

# 컴퓨터공학 All in One

---

C/C++ 문법, 자료구조 및 심화 프로젝트 (나동빈)  
제 12강 - 컴퓨터가 변수를 처리하는 방법

# 학습 목표

## 컴퓨터가 변수를 처리하는 방법

- 1) C언어에서 다양한 변수를 처리하는 방법을 공부합니다.
- 2) 지역 변수, 전역 변수, 레지스터 변수 등을 학습합니다.
- 3) 특정한 함수에 값을 전달하거나 주소를 전달하는 방법을 이해합니다.

# 컴퓨터가 변수를 처리하는 방법

## 프로그램 메모리 주소

- 1) 컴퓨터에서 프로그램이 실행되기 위해서는 프로그램이 메모리에 적재(Load)되어야 합니다.
- 2) 당연히 프로그램의 크기를 충당할 수 있을 만큼의 메모리 공간이 있어야 합니다.
- 3) 일반적인 컴퓨터의 운영체제는 메모리 공간을 네 가지로 구분하여 관리합니다.

코드 영역	데이터 영역	힙 영역	스택 영역
-------	--------	------	-------

소스코드

전역 변수  
정적 변수

동적 할당 변수

지역 변수  
매개변수

# 컴퓨터가 변수를 처리하는 방법

## 전역변수

- 1) 전역변수(Global Variable)란 프로그램의 어디서든 접근 가능한 변수를 말합니다.
- 2) main 함수가 실행되기도 전에 프로그램의 시작과 동시에 메모리에 할당됩니다.
- 3) 프로그램의 크기가 커질 수록 전역 변수로 인해 프로그램이 복잡해질 수 있습니다.
- 4) 메모리의 데이터(Data) 영역에 적재됩니다.

# 컴퓨터가 변수를 처리하는 방법

## 전역변수

```
#include <stdio.h>

int a = 5;

int changeValue() {
    a = 10;
}

int main(void) {
    printf("%d\n", a);
    changeValue();
    printf("%d\n", a);
    system("pause");
    return 0;
}
```

# 컴퓨터가 변수를 처리하는 방법

## 지역변수

- 1) 지역변수(Local Variable)란 프로그램에서 특정한 블록(Block)에서만 접근할 수 있는 변수를 말합니다.
- 2) 함수가 실행될 때마다 메모리에 할당되어 함수가 종료되면 메모리에서 해제됩니다.
- 3) 메모리의 스택(Stack) 영역에 기록됩니다.

# 컴퓨터가 변수를 처리하는 방법

## 지역변수

```
#include <stdio.h>

int main(void) {
    int a = 7;
    if (1) {
        int a = 5;
    }
    printf("%d", a);
    system("pause");
    return 0;
}
```

# 컴퓨터가 변수를 처리하는 방법

## 정적변수

- 1) 정적 변수(Static Variable)란 특정한 블록에서만 접근할 수 있는 변수입니다.
- 2) 프로그램이 실행될 때 메모리에 할당되어 프로그램이 종료되면 메모리에서 해제됩니다.
- 3) 메모리의 데이터(Data) 영역에 적재됩니다.



# 컴퓨터가 변수를 처리하는 방법

## 정적변수

```
#include <stdio.h>

void process() {
    static int a = 5;
    a = a + 1;
    printf("%d\n", a);
}

int main(void) {
    process();
    process();
    process();
    system("pause");
    return 0;
}
```

# 컴퓨터가 변수를 처리하는 방법

## 레지스터 변수

- 1) 레지스터 변수(Register Variable)란 메인 메모리 대신 CPU의 레지스터를 사용하는 변수입니다.
- 2) 레지스터는 매우 한정되어 있으므로 실제로 레지스터에서 처리될 지는 장담할 수 없습니다.

# 컴퓨터가 변수를 처리하는 방법

## 레지스터 변수

```
#include <stdio.h>

int main(void) {
    register int a = 10, i;
    for (i = 0; i < a; i++) {
        printf("%d ", i);
    }
    system("pause");
    return 0;
}
```

# 컴퓨터가 변수를 처리하는 방법

## 함수의 매개변수가 처리될 때

- 1) 함수를 호출할 때 함수에 필요한 데이터를 매개변수로 전달합니다.
- 2) 전달 방식은 ① <값에 의한 전달> 방식과 ② <참조에 의한 전달> 방식이 있습니다.
- 3) 값에 의한 전달 방식은 단지 값을 전달하므로 함수 내에서 변수가 새롭게 생성됩니다.
- 4) 참조에 의한 전달 방식은 주소를 전달하므로 원래의 변수 자체에 접근할 수 있습니다.

# 컴퓨터가 변수를 처리하는 방법

## 값에 의한 전달 방식(ADD 함수)

- 1) ADD 함수로 두 개의 값을 넣으면 새롭게 두 변수가 메모리 내에 할당되어 처리됩니다.
- 2) 따라서 원래 변수의 값에는 영향을 미치지 못합니다.

```
#include <stdio.h>

void add(int a, int b) {
    a = a + b;
}

int main(void) {
    int a = 7;
    add(a, 10);
    printf("%d\n", a);
    system("pause");
    return 0;
}
```

# 컴퓨터가 변수를 처리하는 방법

## 참조에 의한 전달 방식(ADD 함수)

- 1) 참조에 의한 전달 방식은 함수의 매개변수로 값을 전달하는 것이 아니라 변수의 주소를 전달합니다.
- 2) 이러한 방식은 이용하면 원래 변수의 값에 접근하여 값을 변경할 수 있습니다.

```
#include <stdio.h>
```

```
void add(int *a) {  
    *a = (*a) + 10;  
}
```

```
int main(void) {  
    int a = 7;  
    add(&a);  
    printf("%d\n", a);  
    system("pause");  
    return 0;  
}
```

# 컴퓨터가 변수를 처리하는 방법

## 참조에 의한 전달 방식(ADD 함수)

- 1) 참조에 의한 전달 방식은 단지 매개변수로 '포인터(Pointer)' 변수를 보낼 뿐 딱히 특별한 게 아닙니다.

# 배운 내용 정리하기

## 컴퓨터가 변수를 처리하는 방법

- 1) C언어에서는 전역 변수, 지역 변수 등의 다양한 종류의 변수가 사용됩니다.
- 2) 함수에 데이터를 전달하는 방법은 값을 전달하는 방식과 주소를 전달하는 방식 두 가지가 있습니다.