Sử dụng keyword const

- Mục đích sử dụng của const
- Các cách truyền con trỏ vào hàm

Mục đích sử dụng

- Mục đích sử dụng của keyword const là báo với trình biên dịch rằng một biến cụ thể nào đó không được phép thay đổi giá trị
- Ta thấy trong hằng số có chứa keyword này, hằng số không thể thay đổi giá trị
- Ta cũng thấy rằng các tham số có keyword này trong thành phần cấu thành của nó thì không thể thay đổi giá trị trong hàm
- Kể cả con trỏ truyền tham chiếu có keyword này cũng không thể thay đổi giá trị của biến gốc

Các cách truyền con trỏ vào hàm

Có tổng cộng 4 cách truyền con trỏ vào hàm:

- Con trỏ có thể thay đổi trỏ đến dữ liệu có thể thay đổi
- Con trỏ không thể thay đổi trỏ đến dữ liệu có thể thay đổi
- Con trỏ có thể thay đổi trỏ đến dữ liệu không thể thay đổi
- Con trỏ không thể thay đổi trỏ đến dữ liệu không thể thay đổi

Cách 1

Con trỏ có thể thay đổi trỏ đến dữ liệu có thể thay đổi

- Đây là trường hợp truy cập dữ liệu được chấp thuận ở level cao nhất
- Dữ liệu tại vị trí con trỏ trỏ tới có thể được phân giải và thay đổi tùy ý
- Con trỏ này cũng có thể thay đổi để trỏ đến một địa chỉ khác
- Ví dụ hàm sau:

```
// non-constant pointer to non-constant data
void toUpperCase(char *a) {
	while (*a != '\0') {
		*a = toupper(*a);
		a++;
	}
}
```

Cách 2

Con trỏ có thể thay đổi trỏ đến dữ liệu không thể thay đổi

- Trường hợp này con trỏ có thể được thay đổi để trỏ đến các địa chỉ khác nhưng giá trị của các phần tử tại các vị trí này không thể thay đổi
- Hữu ích trong trường hợp sử dụng con trỏ để duyệt phần tử mảng khi tìm kiếm, hiển thị danh sách phần tử
- Ví dụ:

```
void showArray(const char* a) {
	puts("String data is: ");
	for (; *a != '\0'; a++) {
		printf("%c", *a);
	}
}
```

Cách 2

- Nếu ta cố tình thay đổi giá trị 1 phần tử sẽ bị lỗi ngay lập tức:

```
void showArray(const char* a) {
      puts("String data is: ");
      for (; *a != '\0'; a++) {
            printf("%c", *a);
      }
       a[0] = ‘H’; // error!
}
```

Cách 3

Con trỏ không thể thay đổi trỏ đến dữ liệu có thể thay đổi

- Con trỏ lúc này chỉ có thể trỏ đến một vị trí cụ thể
- Giá trị tại vị trí con trỏ trỏ tới có thể được thay đổi tùy ý
- Đây là trường hợp mặc định cho tên mảng, một con trỏ không thể thay đổi và trỏ đến vị trí phần tử đầu tiên trong mảng
- Ta sử dụng trường hợp này khi muốn cho phép dữ liệu thay đổi: các chức năng cập nhật, sắp xếp mảng

Cách 3

Con trỏ không thể thay đổi trỏ đến dữ liệu có thể thay đổi

- Nếu con trỏ khai báo với const thì nó phải được khởi tạo khi định nghĩa
- Tham số là const pointer thì nó sẽ được khởi tạo khi hàm được gọi với đối số truyền vào
- Đoạn chương trình sau lỗi vì cố gắng thay đổi giá trị của một const pointer:

```
int a = 120;
int* const aPtr = &a;
*aPtr = -200; // ok
int b = 500;
aPtr = &b; // error
```

Cách 4

Con trỏ không thể thay đổi trỏ đến dữ liệu không thể thay đổi

- Trường hợp này là trường hợp nghiêm ngặt nhất
- Con trỏ không thể thay đổi để trỏ đến địa chỉ khác
- Dữ liệu tại nơi con trỏ trỏ tới không thể thay đổi
- Hữu ích khi duyệt mảng để tìm kiếm, hiển thị và lấy giá trị ra sử dụng
- Ví dụ:

```
void showArray(const int* const a, const size_t size) {
	for (size_t i = 0; i < size; i++)
	{
		printf("%5d", a[i]);
	}
}
```

Tiếp theo

Con trỏ hàm