn, mã máy

1.2.4 Ngôn ngữ lập trình

1.2.5 Chương trình dịch: Trình biên dịch,
trình thông dịch



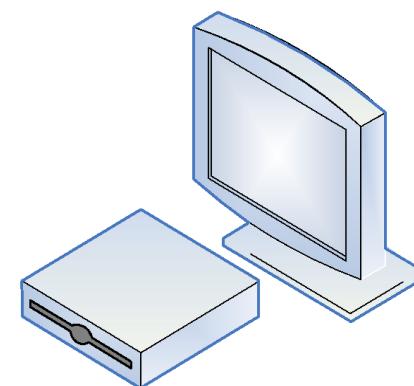


1.2.1 Chương trình máy tính

- Chương trình máy tính** là tập hợp các chỉ dẫn được thể hiện dưới dạng các lệnh, các mã, lược đồ hoặc bất kỳ dạng nào khác, khi gắn vào một phương tiện mà máy tính đọc được, có khả năng làm cho máy tính thực hiện được một công việc hoặc đạt được một kết quả cụ thể.



- Timer Recording**
1. Turn on
 2. Set Channel to **ch01**
 3. Set Date to **1/5/2009**
 4. Set Time to **3:00am**
 5. Confirm setting



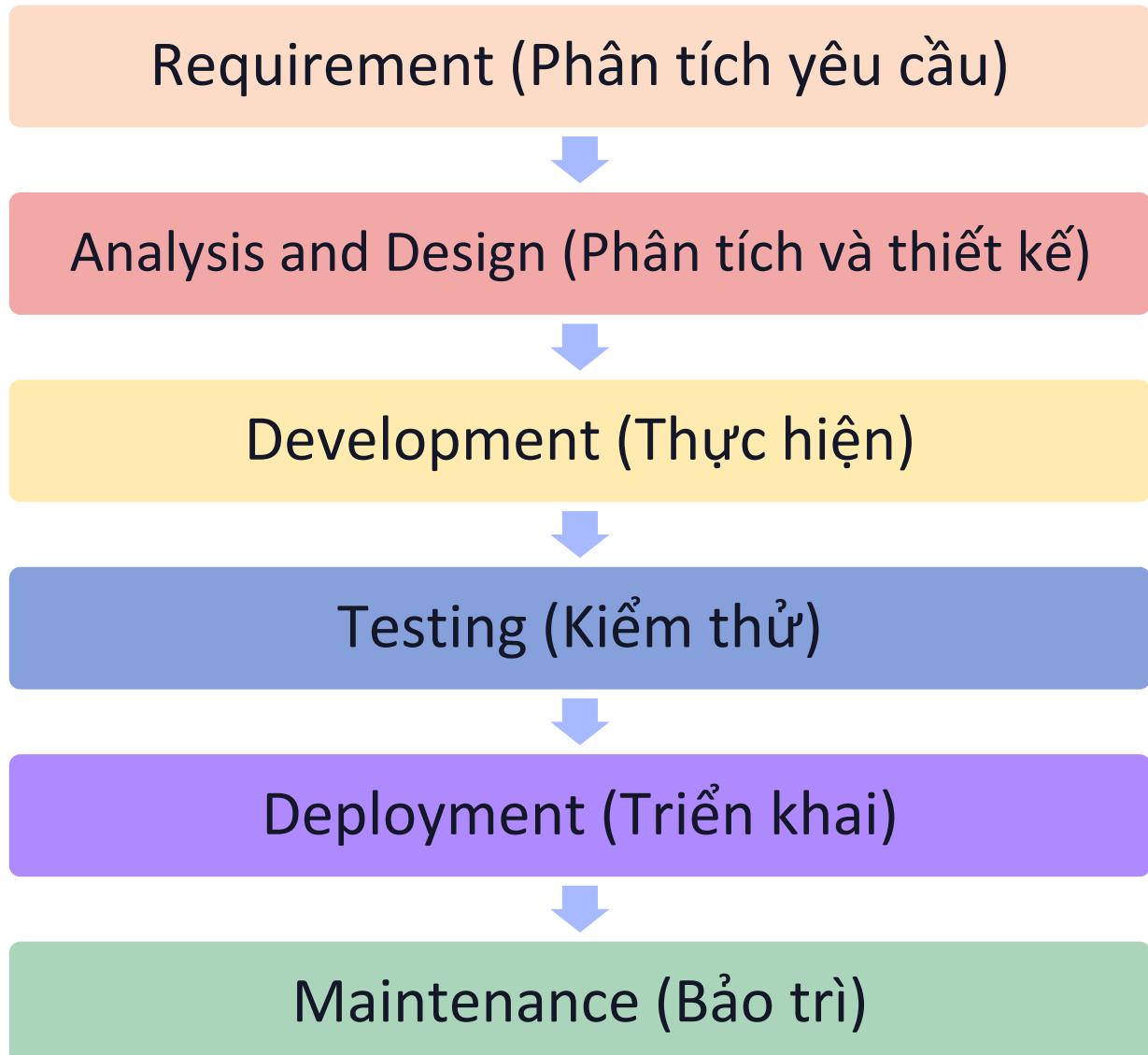
Program X

```
int x=10;  
int y=11;  
y+=x;  
System.out.println(y);  
System.out.println(x);
```



1.2.2 Chương trình máy tính

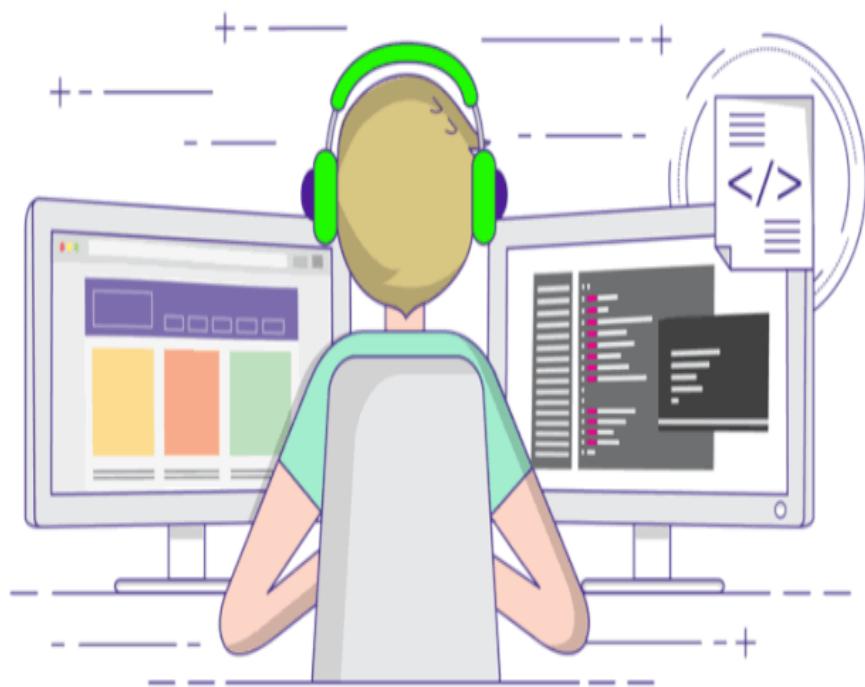
- Các bước xây dựng chương trình máy tính theo Mô hình phát triển phần mềm **Water Fall**:





1.2.2 Lập trình máy tính, lập trình viên

- **Lập trình máy tính** (programming): 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.
- **Lập trình viên** (programmer): Lập trình viên (người lập trình hay thao chương viên điện toán) là người viết ra các chương trình máy tính.

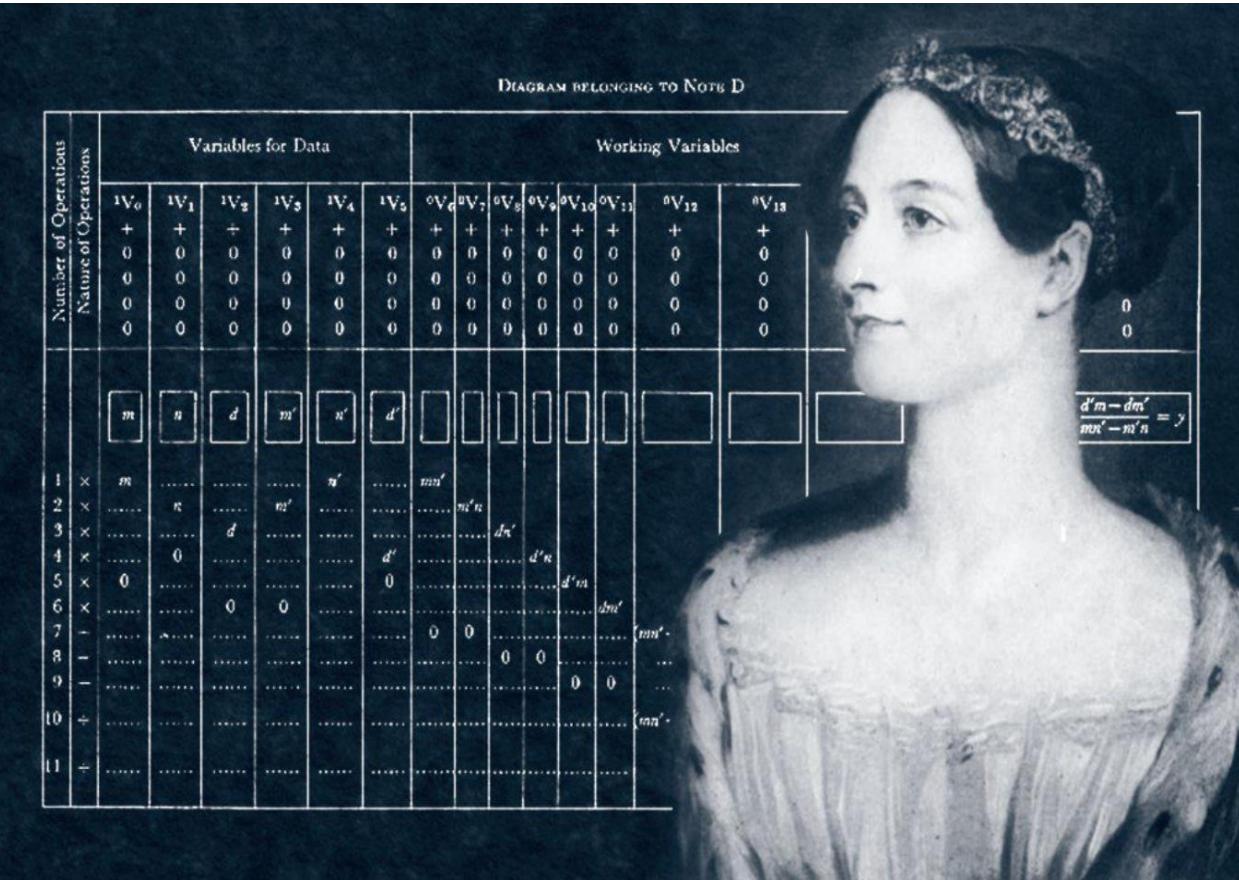




1.2.2 Lập trình máy tính, lập trình viên (tt)

- Ada Lovelace (sinh năm 1852)

là một nhà toán học người Anh, người được mệnh danh là lập trình viên máy tính đầu tiên viết ra một thuật toán (algorithm) với bộ hướng dẫn vận hành cho chiếc máy tính sơ khai do Charles Babbage chế tạo vào năm 1821.





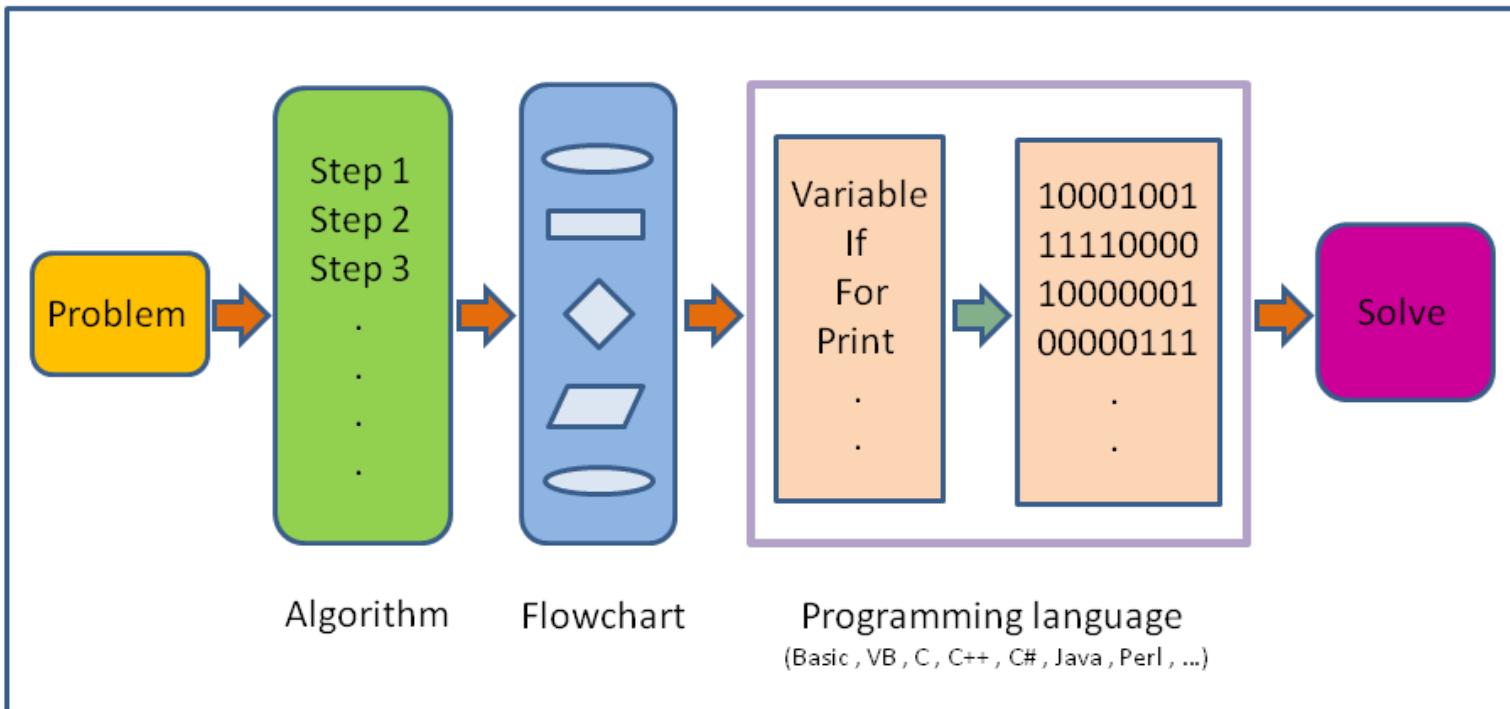
1.2.3 Mã nguồn, mã máy

- **Mã nguồn (source code):** là một chuỗi các câu lệnh hay chỉ thị được viết bằng ngôn ngữ lập trình. Mã nguồn thường được lưu trong tập tin văn bản (text file).
- **Mã máy (machine code):** là một loại mã sử dụng để viết các chỉ dẫn mà **CPU** máy tính có thể hiểu và thực thi được một cách trực tiếp. Mã máy được viết sử dụng các số trong hệ nhị phân (*chỉ gồm 0 và 1 hoặc hệ cơ số 16*)



1.2.4 Ngôn ngữ lập trình (Programming Language)

- **Ngôn ngữ lập trình** là ngôn ngữ dùng để viết các chương trình cho máy tính. Cũng như các ngôn ngữ thông thường, ngôn ngữ lập trình cũng có từ vựng, cú pháp và ngữ nghĩa.

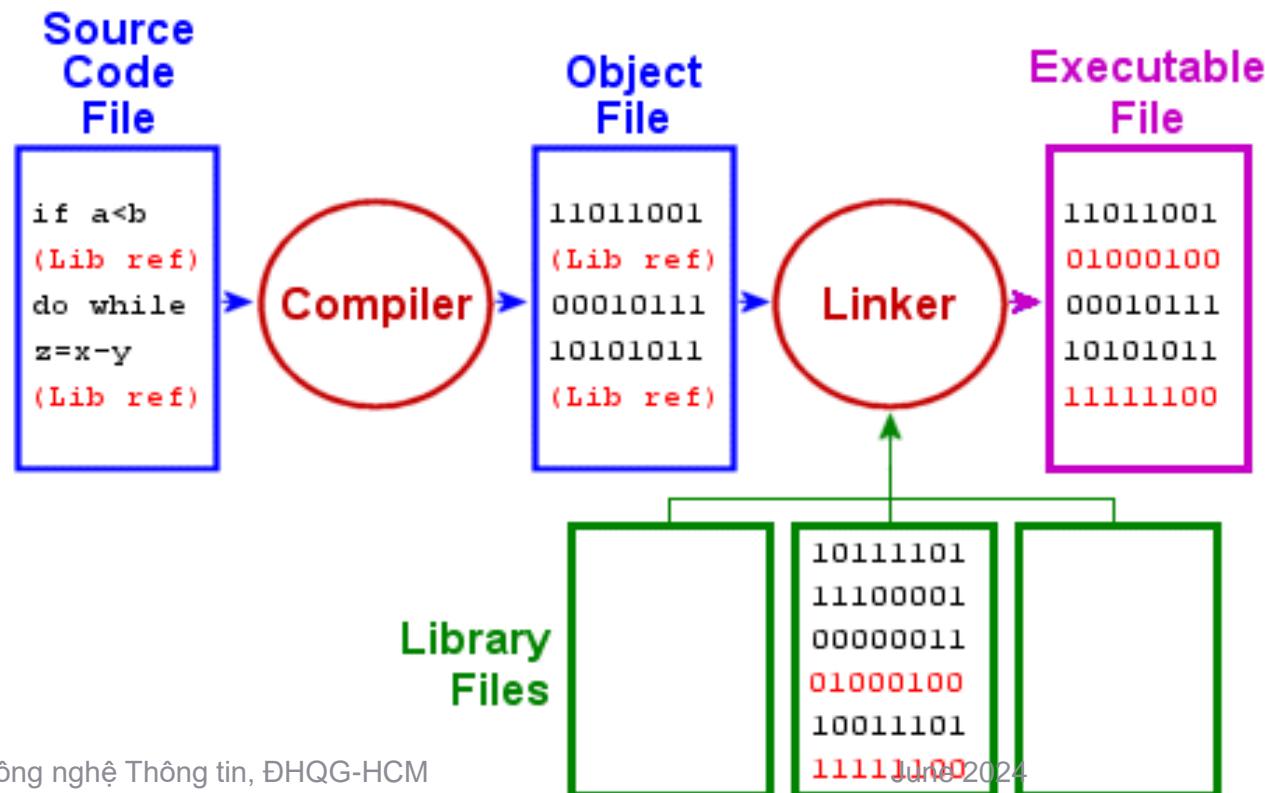


Nguồn: https://en.wikiversity.org/wiki/Introduction_to_Programming/About_Programming



1.2.5 Chương trình dịch

- Chương trình dịch là chương trình đặc biệt có chức năng chuyển đổi chương trình được viết bằng ngôn ngữ lập trình bậc cao thành chương trình thực hiện được trên máy tính.





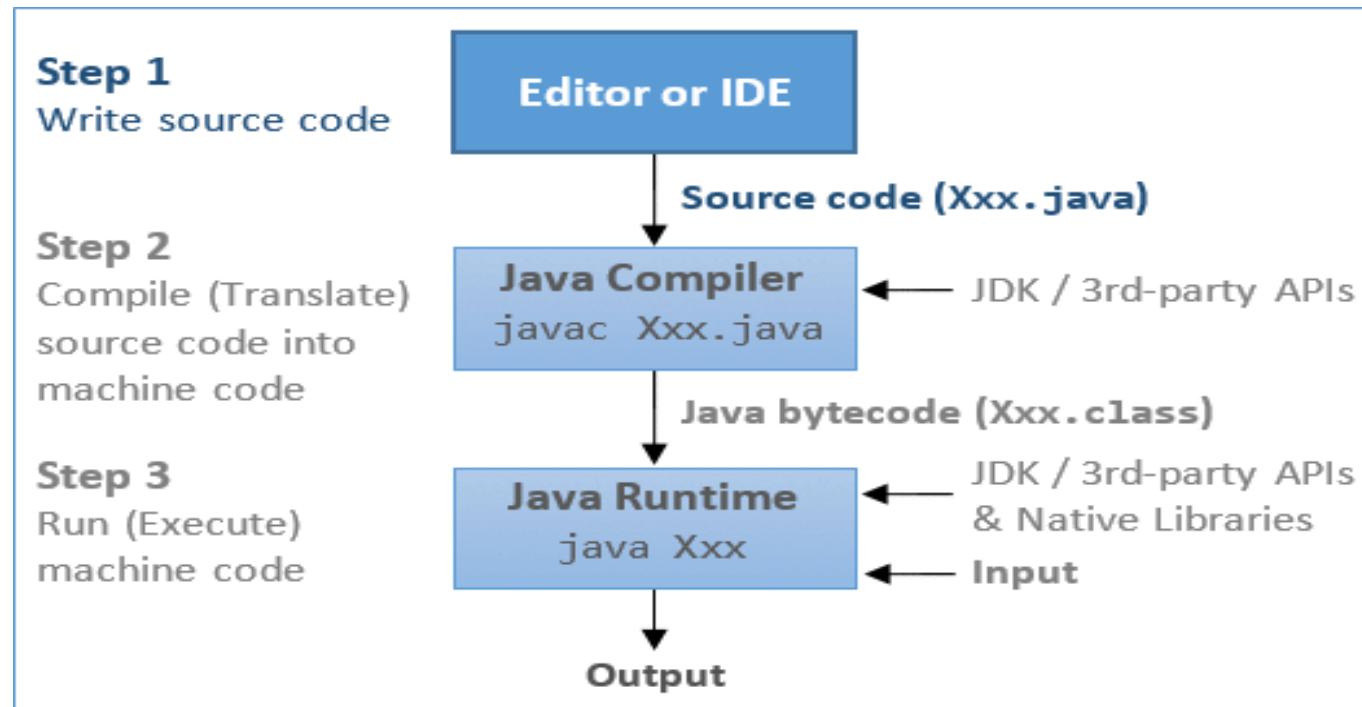
1.2.5 Chương trình dịch (tt)

- **Trình thông dịch (interpreter):** Kiểm tra tính đúng đắn của câu lệnh chuyển đổi thành ngôn ngữ máy và thực thi câu lệnh đó mà không cần dịch toàn bộ mã nguồn, tức là dịch tới đâu sẽ thực thi tới đó.
- **Trình biên dịch (compiler):** Dịch toàn bộ mã nguồn (source code) của một ngôn ngữ lập trình cụ thể sang một ngôn ngữ cấp thấp hơn (hợp ngữ, mã máy, mã bytecode). Trong quá trình dịch chương trình sẽ thông báo lỗi nếu có.
 - Ví dụ: Các trình biên dịch C++ phổ biến như: **GNU C++, Visual C++, Intel C++, Clang, Borland C++**



1.2.5 Chương trình dịch (tt)

- Một số ngôn ngữ lập trình kết hợp cả compiler và interpreter, chẳng hạn như Java: Mã Java được biên dịch thành mã bytecode → Máy ảo chạy mã bytecode ở dạng thông dịch.





1.3 Giới thiệu sơ lược về ngôn ngữ lập trình



1.3 Giới thiệu sơ lược về ngôn ngữ lập trình

1.3.1 Vai trò của Ngôn ngữ lập trình

1.3.2 Ngôn ngữ lập trình cấp thấp

1.3.3 Ngôn ngữ lập trình cấp cao

1.3.4 Các ngôn ngữ lập trình thông dụng



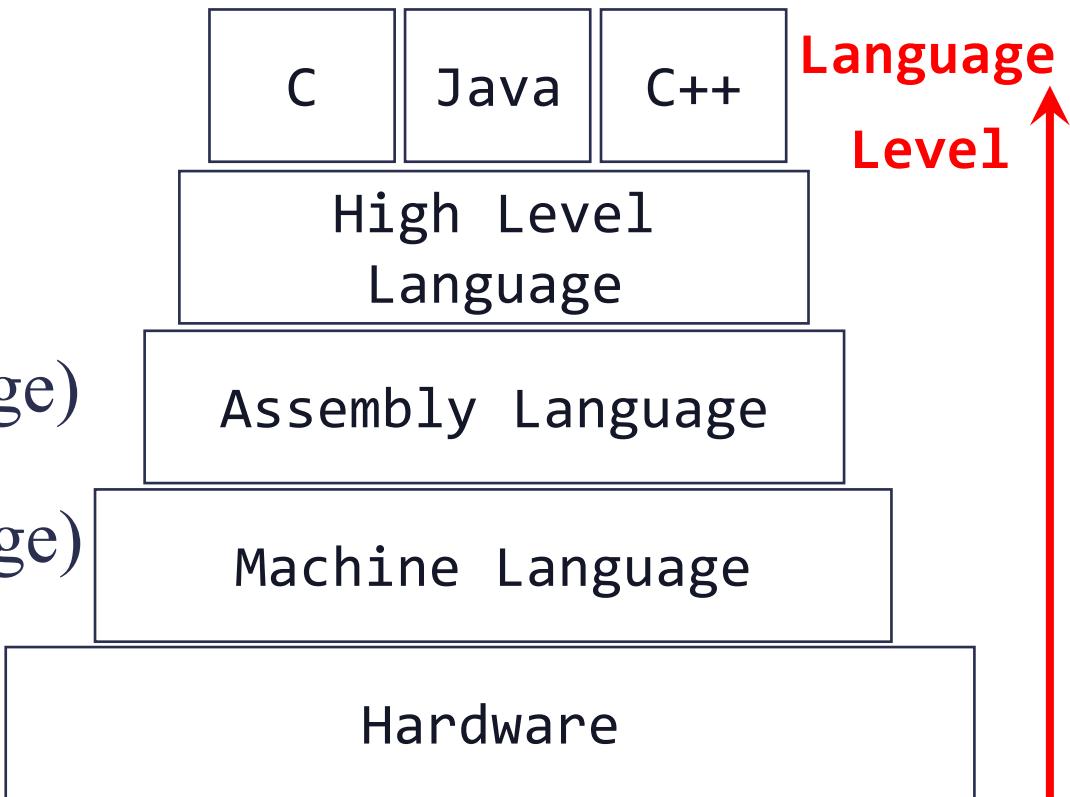
1.3.1 Vai trò của ngôn ngữ lập trình

- **Giao tiếp giữa người và máy tính**
- **Xây dựng ứng dụng và phần mềm:** Tạo ra các ứng dụng và phần mềm từ các định dạng ý tưởng ban đầu.
- **Tối ưu hóa hiệu suất:** Sử dụng ngôn ngữ phù hợp để tối ưu hóa cho nhu cầu cụ thể.
- **Hỗ trợ cho cộng đồng phát triển:** Cung cấp thư viện, framework và công cụ hỗ trợ cho việc phát triển phần mềm và chia sẻ kiến thức trong cộng đồng.
- **Khả năng mở rộng và duy trì:** Đảm bảo tính mở rộng và dễ bảo trì của mã nguồn trong quá trình phát triển và duy trì dự án.



1.3.2 Ngôn ngữ lập trình

- Ba cấp độ ngôn ngữ lập trình :
 - Ngôn ngữ máy (machine language)
 - Ngôn ngữ hợp ngữ (assembly language)
 - Ngôn ngữ cấp cao (high-level language)





Ngôn ngữ cấp thấp

- **Ngôn ngữ máy (*machine language*)** là các chỉ thị dưới dạng nhị phân, can thiệp trực tiếp vào trong các mạch điện tử.
- Chương trình được viết bằng ngôn ngữ máy thì có thể được thực hiện ngay không cần qua bước trung gian nào.

Machine Language

Language directly
understood by the
computer

.....

binary code





Ngôn ngữ cấp thấp

- Ví dụ:

1	00000000	00000100	0000000000000000
2	01011110	00001100	11000010 00000000000010
3		11101111	00010110 000000000000101
4		11101111	10011110 000000000001011
5	11111000	10101101	11011111 000000000010010
6		01100010	11011111 000000000010101
7	11101111	00000010	11111011 000000000010111
8	11110100	10101101	11011111 000000000011110
9	00000011	10100010	11011111 000000000010001
10	11101111	00000010	11111011 0000000000100100
11	01111110	11110100	10101101
12	11111000	10101110	11000101 0000000000101011
13	00000110	10100010	11111011 0000000000110001
14	11101111	00000010	11111011 0000000000110100
15		01010000	11010100 0000000000111011
16			00000100 0000000000111101

Tuy nhiên chương trình viết bằng ngôn ngữ máy dễ sai sót, cồng kềnh và khó đọc, khó hiểu vì toàn những con số 0 và 1.



Ngôn ngữ cấp thấp

- **Hợp ngữ** (Assembly language) được thiết kế để máy tính trở nên thân thiện hơn với người sử dụng.
- Các câu lệnh bao gồm hai phần: phần mã lệnh (viết tựa tiếng Anh) chỉ phép toán cần thực hiện và phần tên biến chỉ địa chỉ chứa toán hạng của phép toán đó.

Machine Language

Language directly understood by the computer

binary code

Symbolic Language

English-like abbreviations representing elementary computer operations

assembly language





Ngôn ngữ cấp thấp

- Ví dụ:

```
section .data
    hello:    db 'Hello, World!',10      ; 'Hello, World!' plus a linefeed character
    helloLen: equ $-hello              ; Length of the 'Hello world!' string

section .text
    global _start

_start:
    mov eax,4          ; The system call for write (sys_write)
    mov ebx,1          ; File descriptor 1 - standard output
    mov ecx,hello     ; Put the offset of hello in ecx
    mov edx,helloLen  ; helloLen is a constant, so we don't need to say
                      ; mov edx,[helloLen] to get it's actual value
    int 80h           ; Call the kernel
    mov eax,1          ; The system call for exit (sys_exit)
    mov ebx,0          ; Exit with return "code" of 0 (no error)
    int 80h;
```

- Để máy thực hiện được một chương trình viết bằng **hợp ngữ** thì chương trình đó phải được dịch sang ngôn ngữ máy. Công cụ thực hiện việc dịch đó được gọi là **Assembler**.



Ngôn ngữ cấp cao

- Ngôn ngữ cấp cao (High level language): là ngôn ngữ được tạo ra và phát triển nhằm phản ánh cách thức người lập trình nghĩ và làm.
- Ngôn ngữ cấp cao rất gần với ngôn ngữ con người (tiếng Anh) nhưng chính xác như ngôn ngữ toán học.

Machine Language

Language directly understood by the computer

binary code

Symbolic Language

English-like abbreviations representing elementary computer operations

assembly language

High-level Language

Close to human language.

Example: $a = a + b$

[add values of a and b , and store the result in a , replacing the previous value]

C, C++, Java, Basic





Các ngôn ngữ lập trình thông dụng

- Hiện nay có rất nhiều ngôn ngữ lập trình, nhưng phổ biến là Python, C, C++, Java, C#, JavaScript, PHP,...

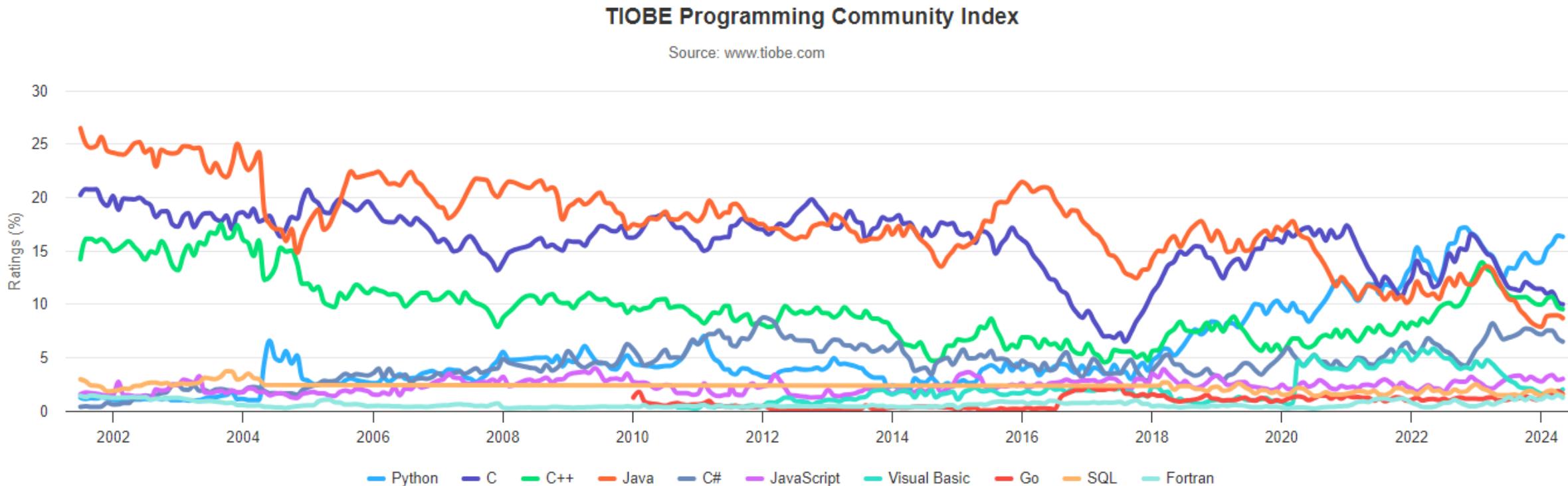
Bảng sau liệt kê top 10 các ngôn ngữ lập trình phổ biến từ tháng 05/2023-05/2024:

May-24	May-23	Programming Language	Ratings
1	1	Python	16.33%
2	2	C	9.98%
3	4	C++	9.53%
4	3	Java	8.69%
5	5	C#	6.49%
6	7	JavaScript	3.01%
7	6	Visual Basic	2.01%
8	12	Go	1.60%
9	9	SQL	1.44%
10	19	Fortran	1.24%

<https://www.tiobe.com/tiobe-index/>



Các ngôn ngữ lập trình thông dụng



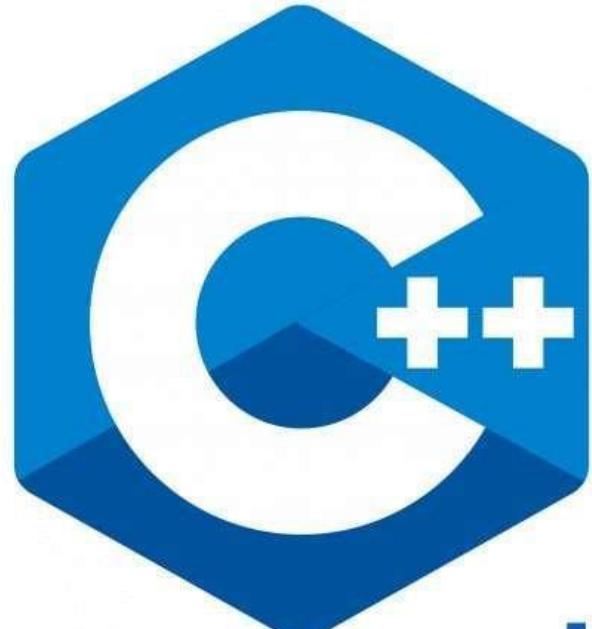
<https://www.tiobe.com/tiobe-index/>



1.4 Giới thiệu ngôn ngữ lập trình C++



1.4 Giới thiệu ngôn ngữ lập trình C++



- 1.4.1 Giới thiệu tổng quan về ngôn ngữ C++
- 1.4.2 Các ưu điểm và ứng dụng của C++
- 1.4.3 Môi trường, công cụ hỗ trợ lập trình C++
- 1.4.4 Biên dịch chương trình C++
- 1.4.5 Một số ví dụ minh họa chương trình C++



1.4.1 Giới thiệu tổng quan về ngôn ngữ C++



- C++ được phát triển bởi Bjarne Stroustrup năm 1979 tại Bell Labs ở Murray Hill, New Jersey, như là một bản nâng cao của ngôn ngữ C và với tên gọi đầu tiên là “C với các Lớp”, nhưng sau đó được đổi tên thành C++ vào năm 1983.
- Hiện tại tiêu chuẩn mới nhất của ngôn ngữ C++ là C++20.
- C++ là ngôn ngữ bậc cao (high-level languages), C++ hỗ trợ lập trình thủ tục và lập trình hướng đối tượng.



Tiêu chuẩn ngôn ngữ C++

Năm	Tiêu chuẩn C++	Tên gọi
1998	ISO/IEC 14882:1998 ^[11]	C++98
2003	ISO/IEC 14882:2003 ^[12]	<u>C++03</u>
2011	ISO/IEC 14882:2011 ^[13]	<u>C++11</u> , C++0x
2014	ISO/IEC 14882:2014 ^[14]	<u>C++14</u> , C++1y
2017	ISO/IEC 14882:2017 ^[8]	<u>C++17</u> , C++1z
2020	ISO/IEC 14882:2020 ^[15]	<u>C++20</u> ^[16] , C++2a

<https://en.wikipedia.org/wiki/C%2B%2B>



1.4.2 Ưu điểm của ngôn ngữ C++

- **Tốc độ xử lý nhanh:** Mã nguồn C++ được biên dịch và gửi thẳng xuống phần cứng tiếp với phần cứng
- **Hỗ trợ lập trình hướng đối tượng:**
 - OOP xem đối tượng là trọng tâm, mỗi đối tượng bao gồm thuộc tính kèm theo các thao tác liên quan trên đối tượng.
 - Code dễ hiểu, dễ phát triển và bảo trì.
- **Hỗ trợ tối đa thư viện:** Thư viện template của C++ (STL)
- **Hỗ trợ con trỏ:** có khả năng truy xuất trực tiếp tới ô nhớ trong RAM



1.4.2 Ưu điểm của ngôn ngữ C++

- **Sử dụng lại mã nguồn với mẫu** (templates)
 - Templates cho phép bạn viết mã tổng quát hơn, có thể tái sử dụng cho nhiều kiểu dữ liệu khác nhau mà không cần phải viết lại mã cho từng kiểu dữ liệu cụ thể.
- **Xử lý lỗi**: C++ exception handling giúp phát hiện và xử lý lỗi trong quá trình thực thi chương trình
- **Lập trình quy mô lớn**:
 - Khả năng quản lý tài nguyên và điều khiển chi tiết quá trình thực thi giúp C++ thích hợp cho các dự án phần mềm lớn.



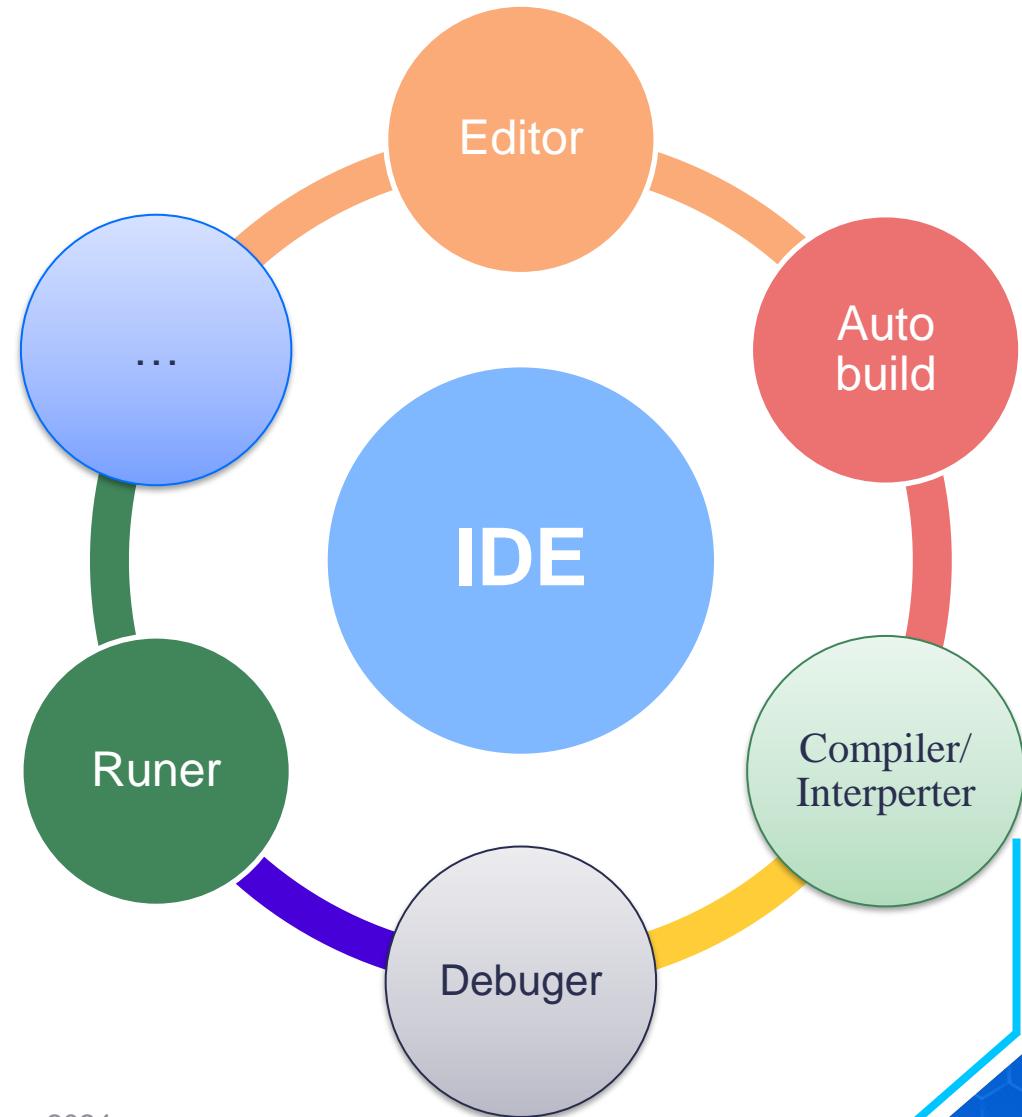
Ứng dụng của ngôn ngữ C++

- **Hệ điều hành:** MS Window, IOS, Apple OSX, ...
- **Trình duyệt web:** Google chrome, Mozilla Firefox, Internet Explorer ...
- **Phát triển Game:** Counter-Strike, World of Warcraft, ...
- **Phần mềm quản trị cơ sở dữ liệu:** MySQL, Oracle
- **Trình biên dịch:** Dev-C++, Clang C++, MINGW, ...
- **Nền tảng của một số ngôn ngữ lập trình:** C#, Java, PHP, Matlab
- **Ứng dụng có giao diện người dùng (GUI):** Microsoft Office, IDE Visual Studio, Adobe Photoshop, ...
- **Hệ thống nhúng**
- ...



1.4.3 Môi trường, công cụ hỗ trợ lập trình C++

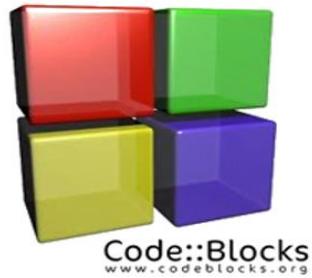
- Môi trường phát triển tích hợp **IDE (Integrated Development Environment)**
 - Trình soạn thảo (**SOURCE CODE EDITOR**)
 - Trình Biên dịch/ Thông dịch (**COMPILER/INTERPETER**)
 - Công cụ xây dựng tự động (**AUTO BUILD**)
 - Trình gõ lỗi hỗ trợ dò tìm lỗi (**Trình DEBUGER**)
 - ...





1.4.3 Môi trường, công cụ hỗ trợ lập trình C++

- Một số IDE hỗ trợ lập trình C++:



NetBeans. IDE



Xcode



CLion



Codelite

IDE multiplataforma para C/C++



Builder
embarcadero



1.4.3 Môi trường, công cụ hỗ trợ lập trình C++

- IDE CodeBlocks:

The screenshot shows the CodeBlocks 20.03 IDE interface. The title bar reads "*main.cpp [Bai 1] - Code::Blocks 20.03". The menu bar includes File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, DoxyBlocks, Settings, and Help. The toolbar has icons for New, Open, Save, Build, Run, Stop, and Exit. The left sidebar has a Management panel with Projects, Files, and FSys tabs, showing a Workspace with a Bai 1 project containing Sources and main.cpp. The main editor window displays the following C++ code:

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5
6     cout << "Welcome to my class.";
7
8     return 0;
9 }
10
```

The status bar at the bottom shows D/C/C++ Windows (CR+LF) UTF-8 Line 10, Col 1, Pos 116 Insert Modified Read/Write default.

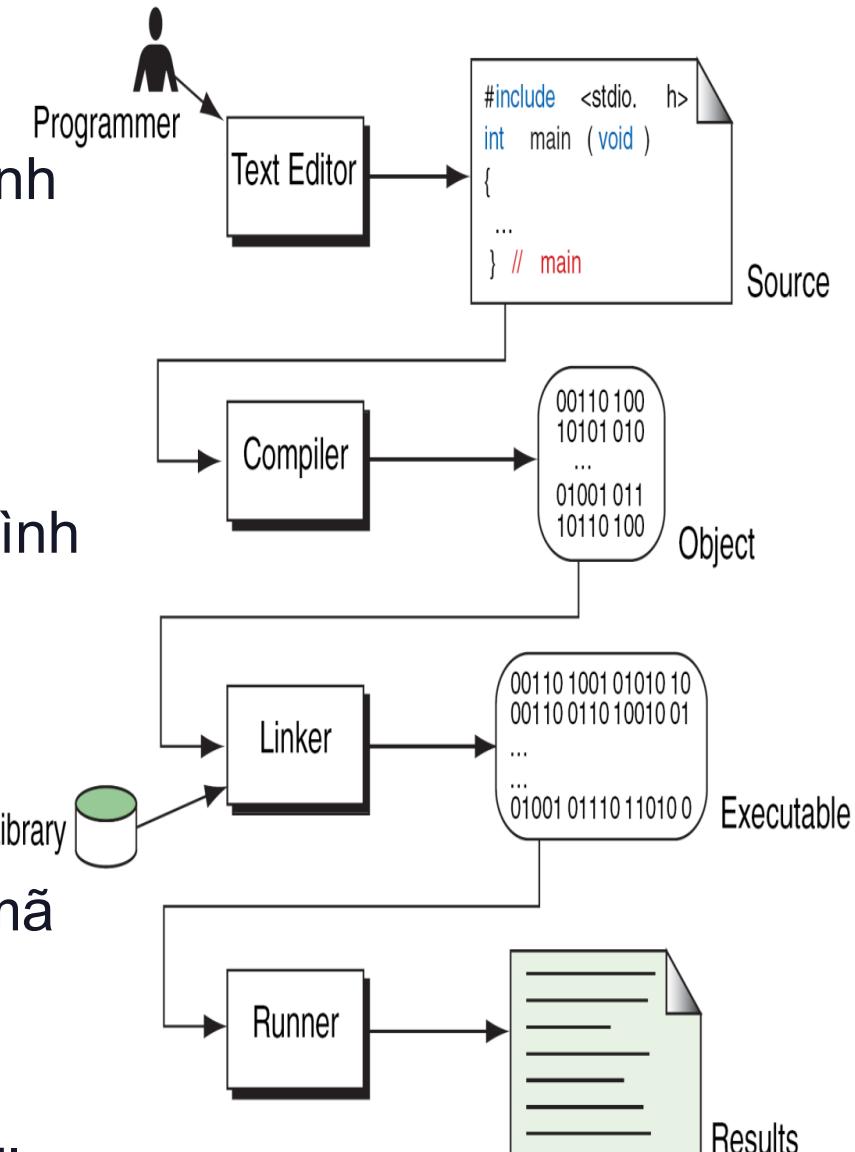


1.4.4 Biên dịch chương trình C++

- Source code (mã chương trình): file chứa mã chương trình C++ => đầu ra file “.cpp”.

Các bước biên dịch chương trình C++:

- **Preprocessing (Tiền xử lý)** => đầu ra file “.i”
 - Nhận mã nguồn, xóa bỏ tất cả chú thích của chương trình
 - Chỉ thị tiền xử lý được xử lý
- **Compilation (Biên dịch)** => đầu ra file “.o”, “.obj”
 - Kiểm tra lỗi cú pháp
 - Chuyển source code thành các file object code chứa mã máy
- **Linking (Trình liên kết)** => đầu ra file .exe, .dll
 - Kết hợp các object code và các thư viện để tạo thành file thực thi hoặc file thư viện.





1.4.5 Ví dụ minh họa chương trình C++

- **Ví dụ 1:** Nhập xuất đơn giản như “Hello World”.
- **Ví dụ 2:** Chương trình có nhập xuất dữ liệu và tính toán xử lý đơn giản như “Nhập độ dài 2 cạnh của hình chữ nhật, xuất diện tích của hình”.
- **Ví dụ 3:** Chương trình phức tạp hơn, có sử dụng vòng lặp: kiểm tra một số nguyên n có phải là số nguyên tố không.



Ví dụ 1

- Xuất đơn giản như “Hello World”:

```
/* The traditional first program in honor of  
Dennis Ritchie who invented C at Bell Labs  
in 1972 */  
  
#include <iostream>  
using namespace std;  
int main() {  
    cout << "Hello, world!\n";  
    return 0;  
}
```



Ví dụ 2

- Chương trình có nhập xuất dữ liệu và tính toán xử lý đơn giản như “Nhập độ dài 2 cạnh của hình chữ nhật, xuất diện tích của hình”

```
/* Minh hoa chuong trinh tinh dien tich hinh chu nhat */

#include <iostream>
using namespace std;
void main(){
    int chieu_dai, chieu_rong;
    cout<< "Nhap chieu dai = ";
    cin>> chieu_dai;
    cout<< "Nhap chieu rong = ";
    cin>> chieu_rong;
    // Tinh dien tich hinh chu nhat
    int dien_tich = chieu_dai*chieu_rong;
    // In ket qua ra man hinh
    cout << "Dien tich HCN = "<<dien_tich;
    return 0;
}
```



Ví dụ 3

- Chương trình phức tạp hơn, có sử dụng vòng lặp: kiểm tra một số nguyên n có phải là số nguyên tố không.

```
/* Chuong trinh minh hoa kiem tra so nguyen to */
#include <iostream>
using namespace std;
// Ham kiem tra so nguyen n co phai la so nguyen to (true) hay khong (false).
bool kiemtralasonguyento(int n) {
    if( n<2) return false;
    for(int i=2; i<n;i++)
        if(n%i==0) return false;
    return true;
}
int main() {
    int n;
    cout << "Nhập vào số nguyên n = ";
    cin >> n;
    if (kiemtralasonguyento(n)== true)
        cout << "So "<<n<<" la so nguyen to !";
    else
        cout << "So "<<n<<" khong phai la so nguyen to !";
    return 0;
}
```



Bài tập lý thuyết

- Câu 1: Cho biết ý nghĩa của các thành phần cấu tạo phần cứng máy tính cơ bản.
- Câu 2: Vì sao máy tính điện tử lại sử dụng hệ nhị phân để biểu diễn thông tin
- Câu 3: Đổi các số ở hệ cơ số a sang hệ cơ số b tương ứng:

$11000110101_2 = ?_{10}$	$175_8 = ?_{10}$	$2AB_{16} = ?_2$
$11000110101_2 = ?_{16}$	$2AB_{16} = ?_{10}$	$256_8 = ?_2$
$11000110101_2 = ?_8$		

- Câu 4: Chứng minh dãy có n bit thì có thể biểu diễn được số lớn nhất $2^n - 1$. Ví dụ: $n=2 \Rightarrow 2^2 - 1 = 3$, $n=4 \Rightarrow 2^4 - 1 = 15$



Bài tập lý thuyết

Câu 5: Trình bày cấu trúc tổng quan của máy tính.

Câu 6: Sự khác nhau của trình biên dịch và trình thông dịch.

Câu 7: Cho biết các ưu điểm và các ứng dụng của ngôn ngữ C++

Câu 8: Trình bày quá trình biên dịch chương trình C++



Tài liệu tham khảo

- D.S. Malik, “C++ Programming: From Problem Analysis to Program Design, 5th Edition, 2011”
- Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman, “Compilers: Principles, Techniques and Tools, 2nd edition”



Chúc các em học tốt!

