



# **BÀI 01:**

# **TỔNG QUAN VỀ MÁY TÍNH**

# **VÀ LẬP TRÌNH**

1. Tổng quan về máy tính
2. Các khái niệm cơ bản về lập trình:
3. Các ngôn ngữ lập trình
4. Giới thiệu ngôn ngữ C++, chương trình C++.
5. Ví dụ minh họa về chương trình C++ và chạy thử.
6. Một số quy tắc cần nhớ khi viết chương trình

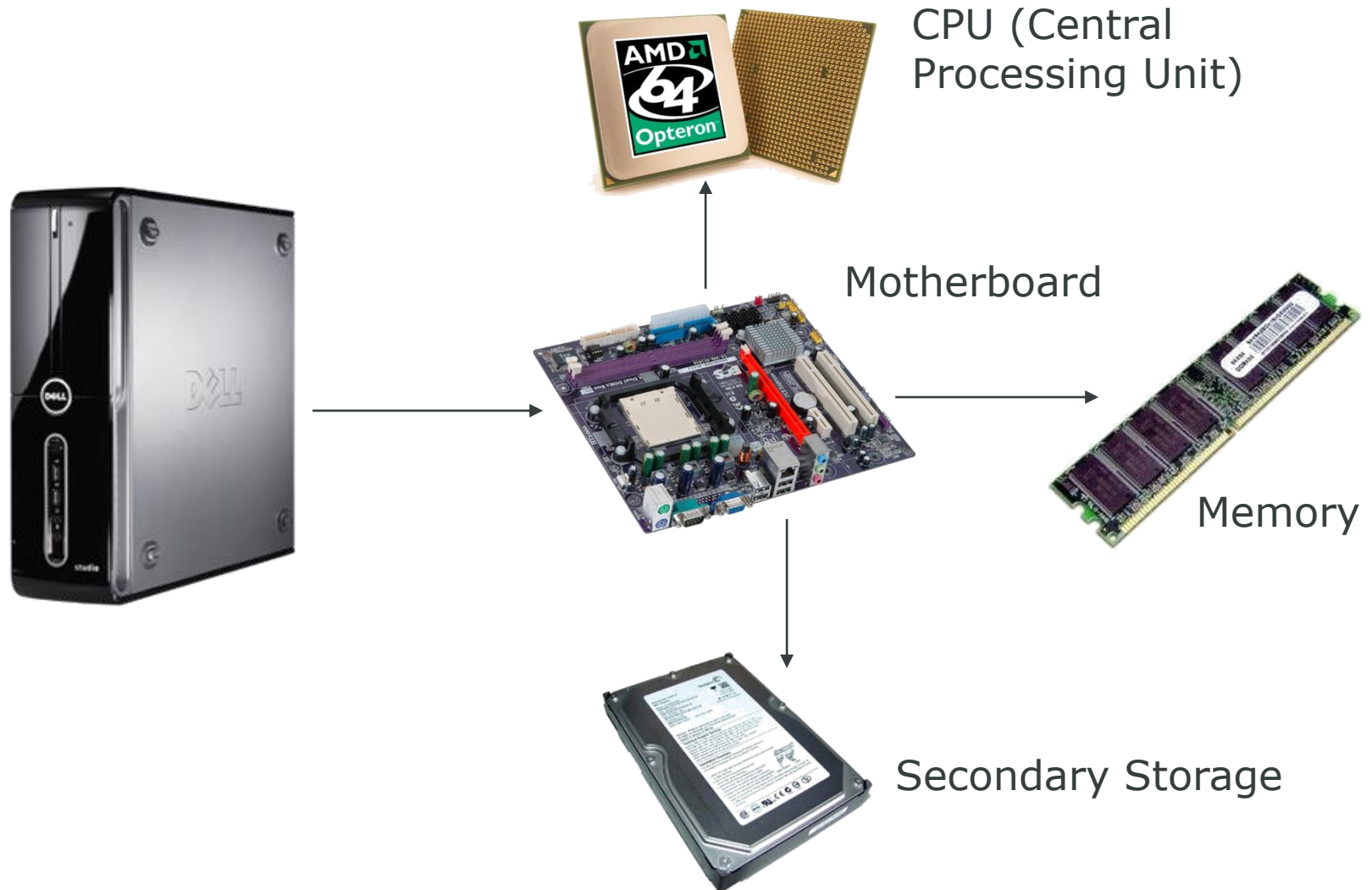


# 1. Tổng quan về máy tính



- a) Máy tính là gì?
- b) Cấu trúc tổng quan của máy tính?
- c) Các thiết bị nội vi, ngoại vi máy tính?
- d) Phần mềm máy tính?
- e) Thông tin được biểu diễn và đo lường trên máy tính?
- f) Máy tính xử lý được thông tin?

# 1. a) Máy tính (Computer) là gì ?



# 1. a) Máy tính (Computer) là gì ?



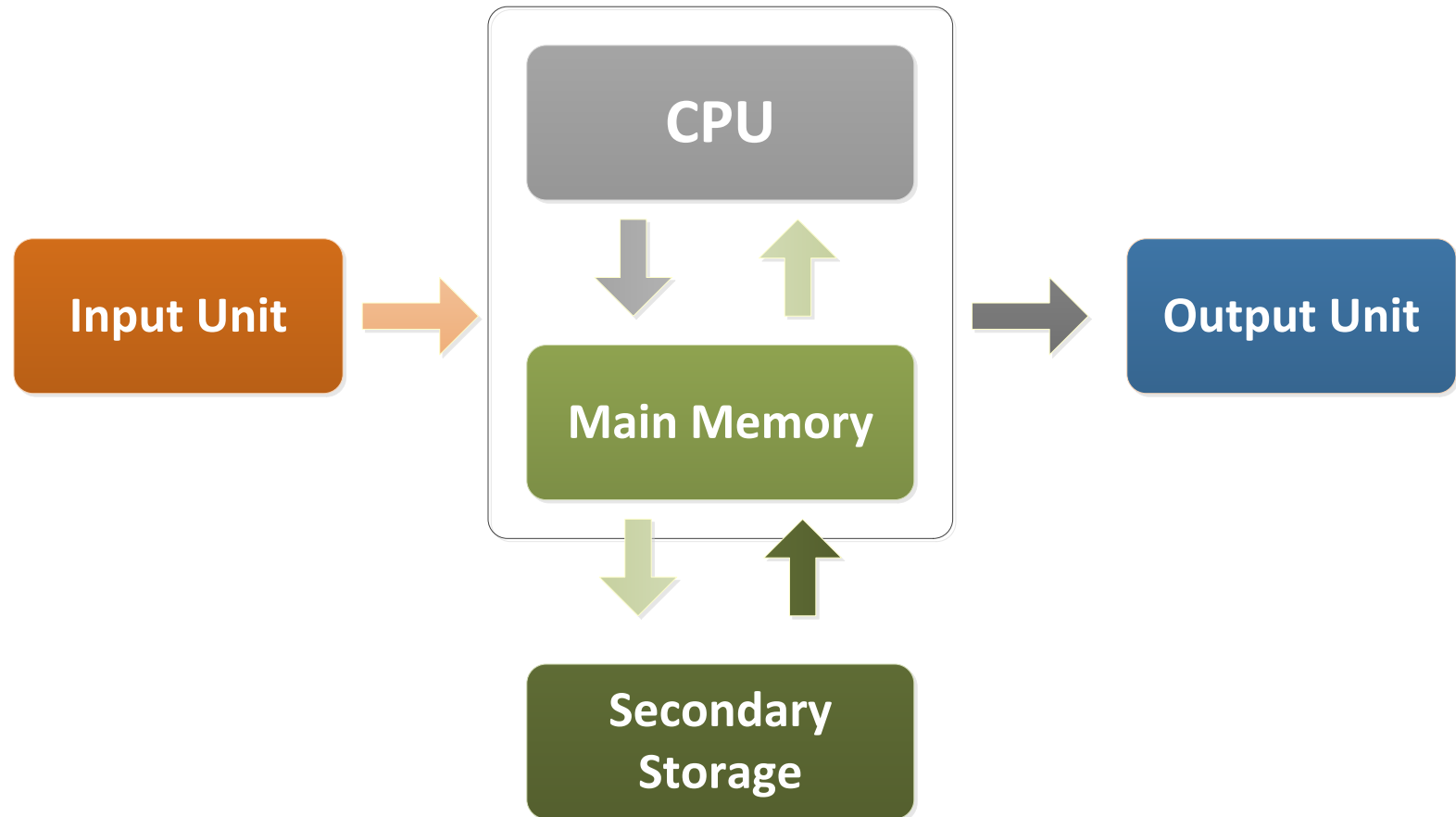
- ❖ Máy tính là những thiết bị hay hệ thống dùng để tính toán hoặc kiểm soát các hoạt động mà có thể biểu diễn dưới dạng số hoặc qui luật logic



# 1. b) Cấu trúc tổng quan của máy tính



## ❖ Cấu trúc tổng quan của máy tính



# 1. c) Các thiết bị nội vi, ngoại vi



## ❖ Các thiết bị nhập, xuất



**Input Unit**

**Thiết bị nhập (Input Unit):**  
nhận dữ liệu từ người dùng hoặc từ các chương trình khác



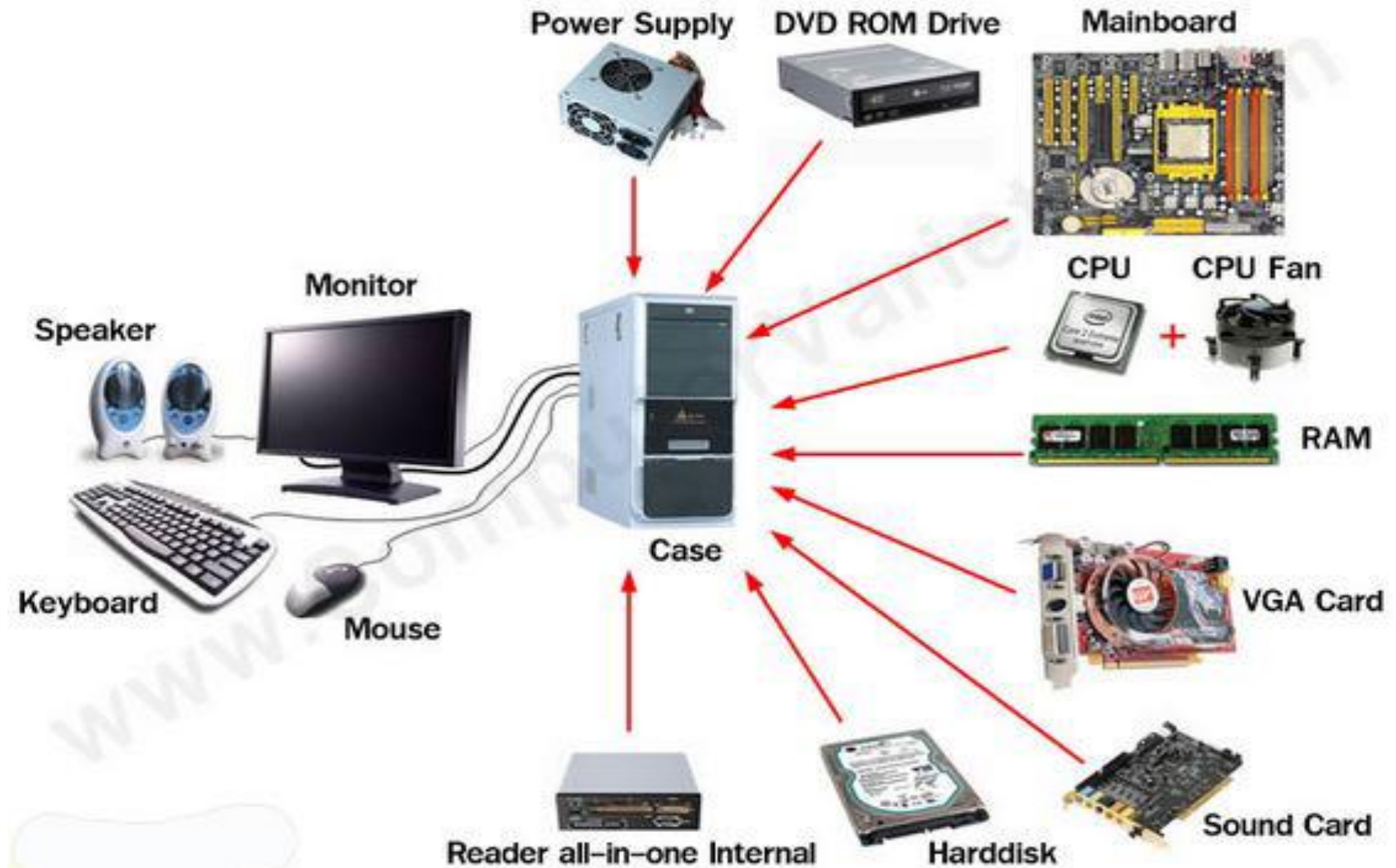
**Output Unit**

**Đơn vị xuất (Output Unit):**  
hiển thị kết quả cho người dùng hoặc cho các chương trình khác.

# 1. c) Các thiết bị nội vi, ngoại vi



## ❖ Phần cứng máy tính





# 1. c) Các thiết bị nội vi, ngoại vi



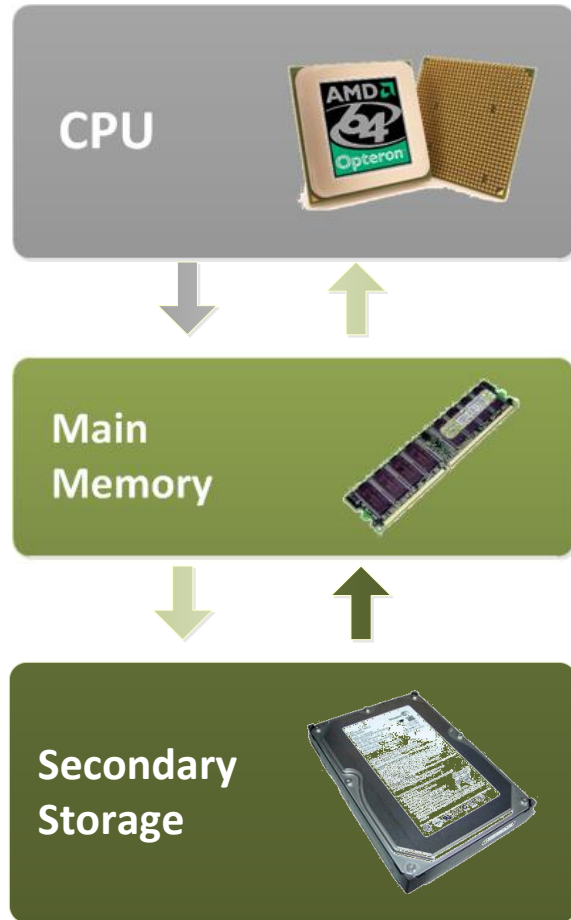
## ❖ Phần cứng máy tính



# 1. c) Các thiết bị nội vi, ngoại vi



## ❖ Các thiết bị nội vi.



**CPU (Central Processing Unit):** Đọc các chỉ thị từ bộ nhớ chính và thực thi các chỉ thị.

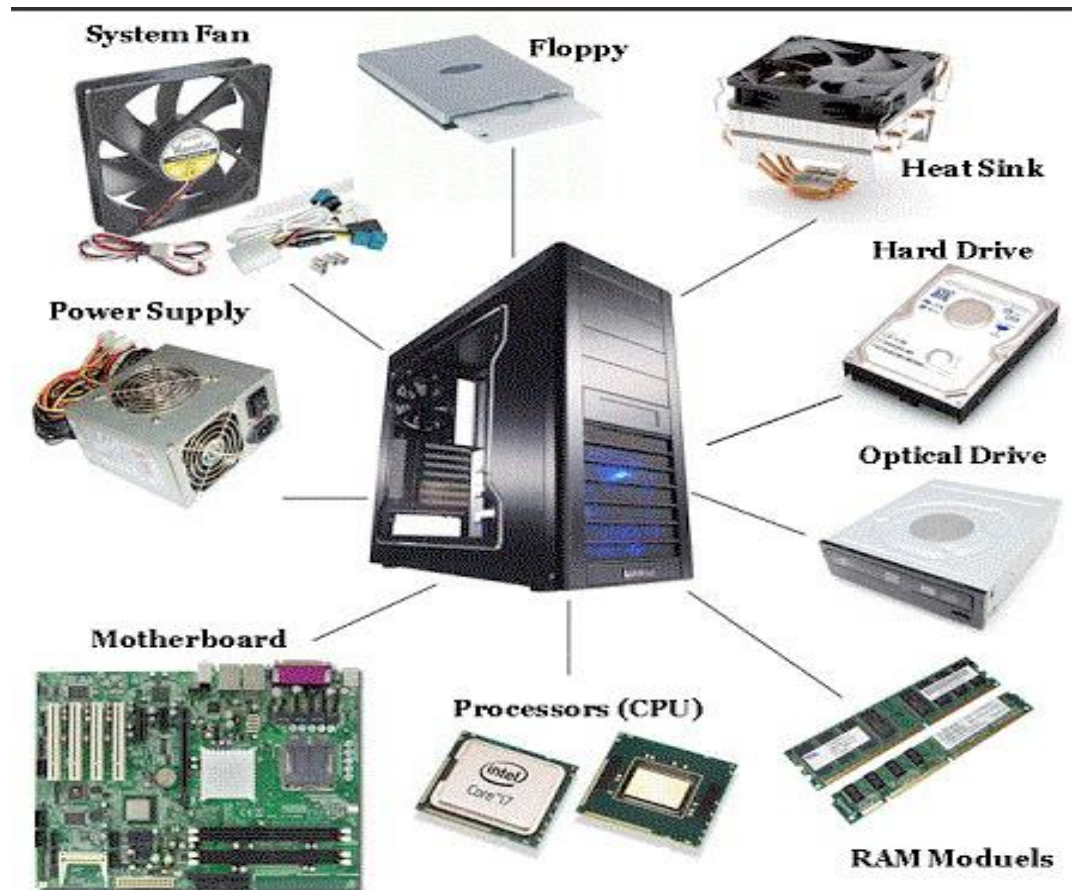
**Main Memory:** lưu trữ các chương trình đang thực thi và các dữ liệu liên quan

**Secondary Storage:** lưu trữ chương trình và các tập tin chứa dữ liệu.

# 1. c) Các thiết bị nội vi, ngoại vi



- ❖ Các thiết bị nội vi thường được hiểu là các thiết bị được chứa đựng bên trong thùng máy: CPU, RAM, HDD, Mainboard,...



# 1. c) Các thiết bị nội vi, ngoại vi



- ❖ Các thiết bị nội ngoại vi chỉ các thiết bị bên ngoài thùng máy được gắn kết với máy tính thông qua các cổng giao tiếp như máy in, máy scan, màn hình, loa...

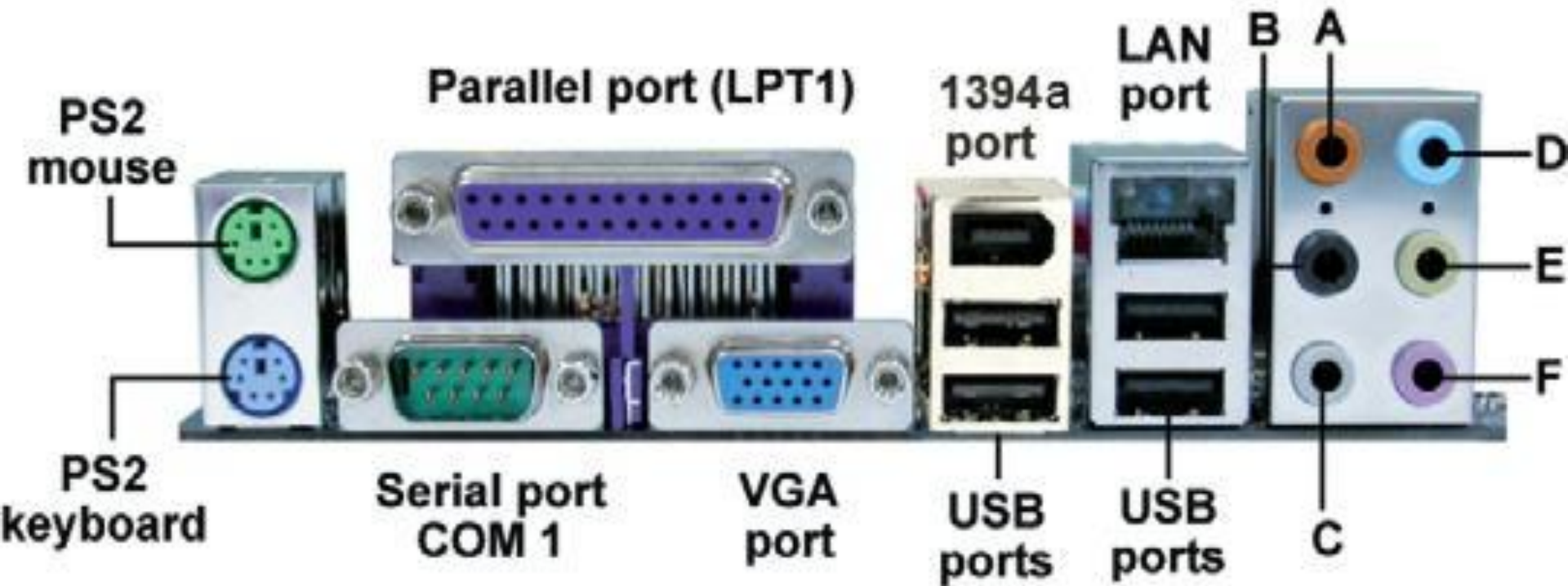




# 1. c) Các thiết bị nội vi, ngoại vi



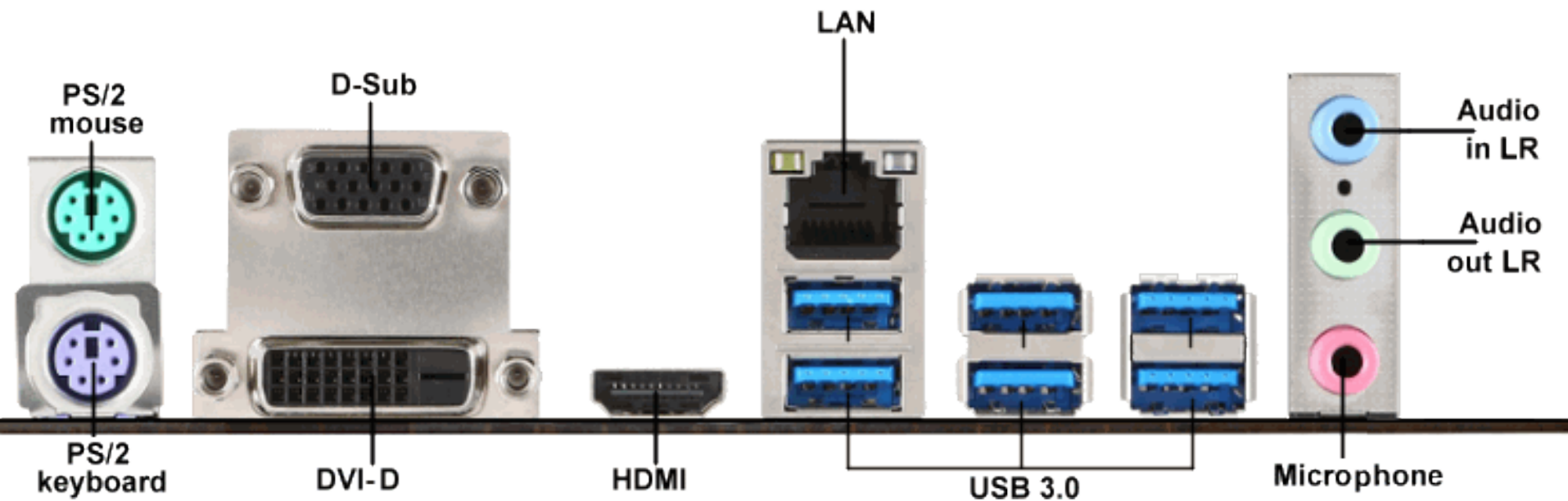
- ❖ Các cổng giao tiếp của máy tính được thiết kế trên bản mạch chính có nhiệm vụ kết nối các thiết bị ngoại vi nhằm tương tác với máy tính.



# 1. c) Các thiết bị nội vi, ngoại vi




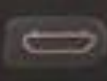





## ❖ Các cổng giao tiếp của máy tính.



## c) Các thiết bị nội vi, ngoại vi



### ❖ Các cổng giao tiếp của máy tính.

USB							
							
Micro-USB AB Type AB	Micro-B Type B	USB Type A	USB Type B	USB 3.0 Type A	USB 3.0 Type B	USB 3.0 Type Micro-B	Mini-USB Type A
Micro USB	Micro USB	USB 2.0	USB 2.0	USB 3.0	USB 3.0	USB 3.0	Mini USB
Loại AB	Loại B	Loại A	Loại B	Loại A	Loại B	Loại Micro B	Loại A

# 1. c) Các thiết bị nội vi, ngoại vi



## ❖ Các thành phần cơ bản của máy tính.

Northbr  
IDE  
DRAM Mem  
20-pin ATX Po  
Connecto





❖ 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, và các dữ liệu hay tài liệu liên quan nhằm tự động thực hiện một số nhiệm vụ hay chức năng hoặc giải quyết một vấn đề cụ thể nào đó.



# 1. d) Phần mềm máy tính

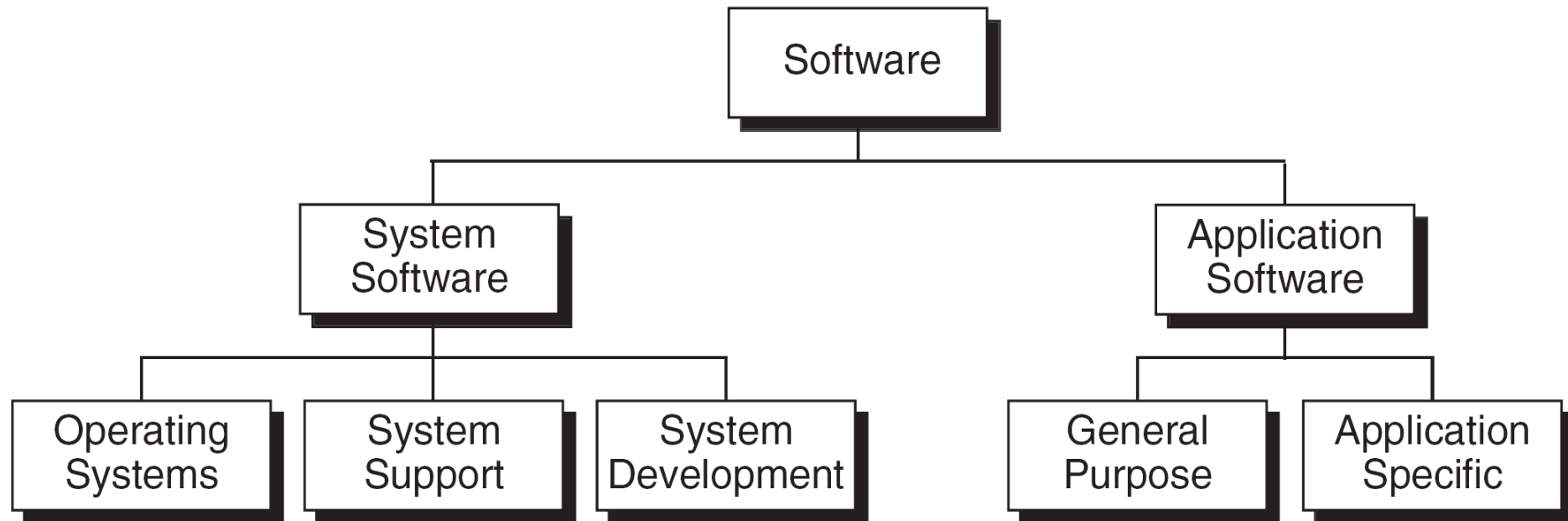


- ❖ Phần mềm thực hiện các chức năng của nó bằng cách gửi các chỉ thị trực tiếp đến phần cứng (hay phần cứng máy tính, Computer Hardware) hoặc bằng cách cung cấp dữ liệu để phục vụ các chương trình hay phần mềm khác.
- ❖ Phần mềm là một khái niệm trừu tượng, nó khác với phần cứng ở chỗ là "phần mềm không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

# 1. d) Phần mềm máy tính



## ❖ Các loại phần mềm.



# 1. d) Phần mềm máy tính



❖ **Phân loại phần mềm máy tính:** phân loại dựa trên tính chất của phần mềm (**phần mềm hệ thống**, **phần mềm ứng dụng**), dạng cung cấp phần mềm (**phần mềm mã mở**, **phần mềm mã nguồn đóng**), hoặc chi phí sử dụng phần mềm (**phần mềm thương mại**, **phần mềm dùng thử**, **phần mềm miễn phí**)



# 1. d) Phần mềm máy tính



❖ **Phần mềm hệ thống (System Software):** bao gồm hệ điều hành (có nhiệm vụ quản lý hoạt động của phần cứng, phần mềm trên máy) và các trình điều khiển thiết bị (driver) như: Microsoft Windows 7, Linux, driver máy in, driver máy quét.





# 1. d) Phần mềm máy tính



❖ **Phần mềm ứng dụng (Application Software):** phục vụ cho một mục đích cụ thể của người dùng như: MS Word, MS Excel, Windows Media Player, Skype, ...



# 1. d) Phần mềm máy tính



❖ **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 ai cũng có thể nghiên cứu, thay đổi và cải tiến phần mềm, ...



# 1. d) Phần mềm máy tính



❖ **Phần mềm nguồn đóng (Closed Source Software):** là phần mềm mà mã nguồn không được công bố. Muốn sử dụng phần mềm nguồn đóng, người dùng phải bỏ chi phí sử dụng. Các hình thức bẻ khóa, phổ biến phần mềm nguồn đóng được xem là bất hợp pháp





# 1. d) Phần mềm máy tính



❖ **Phần mềm thương mại (Commercial Software):** là phần mềm được sản xuất nhằm mục đích thương mại. Người dùng phải bỏ một khoản phí nhất định để sử dụng phần mềm

## COMMERCIAL SOFTWARE



# 1. e) Thông tin được biểu diễn và đo lường



**Thông tin được biểu diễn và đo lường trên máy tính:** bằng hệ đếm nhị phân còn gọi là bit. Có 2 trạng thái **0** hoặc **1** gọi (có 4 hệ số đo).

- ❖ **Hệ nhị phân (0 hoặc 1)**
- ❖ **Hệ thập phân (0-9)**
- ❖ **Hệ bát phân (0-7)**
- ❖ **Hệ thập lục (0-9 và A-F)**



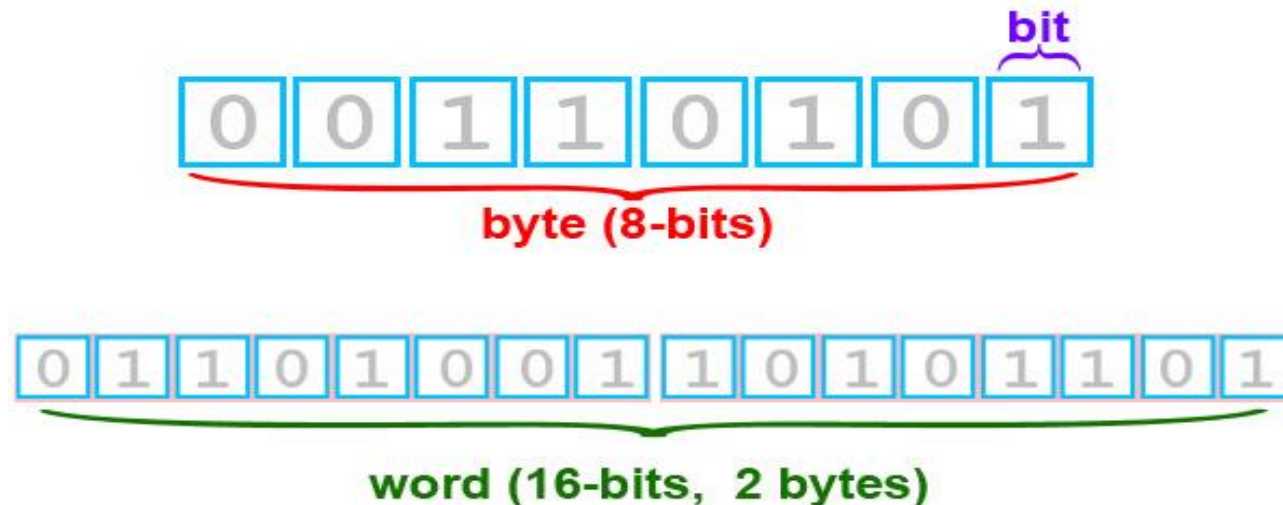
# 1. e) Thông tin được biểu diễn và đo lường



❖ **Hệ nhị phân (0 hoặc 1)**: Để lượng hóa lưu trữ thông tin trên bộ nhớ.

❖ **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. e) Thông tin được biểu diễn và đo lường



❖ Bảng quy đổi  
hệ nhị phân.

Tên gọi	Ký hiệu	Qui đổi
Byte	B	1 B = 8 bit
KiloByte	KB	1 KB = 1024 B = $2^{10}$ B
MegaByte	MB	1 MB = 1024 KB = $2^{10}$ KB
GigaByte	GB	1 GB = 1024 MB = $2^{10}$ MB
TeraByte	TB	1 TB = 1024 GB = $2^{10}$ GB
PetaByte	PB	1 PB = 1024 TB = $2^{10}$ TB
ExaByte	EB	1 EB = 1024 PB = $2^{10}$ PB
ZettaByte	ZB	1 ZB = 1024 EB = $2^{10}$ EB
YottaByte	YB	1 YB = 1024 ZB = $2^{10}$ ZB

# 1. e) Thông tin được biểu diễn và đo lường



## ❖ Các đơn vị đo lường dữ liệu

Table 1: Data Measurement Units

Unit	Abbreviation	Decimal Value	Binary Value	Decimal Size
bit	b	0 or 1	0 or 1	1/8 of a byte
byte	B	8 bits	8 bits	1 byte
kilobyte	KB	1,000 <sup>1</sup> bytes	1,024 <sup>1</sup> bytes	1,000 bytes
megabyte	MB	1,000 <sup>2</sup> bytes	1,024 <sup>2</sup> bytes	1,000,000 bytes
gigabyte	GB	1,000 <sup>3</sup> bytes	1,024 <sup>3</sup> bytes	1,000,000,000 bytes
terabyte	TB	1,000 <sup>4</sup> bytes	1,024 <sup>4</sup> bytes	1,000,000,000,000 bytes
petabyte	PB	1,000 <sup>5</sup> bytes	1,024 <sup>5</sup> bytes	1,000,000,000,000,000 bytes
exabyte	EB	1,000 <sup>6</sup> bytes	1,024 <sup>6</sup> bytes	1,000,000,000,000,000,000 bytes
zettabyte	ZB	1,000 <sup>7</sup> bytes	1,024 <sup>7</sup> bytes	1,000,000,000,000,000,000,000 bytes
yottabyte	YB	1,000 <sup>8</sup> bytes	1,024 <sup>8</sup> bytes	1,000,000,000,000,000,000,000,000 bytes



# 1. e) Thông tin được biểu diễn và đo lường



## ❖ Bảng quy đổi hệ số thập phân sang nhị phân

$$\begin{array}{r} 2 \overline{)156} \\ 2 \overline{)78} \\ 2 \overline{)39} \\ 2 \overline{)19} \\ 2 \overline{)9} \\ 2 \overline{)4} \\ 2 \overline{)2} \\ 2 \overline{)1} \end{array}$$

Remainder:

0  
0  
1  
1  
1  
0  
0  
1



$$156_{10} = \underline{\underline{10011100_2}}$$

# 1. e) Thông tin được biểu diễn và đo lường



❖ Bảng quy đổi hệ số bát phân (cơ số 8) sang nhị phân

Bát phân	Nhị phân
0	000
1	001
2	010
3	011

Bát phân	Nhị phân
4	100
5	101
6	110
7	111

# 1. e) Thông tin được biểu diễn và đo lường



## ❖ Bảng quy đổi hệ thập lục(ơ số 16) sang nhị phân

Thập lục phân	Thập phân	Nhị phân
0	0	0000
1	1	0001
2	2	0010
3	3	0011

Thập lục phân	Thập phân	Nhị phân
4	4	0100
5	5	0101
6	6	0110
7	7	0111

Thập lục phân	Thập phân	Nhị phân
8	8	1000
9	9	1001
A	10	1010
B	11	1011

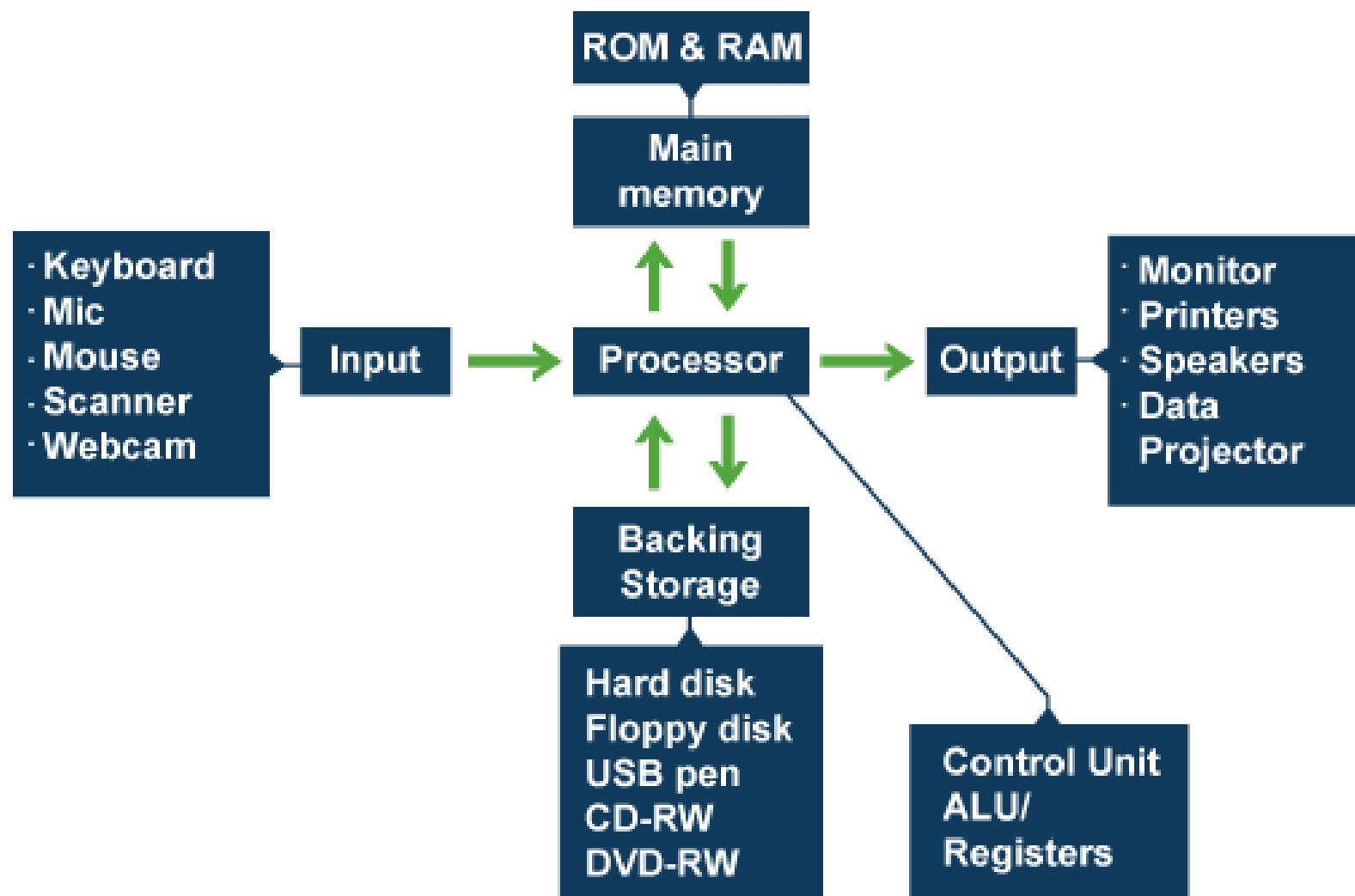
Thập lục phân	Thập phân	Nhị phân
C	12	1100
D	13	1101
E	14	1110
F	15	1111



# 1. f) Máy tính xử lý thông tin



❖ Quá trình xử lý thông tin trên máy tính?



## 2. Các khái niệm cơ bản về lập trình



- a) Lập trình máy tính, lập trình viên
- b) Chương trình máy tính, mã nguồn, mã máy
- c) Ngôn ngữ lập trình.
- d) Chương trình dịch: Trình biên dịch, trình thông dịch,



## 2.a) 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 thảo chương viên điện toán) là người viết ra các chương trình máy tính.





### ❖ The first computer programmer

# Ada Lovelace

**Ada Lovelace** has been called the world's first computer programmer. What she did was write the world's first machine algorithm for an early computing machine that existed only on paper. Of course, someone had to be the first, but Lovelace was a woman, and this was in the 1840s.



Ada Lovelace: The First Computer Programmer | Mental Floss

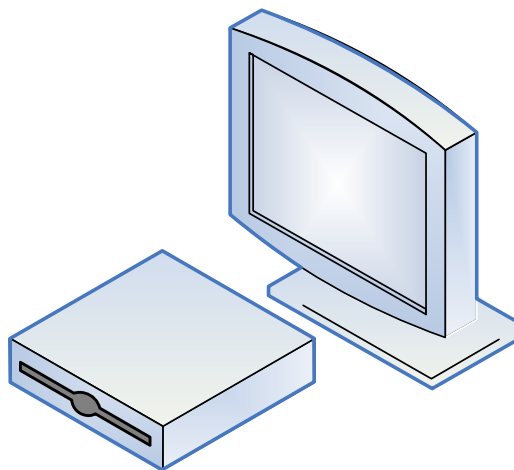
[https://mentalfloss.com](https://mentalfloss.com/article/ada-lovelace-first-computer-programmer) › article › ada-lovelace-first-computer-programmer

## 2. b) Chương trình máy tính, mã nguồn, mã máy

❖ **Chương trình máy tính** là tập hợp các câu lệnh thực hiện một tác vụ cụ thể khi được **máy tính** thực thi. Hầu hết các thiết bị **máy tính** yêu cầu các **chương trình** để hoạt động đúng. Một **chương trình máy tính** thường được viết bởi một **lập trình viên máy tính** bằng ngôn ngữ **lập trình**

**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);
```

## 2. b) Chương trình máy tính, mã nguồn, mã máy

❖ **Chương trình máy tính** là một cách giao tiếp với máy tính và được viết bằng ngôn ngữ lập trình.



Con người giao tiếp với nhau



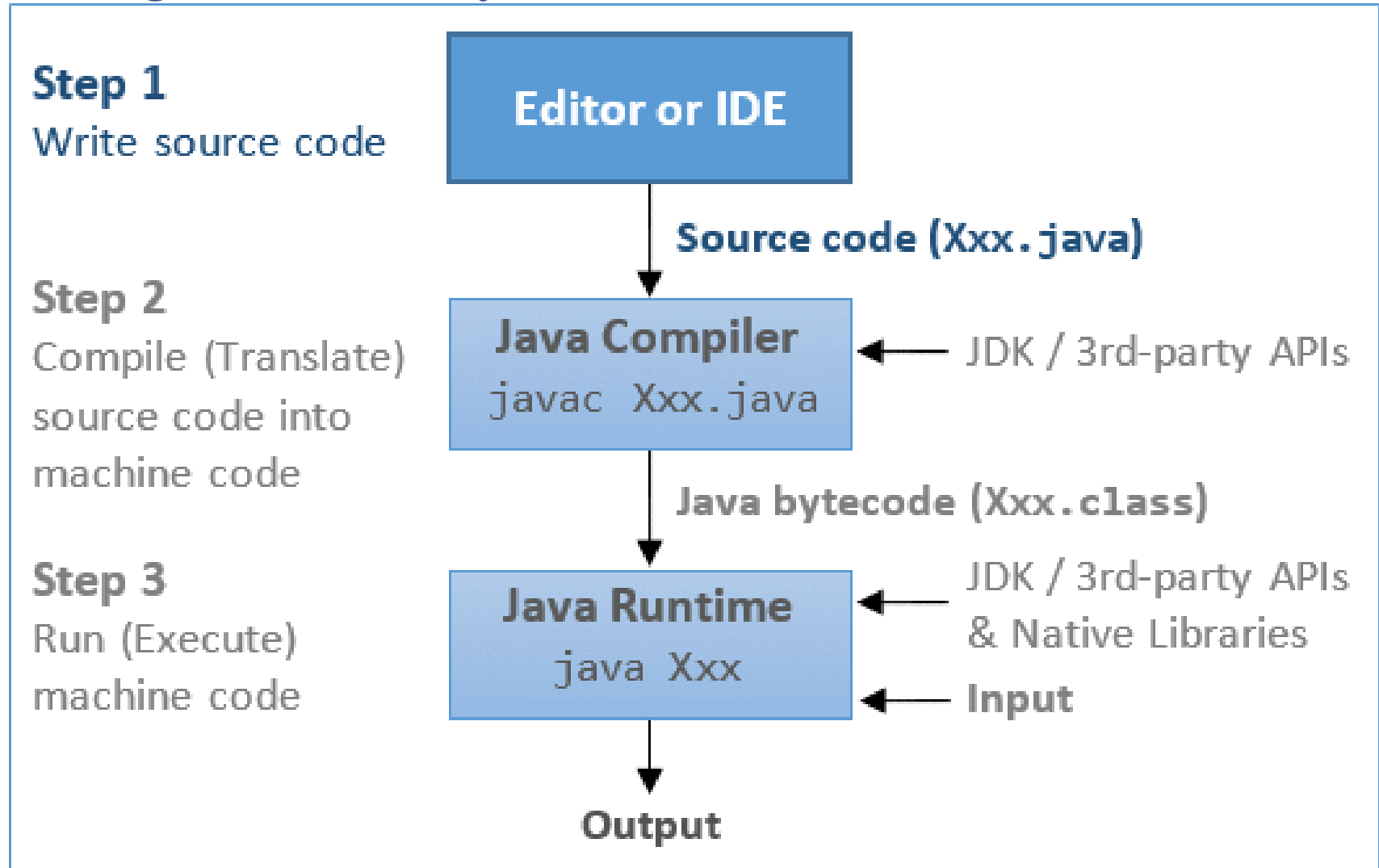
Con người giao tiếp với máy tính

## 2. b) Chương trình máy tính, mã nguồn, mã máy

- ❖ **Mã nguồn** (*source code*) là một **bộ luật** quy định **cách viết code** của một chương trình mà các lập trình viên **bắt buộc phải tuân** theo khi tham gia viết chương trình. (Ví dụ: đặt tên hàm, thuộc tính, khai báo sử dụng biến, chỉ chú trong mã nguồn...)
- ❖ **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** (Run) được một cách trực tiếp. Mã máy được viết sử dụng các số trong hệ nhị phân binary (*chỉ gồm 2 số 0 và 1 hoặc hệ cơ số 16 hexadecimal*)

## 2. b) Chương trình máy tính, mã nguồn, mã máy

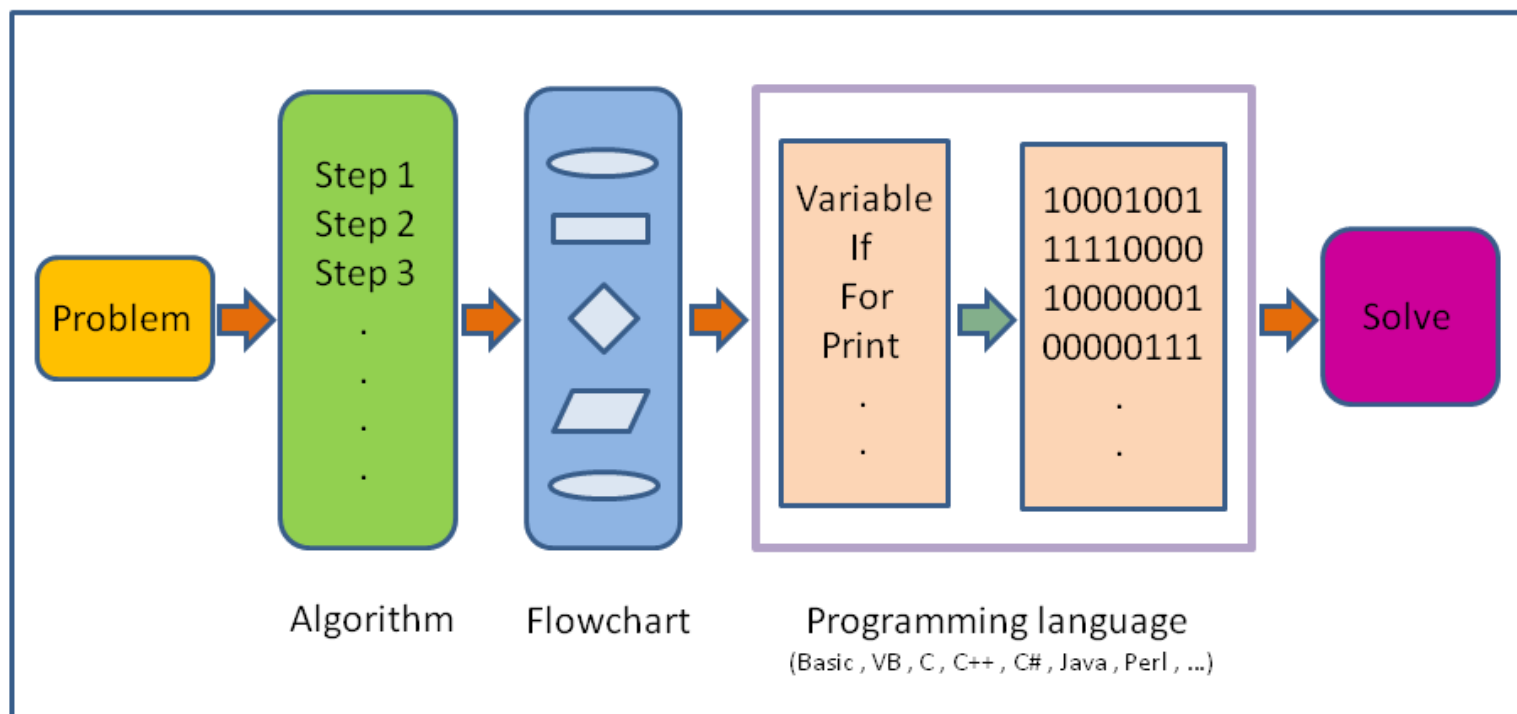
### ❖ Mã nguồn, mã máy:





## 2. c) 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.

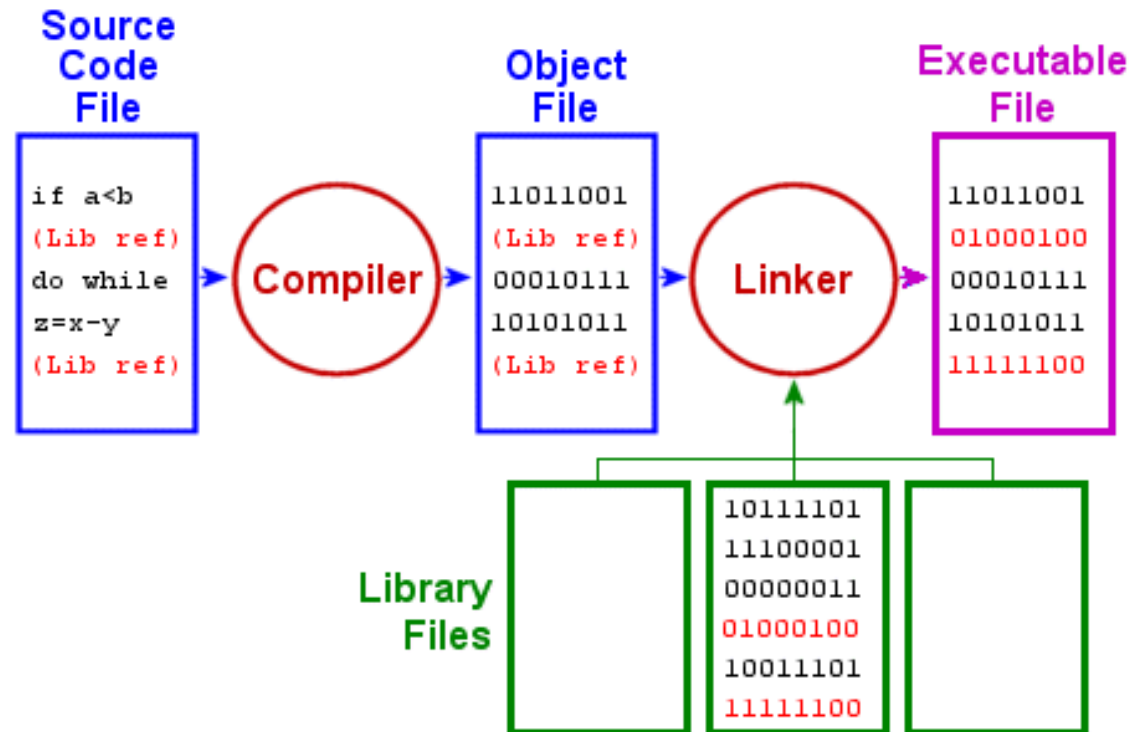


Nguồn: [https://en.wikiversity.org/wiki/Introduction\\_to\\_Programming/About\\_Programming](https://en.wikiversity.org/wiki/Introduction_to_Programming/About_Programming)

## 2. d) Chương trình dịch



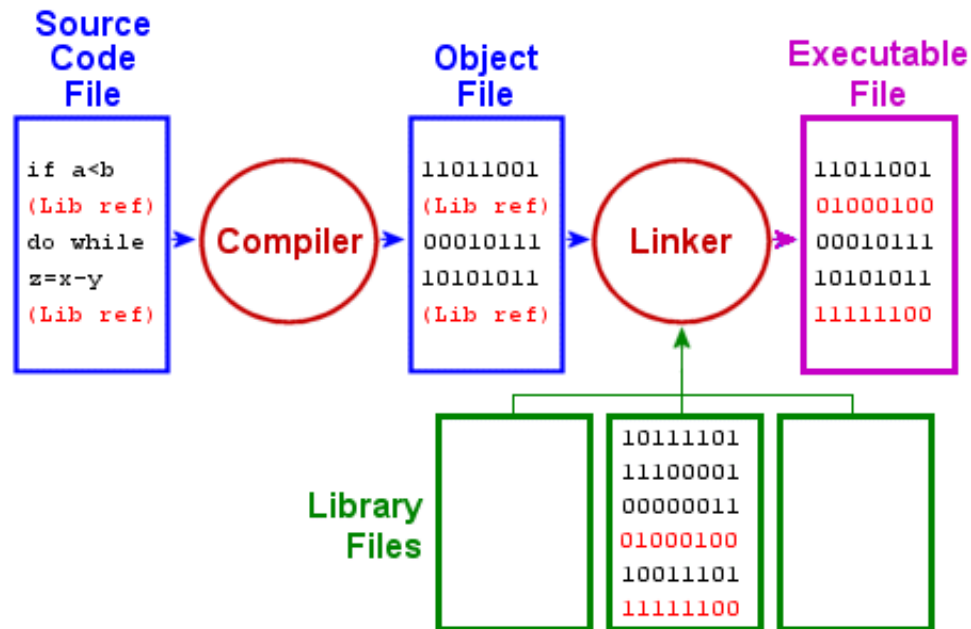
- ❖ Chương trình dịch là chương trình đặc biệt có chức năng **chuyển đổi** chương trình được viết bằng ngôn ngữ lập trình bậc cao **thành chương trình thực hiện được trên máy tính.**



## 2. d) Chương trình dịch



- ❖ **Trình thông dịch (interpreter):** Kiểm tra tính đúng đắn của câu lệnh chuyển đổi thành ngôn ngữ máy và thực thi câu lệnh đó.
- ❖ **Trình biên dịch (compiler):** Duyệt, kiểm tra rồi dịch toàn bộ chương trình để thực thi trên máy hoặc lưu để sử dụng sau.



### 3. Các ngôn ngữ lập trình



- a) Vai trò của NNLT đối với công nghệ lập trình.
- b) Ngôn ngữ lập trình cấp thấp
- c) Ngôn ngữ lập trình cấp cao
- d) Một vài ngôn ngữ lập trình thông dụng



### 3. a) Vai trò của ngôn ngữ lập trình



#### Mô hình phát triển phần mềm



### 3. b) 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*



## 3.b) Ngôn ngữ cấp thấp



❖ Ví dụ:

1		00000000	00000100	000000000000000000
2	01011110	00001100	11000010	000000000000000010
3		11101111	00010110	000000000000000101
4		11101111	10011110	000000000000001011
5	11111000	10101101	11011111	00000000000010010
6		01100010	11011111	00000000000010101
7	11101111	00000010	11111011	00000000000010111
8	11110100	10101101	11011111	00000000000011110
9	00000011	10100010	11011111	00000000000100001
10	11101111	00000010	11111011	00000000000100100
11	01111110	11110100	10101101	
12	11111000	10101110	11000101	00000000000101011
13	00000110	10100010	11111011	00000000000110001
14	11101111	00000010	11111011	00000000000110100
15		01010000	11010100	00000000000111011
16			00000100	00000000000111101

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.

### 3. b) 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*

## 3.b) Ngôn ngữ cấp thấp



❖ Ví dụ:

1	entry	main, ^m<r2>
2	subl2	#12, sp
3	jsb	C\$MAIN_ARGS
4	movab	\$CHAR_STRING_CON
5		
6	pushal	-8(fp)
7	pushal	(r2)
8	calls	#2, SCANF
9	pushal	-12(fp)
10	pushal	3(r2)
11	calls	#2, SCANF
12	mull3	-8(fp), -12(fp), -
13	pusha	6(r2)
14	calls	#2, PRINTF
15	clrl	r0
16	ret	

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**

### 3. c) 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 (Anh ngữ) 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*

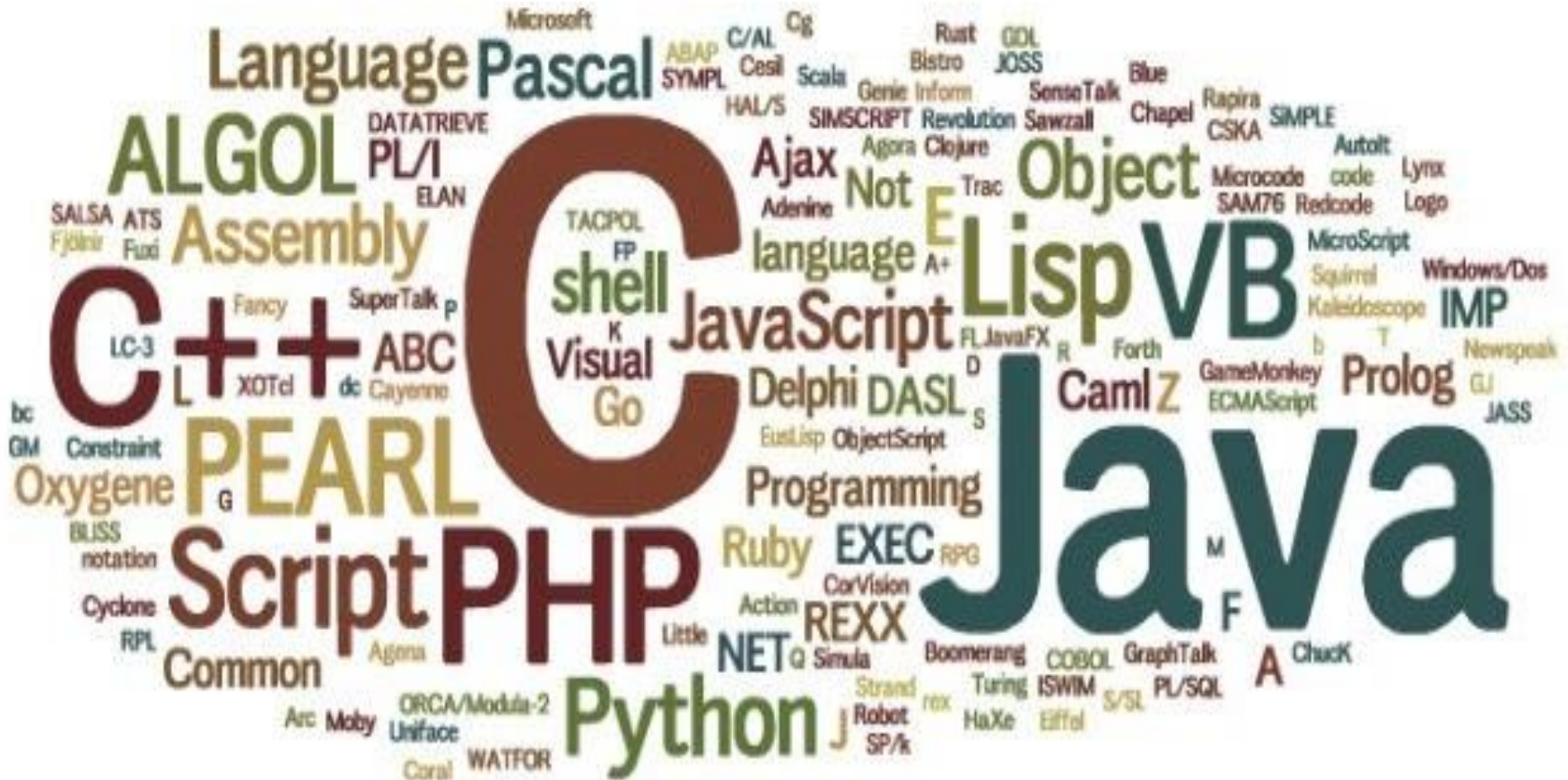
#### High-level Language

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*



## 4. Giới thiệu sơ lược C++



- a) Giới thiệu tổng quan về ngôn ngữ C++
- b) Giới thiệu sơ lược về cấu trúc chương trình
- c) Giới thiệu môi trường, công cụ hỗ trợ việc lập trình
- d) Quy trình tổng quát viết, dịch, chạy thử chương trình





## 4. a) Giới thiệu tổng quan về ngôn ngữ C++



- ❖ C++ là một ngôn ngữ lập trình hỗ trợ lập trình hướng đối tượng, lập trình thủ tục.
- ❖ C++ được coi như là ngôn ngữ bậc trung (middle-level), khi nó kết hợp các đặc điểm và tính năng của ngôn ngữ bậc cao và bậc thấp.
- ❖ 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.
- ❖ C++ là một Superset của C, và bất kỳ chương trình C nào cũng là một chương trình C++.

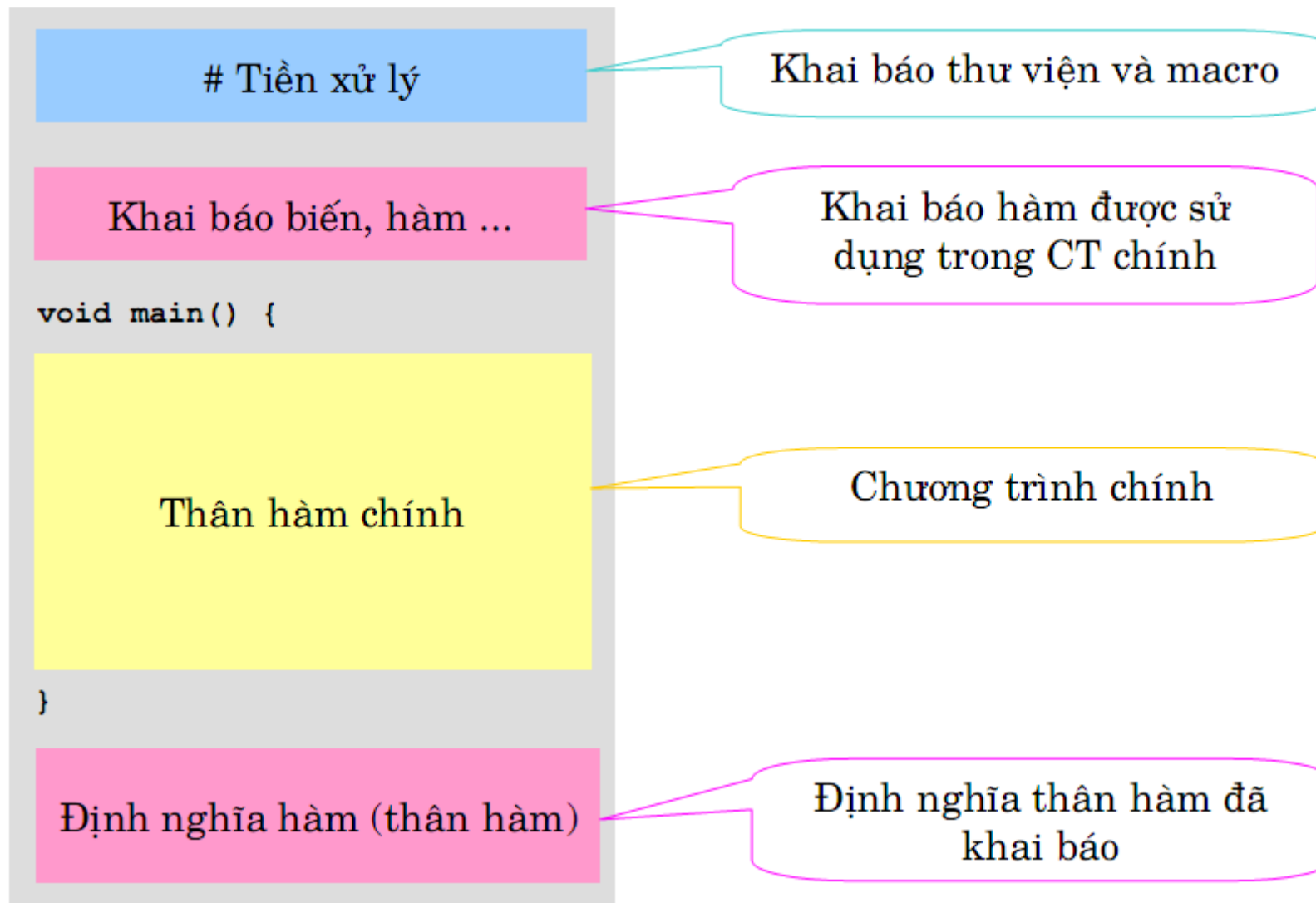
## 4. a) Giới thiệu tổng quan về ngôn ngữ C++



### ❖ Các từ khóa

<b>asm</b>	<b>auto</b>	bool	break
case	<b>catch</b>	char	<b>class</b>
const	<b>const_cast</b>	continue	default
<b>delete</b>	else	extern	do
enum	<b>false</b>	double	<b>explicit</b>
float	<b>dynamic_cast</b>	<b>export</b> for	
<b>friend</b>	goto	if	<b>inline</b>
int	long	<b>mutable</b>	<b>namespace</b>
<b>new</b>	<b>operator</b>	<b>private</b>	<b>protected</b>
<b>public</b>	register	<b>reinterpret_cast</b>	return
short	signed	sizeof	static
<b>static_cast</b>	struct	switch	<b>template</b>
<b>this</b>	<b>throw</b>	<b>true</b>	<b>try</b>
typedef	<b>typeid</b>	<b>typename</b>	union
unsigned	<b>using</b>	<b>virtual</b>	void
volatile	<b>wchar_t</b>	while	

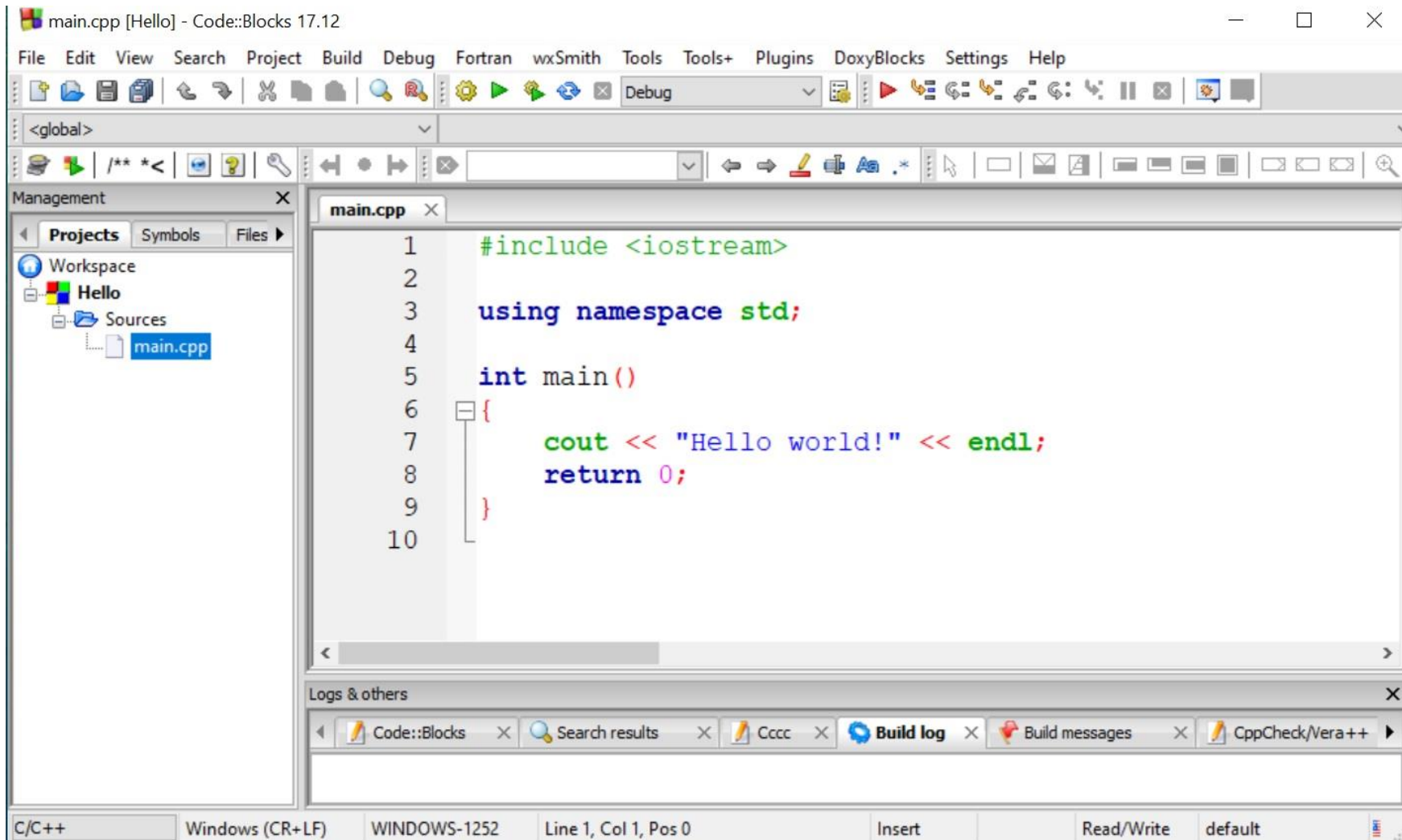
## 4. b) Giới thiệu sơ lược về cấu trúc chương trình



# 4. c) Giới thiệu sơ lược về CodeBlock

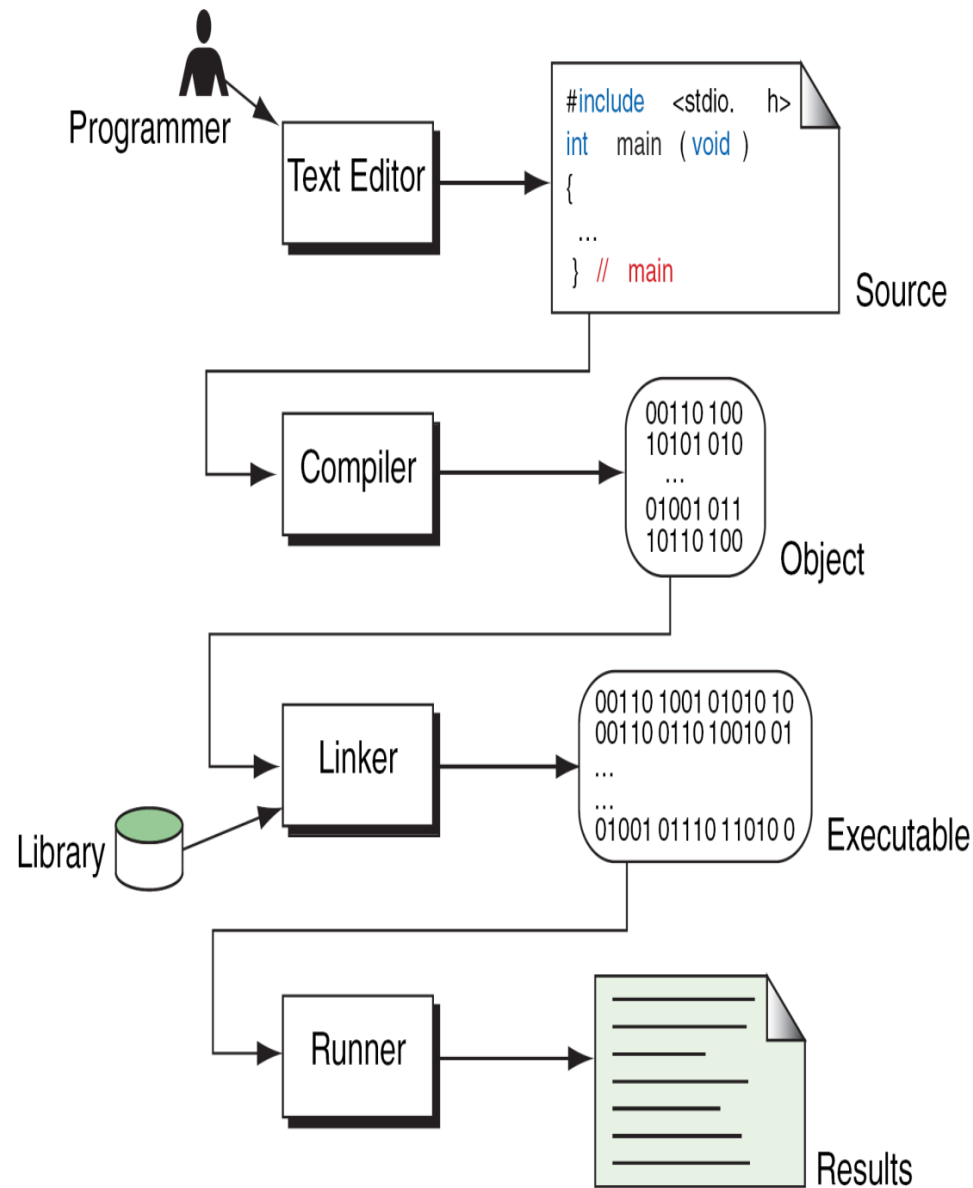


- Tạo project



## 4. d) Qui trình tổng quát viết, dịch, chạy thử chương trình

- **Writing** source code as an C++ file.
  - e.g. “*hello.cpp*” file
- **Preprocessing**
  - Processes the source code for compilation.
- **Compilation**
  - Checks the grammatical rules (syntax)
  - Source code is converted to object code in machine language (e.g. “*hello.obj*” file)
- **Linking**
  - Combines object code and libraries to create an executable (e.g. “*hello.exe*” file).
  - Library: common functions (input, output, math, etc).



## 5. Ví dụ minh họa về chương trình C++ và chạy thử

- 1) Ví dụ 1: Nhập xuất đơn giản như “Hello World” (code: hello.cpp).
- 2) 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” (code: tinhdientich.cpp)
- 3) 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 (code: kiemtrasnt.cpp)





### 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;  
void main()  
{  
    cout << "Hello, world!\n";  
}
```



## 5.b) 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 chương trình tính diện tích hình chu nhật */  
  
#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;  
    // Tính diện tích hình chu nhật  
    int dien_tich = chieu_dai*chieu_rong;  
    // In kết quả ra màn hình  
    cout << "Diện tích HCN = "<< dien_tich;  
}
```

## 5.c) 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

```
/* Chương trình minh họa kiểm tra số nguyên tố */  
  
#include <iostream>  
using namespace std;  
// Hàm kiểm tra số nguyên n có phải là số nguyên tố (true)  
// hay không (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;  
}  
void main()  
{  
    int n;  
    cout<< "Nhập vào số nguyên n = ";  
    cin>>n;  
    if (kiemtralasonguyento(n)== true)  
        cout<< "Số "<<n<<" là số nguyên tố !";  
    else  
        cout<< "Số "<<n<<" không phải là số nguyên tố !";  
}
```

## 6. Một số quy tắc cần nhớ khi viết chương trình



- ❖ Chương trình nên được tách thành nhiều đơn thể (mô-đun), mỗi đơn thể thực hiện một công việc và càng độc lập với nhau.
- ❖ Cách trình bày chương trình càng nhất quán sẽ càng dễ đọc và dễ hiểu (định hướng về phong cách lập trình).
- ❖ Mỗi câu lệnh có thể viết trên một hay nhiều dòng nhưng phải được kết thúc bằng dấu.
- ❖ Quy tắc viết lời giải thích, lời giải thích không có tác dụng với sự làm việc của chương trình trên máy tính, chỉ có tác dụng với người đọc
- ❖ Sử dụng các hàm chuẩn: sử dụng #include.
- ❖ Hàm chính main