Con trỏ

- **Khái niệm, mục đích sử dụng**
- **Cú pháp tổng quát**
- **Khởi tạo con trỏ**
- **Toán tử &, ***
- **Ví dụ minh họa**

Khái niệm & mục đích sử dụng

- Con trỏ là biến mà giá trị chứa trong bản thân nó là địa chỉ của các biến khác
- Mục đích sử dụng: do tính đa năng của con trỏ nên nó được sử dụng rộng rãi trong truyền tham chiếu, truyền mảng vào hàm, cấp phát bộ nhớ động, tạo các cấu trúc dữ liệu động như danh sách liên kết, cây nhị phân, stack, queue…
- Đây là phần khó với nhiều người nhưng sau khi học bài này các bạn sẽ thấy nó đơn giản!

Cú pháp tổng quát

- Cú pháp tổng quát khai báo con trỏ:
  *kiểu *tên_con_trỏ;*
- Trong đó:
  - Kiểu là bất kì kiểu dữ liệu hợp lệ nào trong C, kể cả void
  - Trước tên con trỏ luôn có dấu *
  - Tên con trỏ đặt như tên biến bình thường them đuôi Ptr
  - Kết thúc khai báo bằng chấm phẩy ;
- Ví dụ:

```
int* aPtr;
char* namePtr;
float* fPtr;
```

Cú pháp tổng quát

```
int* aPtr;
char* namePtr;
float* fPtr;
```

- Với aPtr ta đọc là: *aPtr là một con trỏ trỏ đến int* hoặc *aPtr trỏ đến một đối tượng của kiểu int*
- Ta chỉ nên khai báo một biến trên một dòng để tránh nhầm lẫn
- Ví dụ sau chỉ có biến đầu tiên là con trỏ:
  `int* aPtr, bPtr;`
- Để biến thứ hai cũng là con trỏ thì cú pháp là:
  `int* aPtr, *bPtr;`

Khởi tạo

- Khi khai báo biến cũng như con trỏ, ta nên rèn thói quen khởi tạo giá trị cho nó luôn
- Nếu một con trỏ chưa sử dụng ngay ta thường khởi tạo cho nó bằng NULL
- NULL là một giá trị với ý nghĩa là không xác định, vô định. Một con trỏ trỏ đến NULL là một con trỏ vô định không trỏ đến đâu cả.

Khởi tạo

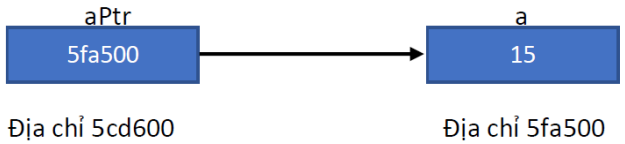
- Bạn có thể khởi tạo con trỏ bằng 0, điều này tương đương NULL nhưng về mặt ngữ nghĩa và tránh sự nhầm lẫn giữa biến thông thường và con trỏ nên ta dùng NULL thay vì 0
- Ví dụ khởi tạo:

```
int* aPtr = NULL;
float *bPtr = NULL;
```

Toán tử &

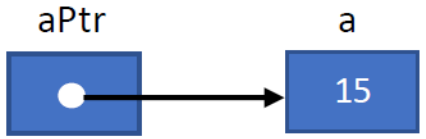
- Toán tử & gọi là toán tử địa chỉ, phân biệt với && - tứ toán tử logic
- Khi muốn khởi tạo hoặc gán địa chỉ của một biến cho con trỏ, ta dùng toán tử &
- Ví dụ:

```
int a = 100;
int *aPtr = &a;
```



- Hoặc ví dụ khác:

```
int a = 15; // bien a
int* aPtr; // bien con tro
aPtr = &a; // gan dia chi cua a cho aPtr
```



Toán tử *

- Toán tử * đặt trước con trỏ khi sử dụng con trỏ gọi là toán tử phân giải địa chỉ
- Toán tử này giúp ta lấy được giá trị tại địa chỉ mà con trỏ đang trỏ tới
- Phân biệt sự khác nhau giữa cú pháp khai báo con trỏ: int *a và cú pháp khi sử dụng con trỏ: *a = 100. Nếu trước * có kiểu dữ liệu thì đó là khai báo con trỏ. Còn trước * không có kiểu thì đó là cú pháp sử dụng con trỏ
- Ví dụ:

```
// ham doi cho hai phan tu
void swap(int* a, int* b)
{
	int t = *a;
	*a = *b;
	*b = t;
}
```

Nâng cao

- Cho aPtr = &a;
- Toán tử & * có thứ tự ưu tiên thực hiện từ phải sang trái, tức là:
- Nếu ta viết &*aPtr tương đương với &(*aPtr) tương đương &(a), kết quả cho địa chỉ của biến a
- Nếu ta viết *&aPtr tương đương *(&aPtr) tức giá trị tại địa chỉ của aPtr, mà aPtr chứa địa chỉ của a do đó ta nhận được địa chỉ của biến a
- Hai cách viết cho cùng kết quả là địa chỉ của biến a hay giá trị của aPtr!

Tiếp theo

Con trỏ và truyền tham chiếu