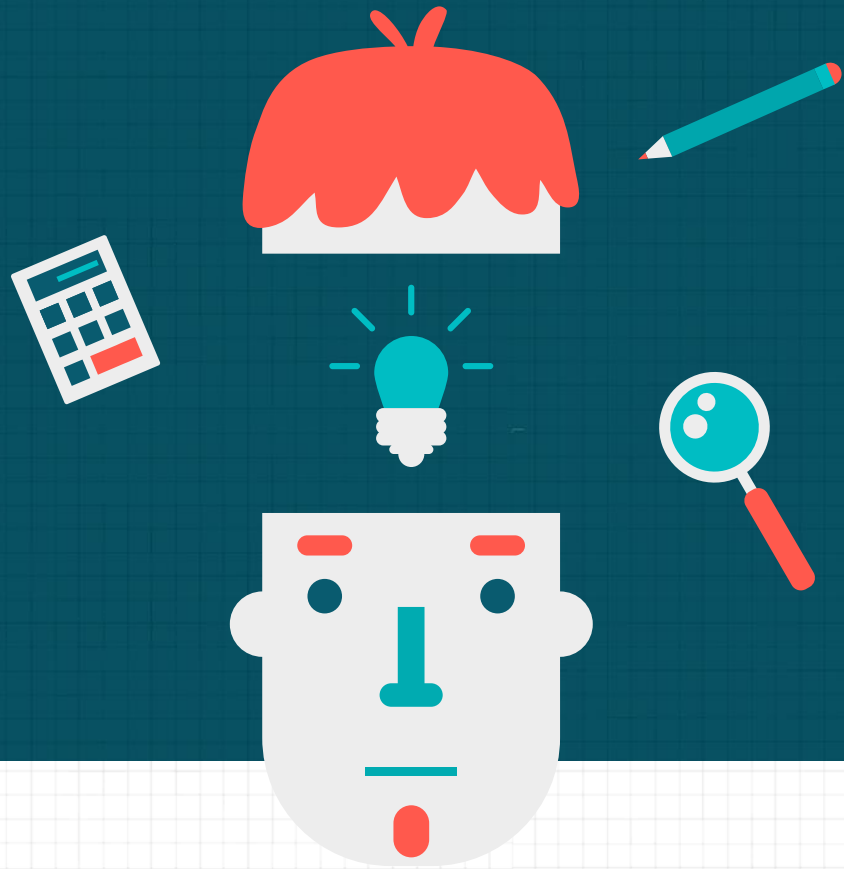


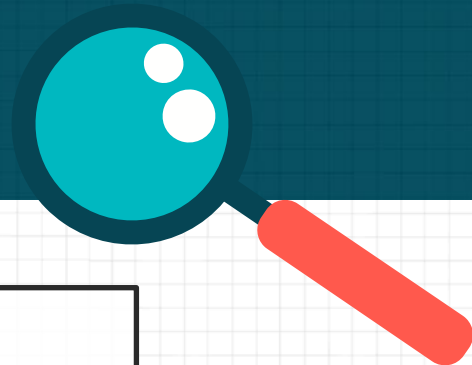
Bài 3

Tập tin

Ths. Phạm Minh Hoàng



Nội dung

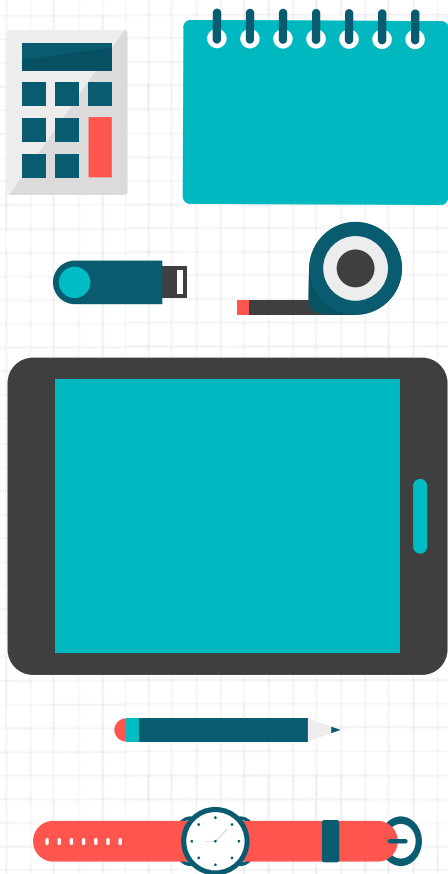


Các khái niệm cơ bản

Xử lý trên tập tin

Thiết kế tập tin

Tham số dòng lệnh



2e

Các thao tác hỗ trợ

Con trỏ file

- Đặt vị trí con trỏ file trong C:

- Cú pháp

```
int fseek(FILE* fp, long int offset, int origin);
```

- Ý nghĩa: Đặt con trỏ file **fp** ở vị trí cách mốc **origin** 1 khoảng **offset**
- Các mốc **origin** đặc biệt
 - **SEEK_SET**: ở đầu file
 - **SEEK_CUR**: vị trí hiện tại
 - **SEEK_END**: ở cuối file

Con trỏ file

- Đặt vị trí con trỏ file trong C++:

- Cú pháp

```
ostream& seekp(streampos pos) ;  
ostream& seekp(streampos off, ios_base::seekdir way) ;  
istream& seekg(streampos pos) ;  
istream& seekg(streampos off, ios_base::seekdir way) ;
```

- Ý nghĩa:
 - Đặt con trỏ file ở vị trí **pos**
 - Đặt con trỏ file ở vị trí cách mốc **way** 1 khoảng **off**
 - Các mốc **way** đặc biệt
 - **ios_base::beg**: ở đầu file
 - **ios_base::cur**: vị trí hiện tại
 - **ios_base::end**: ở cuối file

Xác định vị trí con trỏ file

- Xác định vị trí con trỏ file trong C:

- Cú pháp

```
int ftell(FILE* fp) ;
```

- Ý nghĩa: Trả về vị trí hiện tại của con trỏ file **fp**

Xác định vị trí con trỏ file

- Xác định vị trí con trỏ file trong C++:

- Dành cho ostream

```
streampos tellp();
```

- Dành cho istream

```
streampos tellg();
```

- Ý nghĩa: trả về vị trí hiện tại của stream

Kiểm tra cuối file

- Kiểm tra cuối file trong C:
 - Bằng hằng số **EOF** ($= -1$)

```
while (fgetc(fp) != EOF) ...
```

- Bằng hàm **fEOF**

```
int fEOF(FILE* fp) ;
```

- Trả về 0 nếu chưa đến cuối file
- Trả về số khác 0 nếu đến cuối file

Kiểm tra cuối file

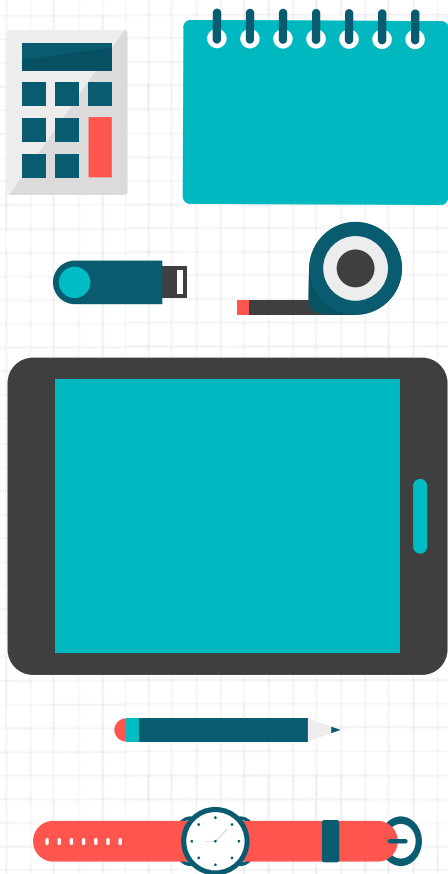
- Kiểm tra cuối file trong C++:
 - Bằng hàm **eof**

```
bool eof() ;
```

- Trả về TRUE nếu đến cuối file
- Trả về FALSE nếu chưa đến cuối file hoặc có lỗi

Bài tập

- Viết hàm tính kích thước của một file bất kỳ
- Đọc lần lượt từng dòng trong file văn bản input
 - Cho biết file có tổng cộng bao nhiêu dòng
 - Xuất từng dòng trên file văn bản output



03

Thiết kế tập tin

Thiết kế tập tin

- Dữ liệu lưu trữ trên file nếu được thiết kế theo cấu trúc thích hợp sẽ có hiệu suất truy xuất cao
- Dữ liệu trên file thường thiết kế theo 2 dạng
 - Dữ liệu dạng văn bản
 - Văn bản phi cấu trúc
 - Văn bản có cấu trúc
 - Dữ liệu dạng nhị phân

Tập tin dạng văn bản

- Văn bản phi cấu trúc
 - Không tổ chức theo ý nghĩa, chức năng
 - Khó xử lý
- Văn bản có cấu trúc
 - Dữ liệu dạng chuỗi và được tổ chức theo quy tắc nhất định
 - Dễ truy xuất, xử lý và thay đổi nội dung
 - Bảo mật kém, tốc độ chậm, kích thước lớn

Tập tin dạng văn bản

- Các dạng văn bản có cấu trúc thường gặp
 - File cấu hình phần mềm (*.INI, *.CFG,...)
 - File *.HTML, *.XML

Tập tin dạng nhị phân

- Dữ liệu được lưu trên file giống cách lưu trên bộ nhớ chính
- Các tính chất
 - Dễ truy xuất và xử lý
 - Tốc độ truy xuất nhanh hơn
 - Tính bảo mật cao hơn
- Chiếm phần lớn số lượng file đang dùng trên máy tính: DOC, XLS, PPT,...

Tập tin dạng nhị phân

- Hai dạng file nhị phân thường gặp
 - File có cấu trúc riêng, tự tổ chức
 - Được thiết kế chuyên biệt cho ứng dụng
 - Cấu trúc file được quy ước ngầm trong ứng dụng
 - File có cấu trúc chuẩn
 - Được công nhận rộng rãi
 - Thường tổ chức dưới dạng file có header

Tập tin dạng nhị phân

- Header là phần dữ liệu thường lưu ở đầu file nhằm cung cấp các thông tin về cấu trúc chi tiết của file
- Ví dụ:
 - Header file ảnh bao gồm
 - Kích thước ảnh W, H
 - Số lượng màu, bảng màu, độ phân giải
 - Header file CSDL
 - Số trường dữ liệu
 - Tên các trường dữ liệu

Tập tin hình ảnh bmp

- Bmp là tập tin dùng để lưu trữ hình ảnh phổ biến trên nhiều môi trường
- Dữ liệu hình ảnh trên bmp được lưu trực tiếp từng điểm ảnh và không cần áp dụng giải thuật nén
- Chất lượng hình ảnh được đảm bảo mặc dù dung lượng cao



Tập tin hình ảnh bmp

- Cấu trúc file bmp gồm 4 phần chính

Thành phần	Kích thước	Ý nghĩa
Header	14 bytes	Thông tin tổng quát của file
DIB	40 bytes	Thông tin chi tiết của file
Color Table	Tùy biến	Thông tin bảng màu
Pixel Data	Tùy biến	Dữ liệu điểm ảnh

Tập tin hình ảnh bmp

- Cấu trúc phần header (14 bytes)

Thành phần	Kích thước	Ý nghĩa
Signature	2 bytes	Chữ ký file. Trong Win là "BM"
FileSize	4 bytes	Kích thước file
Reserved	4 bytes	Phần dành riêng
DataOffset	4 bytes	Địa chỉ phần bắt đầu lưu dữ liệu điểm ảnh

Tập tin hình ảnh bmp

- Cấu trúc phần DIB (40 bytes)

Thành phần	Kích thước	Ý nghĩa
DIB Size	4 bytes	Kích thước phần DIB = 40 bytes
Width	4 bytes	Số pixel theo chiều rộng
Height	4 bytes	Số pixel theo chiều dài
Planes	2 bytes	Số lớp màu (=1)
Bits per pixel	2 bytes	Số bit trong 1 pixel, gồm các giá trị 1, 4, 8, 16, 24, 32

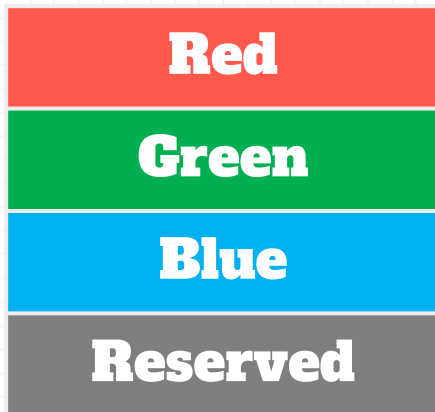
Tập tin hình ảnh bmp

- Cấu trúc phần DIB (40 bytes)

Thành phần	Kích thước	Ý nghĩa
Compression	4 bytes	Cách nén ảnh (0: không nén, 1: 8bit RLE, 2: 4bit RLE)
ImageSize	4 bytes	Kích thước phần dữ liệu điểm ảnh
XpixelsPerM	4 bytes	Độ phân giải theo phương ngang
YpixelsPerM	4 bytes	Độ phân giải theo phương đứng
Colors Used	4 bytes	Số màu trong bảng màu
Important Color	4 bytes	Số màu quan trọng

Tập tin hình ảnh bmp

- Bảng màu
 - Số phần tử bảng màu tương đương số lượng màu là 2^n trong đó n là số bit của 1 pixel
 - 1 phần tử của bảng là 1 block 4 byte gồm



Tập tin hình ảnh bmp

- Pixel data

Pixel format Width Padding bytes

Image Data PixelFormat [x,y]					
Pixel[0,h-1]	Pixel[1,h-1]	Pixel[2,h-1]	...	Pixel[w-1,h-1]	Padding
Pixel[0,h-2]	Pixel[1,h-2]	Pixel[2,h-2]	...	Pixel[w-1,h-2]	Padding
⋮					
Pixel[0,9]	Pixel[1,9]	Pixel[2,9]	...	Pixel[w-1,9]	Padding
Pixel[0,8]	Pixel[1,8]	Pixel[2,8]	...	Pixel[w-1,8]	Padding
Pixel[0,7]	Pixel[1,7]	Pixel[2,7]	...	Pixel[w-1,7]	Padding
Pixel[0,6]	Pixel[1,6]	Pixel[2,6]	...	Pixel[w-1,6]	Padding
Pixel[0,5]	Pixel[1,5]	Pixel[2,5]	...	Pixel[w-1,5]	Padding
Pixel[0,4]	Pixel[1,4]	Pixel[2,4]	...	Pixel[w-1,4]	Padding
Pixel[0,3]	Pixel[1,3]	Pixel[2,3]	...	Pixel[w-1,3]	Padding
Pixel[0,2]	Pixel[1,2]	Pixel[2,2]	...	Pixel[w-1,2]	Padding
Pixel[0,1]	Pixel[1,1]	Pixel[2,1]	...	Pixel[w-1,1]	Padding
Pixel[0,0]	Pixel[1,0]	Pixel[2,0]	...	Pixel[w-1,0]	Padding

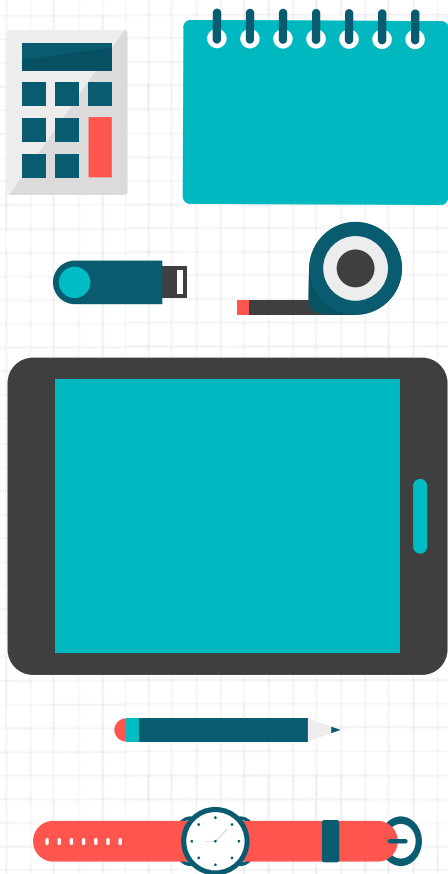
Height

Tập tin hình ảnh bmp

- Pixel format
 - 1 bit per pixel (1bpp): chỉ gồm 2 màu đen & trắng
 - 4 bit per pixel (4bpp): gồm 16 màu phân biệt
 - 8 bit per pixel (8bpp): gồm 256 màu phân biệt, 1 pixel bao gồm 1 byte
 - 16 bit per pixel (16bpp): 1 pixel bao gồm 2 byte
 - 24 bit per pixel (24bpp): 1 pixel bao gồm 3 bytes, mỗi byte tương ứng các màu B, G, R
 - 32 bit per pixel (32bpp): 1 pixel bao gồm 4 bytes, mỗi byte tương ứng các màu A, B, G, R

Tập tin hình ảnh bmp

- **Padding bytes:** Là số byte được thêm vào ở mỗi dòng sao cho tổng số byte trên mỗi dòng là bội số của 4
- **Pixel data** lưu dưới dạng mảng 2 chiều các pixel, trong đó dòng đầu tiên tính từ góc dưới bên trái của ảnh

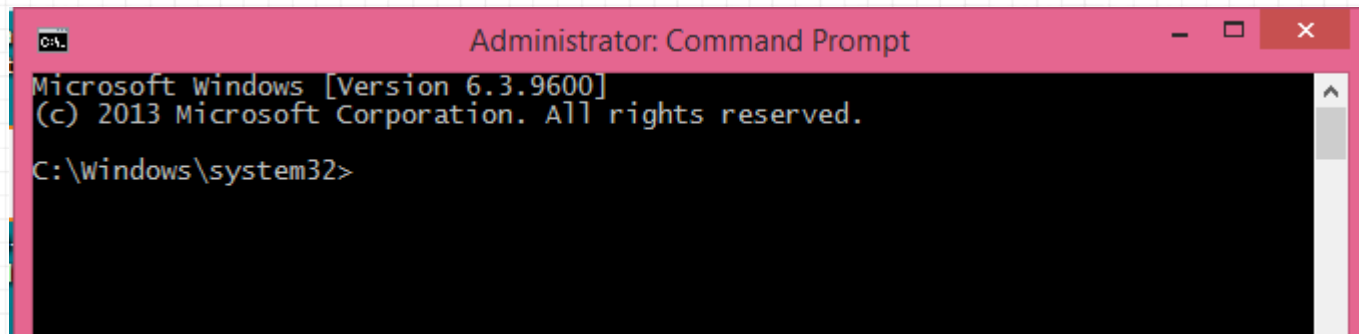


04

Tham số dòng lệnh

Tham số dòng lệnh

- Làm tham số đi kèm với tên chương trình trong giao diện dòng lệnh của hệ điều hành
 - CMD trong Windows
 - Terminal trong Linux/MacOS

A screenshot of the Windows Command Prompt window. The title bar is pink and reads "Administrator: Command Prompt". The window has standard minimize, maximize, and close buttons. The main area is black with white text. It shows the version "Microsoft Windows [Version 6.3.9600]", copyright information "(c) 2013 Microsoft Corporation. All rights reserved.", and the current directory "C:\Windows\system32>".

```
C:\Windows\system32>
```

Tham số dòng lệnh

- Các tham số của hàm main
 - argc: số lượng tham số được truyền vào bao gồm tên chương trình
 - argv: mảng chuỗi các tham số trong đó argv[0] luôn là tên file thực thi của chương trình

```
int main(int argc, char* argv[]) {  
  
}
```

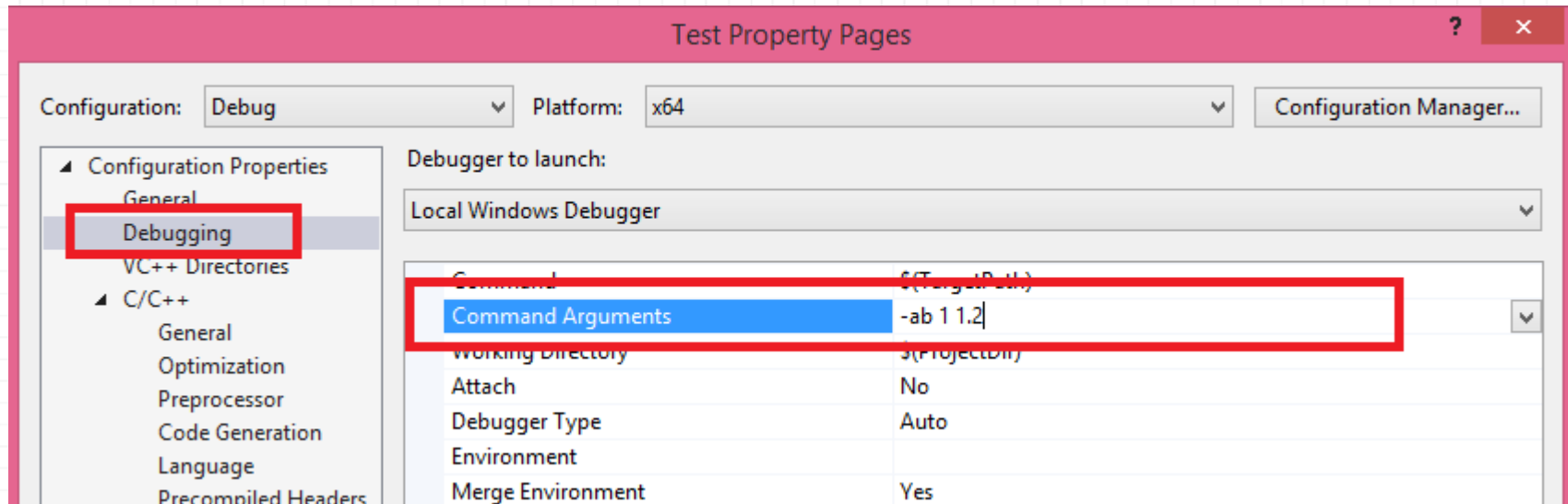
Tham số dòng lệnh

- Đặc điểm các tham số argv
 - argv[0]: tên chương trình/đường dẫn đến file thực thi
 - argv[1]: tham số đầu tiên của chương trình

```
int main(int argc, char* argv[]) {  
  
}
```

Tham số dòng lệnh

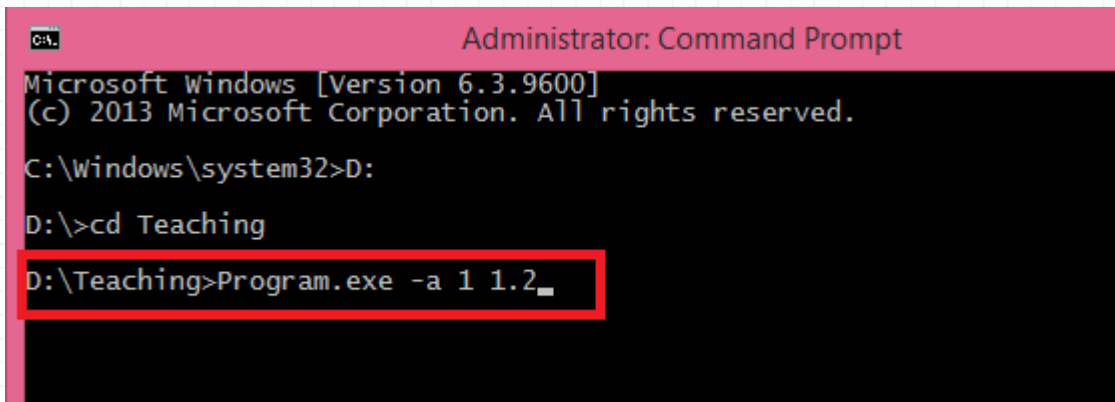
- Truyền tham số dòng lệnh chế độ debug
- Project → Properties → Configuration Properties → Debugging → Command Arguments



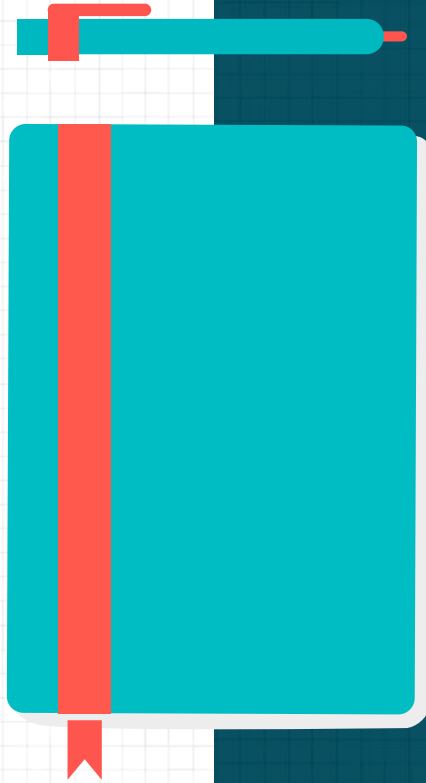
Tham số dòng lệnh

- Gọi chương trình bằng tham số dòng lệnh
- Thực hiện trong cửa sổ command line
- Đi đến thư mục chứa file thực thi và gọi thực hiện bằng tham số dòng lệnh có dạng

`<Tên chương trình> <tham số 1> <tham số 2> ...`



```
Administrator: Command Prompt
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
C:\Windows\system32>D:
D:\>cd Teaching
D:\Teaching>Program.exe -a 1 1.2_
```

The End!

Do you have any questions?