

컴퓨터공학 All in One

C/C++ 문법, 자료구조 및 심화 프로젝트 (나동빈)
제 9강 - 포인터

학습 목표

포인터

- 1) 포인터의 개념과 활용 방법을 학습합니다.
- 2) 포인터를 사용했을 때의 강력한 기능을 이해합니다.

포인터

포인터의 개념

- 1) 지금까지의 변수는 그 자체로 자신의 자료형에 맞는 값을 저장합니다.
- 2) 포인터(Pointer) 변수는 특이한 변수로, 메모리 주소를 저장합니다.

포인터

포인터의 개념

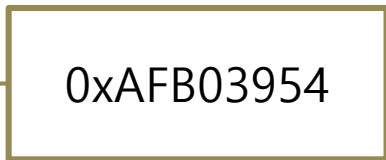
- 1) 포인터는 특정한 변수 자체가 존재하는 메모리 주소의 값을 가집니다.
- 2) 따라서 다음 예제에서는 기존에 a를 이용해서도 5라는 값을 찾을 수 있지만 포인터 변수인 b를 이용해서도 5라는 값을 찾을 수 있습니다.

```
int a = 5;
```

```
int *b = &a;
```



주소: 0xAFB03954



주소: 0xCA29839F



포인터

포인터의 개념

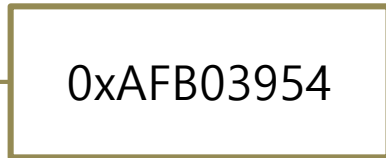
- 1) `int *b = &a;`처럼 '선언할 때' 쓰는 *는 포인터 변수임을 알려주기 위한 목적을 가집니다.
- 2) 이후에 `*b` 라고 쓰게 되면, 이것은 포인터 변수 `b`가 가리키는 주소의 값을 의미합니다.
- 3) 다시 말해서 아래 예제에서 5라는 값 자체가 됩니다.

```
int a = 5;
```



주소: 0xAFB03954

```
int *b = &a;
```



주소: 0xCA29839F



포인터

포인터 관련 연산자

주소 연산자(&)

변수 앞에 붙어서 변수의 메모리 시작 주소 값을 구합니다.

포인터(*)

포인터 변수를 선언할 때 사용합니다.

간접 참조 연산자(*)

선언된 포인터 변수가 가리키는 변수를 구합니다.

포인터

포인터의 개념

- 1) 따라서 선언할 때와 선언 이후의 포인터 연산자(*)는 생긴 것만 같지, 기능은 다릅니다.

```
#include <stdio.h>

int main(void) {
    int a = 5;
    int *b = &a;
    printf("%d\n", *b);
    system("pause");
}
```

포인터

포인터의 개념

- 1) 실제로 `int a = 5;`와 같이 변수를 할당하면 메모리 주소상에서는 다음과 같이 기록됩니다.
- 2) `int`형은 4바이트를 차지하므로 메모리 주소를 1바이트 씩 표현할 때 4칸을 차지합니다.

355	356	357	358	359	360	361	362	363	364
...	...	값	값	값	값

포인터

배열 각 원소의 주소 값 출력하기

```
#include <stdio.h>

int main(void) {
    int a[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
    int i;
    for (i = 0; i < 10; i++) {
        printf("%d\n", &a[i]);
    }
    system("pause");
}
```

포인터

포인터의 강력한 기능

- 1) 포인터는 컴퓨터 시스템의 특정한 메모리에 바로 접근할 수 있습니다.
- 2) 따라서 기존에 존재하던 중요한 메모리 영역에 접근하지 않도록 해야 합니다.
- 3) 다음과 같은 코드는 굉장히 위험한 코드입니다.

```
int *a = 0x33484735;  
*a = 0;
```

포인터

포인터의 강력한 기능

- 1) 포인터는 단지 특정한 메모리 주소를 담고 있는 변수일 뿐입니다.
- 2) 따라서 포인터는 다중으로 사용할 수 있습니다.

```
#include <stdio.h>

int main(void) {
    int a = 5;
    int *b = &a;
    int **c = &b;
    printf("%d\n", **c);
    system("pause");
}
```

포인터

포인터의 강력한 기능

- 1) 배열과 포인터는 사실 동일합니다.
- 2) 배열을 선언한 이후에는 그 이름 자체가 포인터 변수와 동일합니다.

포인터

포인터의 강력한 기능

- 1) 배열과 포인터는 사실 내부적으로 거의 동일합니다.
- 2) 배열을 선언한 이후에는 그 이름 자체를 포인터 변수처럼 쓸 수 있습니다.

```
#include <stdio.h>

int main(void) {
    int a[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
    int *b = a;
    printf("%d\n", b[2]);
    system("pause");
}
```

배운 내용 정리하기

포인터

- 1) 포인터는 특정한 변수가 메모리 상에 존재하는 위치 주소를 저장합니다.
- 2) 포인터는 특정한 메모리 주소에 바로 접근할 수 있으므로 조심스럽게 사용해야 합니다.