

26장. 접근 범위와 존속 기간

01_ 변수와 함수의 접근 범위와 존속 기간

지역 변수와 전역 변수의 비교(1)

- 지역 변수와 전역 변수를 정의하고 사용하는 예

```
int global = 10;           // 전역 변수의 정의

void MyFunction();

int main()
{
    int local = 20;       // 지역 변수의 정의

    global = 100;        // 전역 변수에 접근 - 성공

    local = 200;         // 같은 함수의 지역 변수에 접근 - 성공

    MyFunction();       // 함수 호출

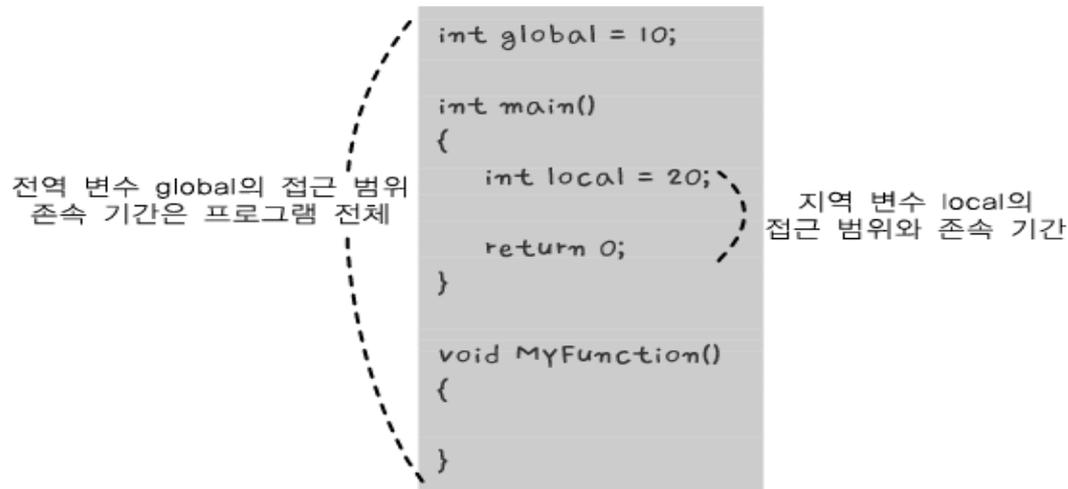
    return 0;
}

void MyFunction()
{
    global = 100;        // 전역 변수에 접근 - 성공

    local = 200;        // 다른 함수의 지역 변수에 접근 - 실패!!
}
```

지역 변수와 전역 변수의 비교(2)

- 지역 변수와 전역 변수의 접근 범위와 존속 기간



	접근 범위	존속 기간
지역 변수	지역 변수를 정의한 함수에서만 접근할 수 있음.	지역 변수를 정의하는 코드를 실행하는 시점에서 생성. 함수를 종료할 때 소멸한다. 객체의 경우는 해당하는 시점에서 생성자와 소멸자를 호출한다.
전역 변수	전역 변수를 정의한 파일에 있는 모든 함수에서 접근할 수 있음. (주의: 다른 파일에서도 접근할 수 있는 방법이 존재한다)	프로그램이 시작할 때 생성. 프로그램이 종료할 때 소멸한다. 객체의 경우는 해당하는 시점에서 생성자와 소멸자를 호출한다.

extern

- 다른 소스 파일에 정의된 전역 변수를 사용하는 예

```
// Example.cpp
// 다른 파일에 있는 전역 변수에
// 접근 하기위한 준비
extern int ga;

int main()
{
    // 다른 파일에 있는 전역 변수에 접근
    ga = 200;          // 성공

    return 0;
}
```

```
// A.cpp
// 전역 변수
int ga = 100;
```

블록 안에서 정의한 변수(1)

- 다른 소스 파일에 정의된 전역 변수를 사용하는 예

```
int main()
{
    // 블록을 만든다.
    {
        // 블록 안에서 변수 정의
        int var_in_block = 10;

        // 변수에 접근
        var_in_block = 100; // 성공
    }

    // 변수에 접근
    var_in_block = 1000;      // 실패

    return 0;
}
```

블록 안에서 정의한 변수(2)

- 블록 안에서 정의한 변수의 접근 범위와 존속 기간

```
int main()
{
    {
        int var_in_block = 10;
    }

    return 0;
}
```

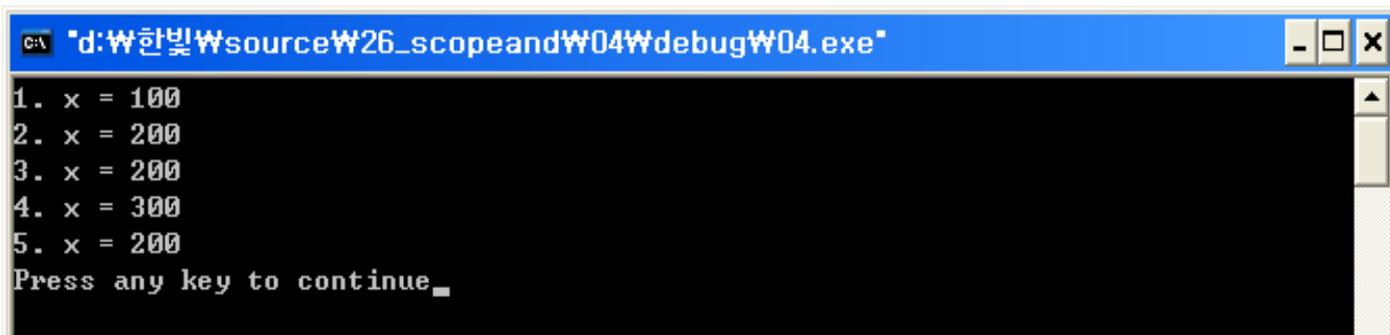
변수 var_in_block의
접근 범위와 존속 기간

다른 영역에 정의한 변수의 이름 중복

- 각각 다른 영역에 같은 이름의 변수를 정의하는 예

```
int x = 100;
int main()
{
    cout << "1. x = " << x << "\n";
    int x = 200;
    cout << "2. x = " << x << "\n";
    {
        cout << "3. x = " << x << "\n";
        int x = 300;
        cout << "4. x = " << x << "\n";
    }
    cout << "5. x = " << x << "\n";
    return 0;
}
```

- 실행 결과



```
C:\> "d:\한빛\source\26_scopeand\04\debug\04.exe"
1. x = 100
2. x = 200
3. x = 200
4. x = 300
5. x = 200
Press any key to continue
```

for 명령 안에서 정의한 변수

- for 명령 안에서 정의한 변수의 존속 기간과 접근 범위는 for 명령의 블록 안쪽으로 제한된다.
- 그러므로, 다음과 같이 for 명령 바깥쪽에서 같은 이름의 변수를 정의하는 것이 가능하다. (표준)

```
for (int i = 0; i < 10; i++)  
{  
}  
  
int i = 100;           // 성공
```

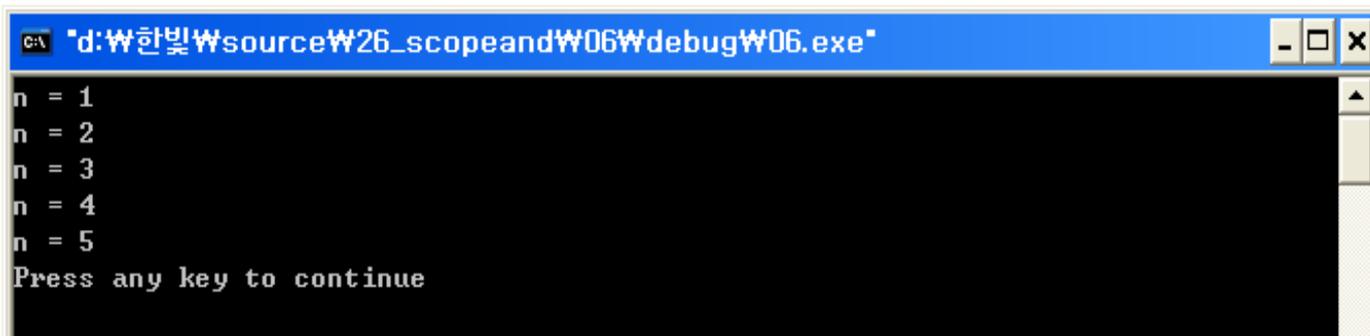
- 그러나, VC++ 6.0 같이 오래된 컴파일러는 for 명령 바깥쪽에서도 변수를 접근할 수 있도록 구현되어 있기 때문에, 위와 같은 코드는 컴파일 오류를 발생시킨다.
- VC++.NET의 경우에는 기본적으로 두 가지 방식 모두를 허용하며, 컴파일 옵션으로 한 가지 방식을 강요할 수도 있다.

static으로 정의한 지역 변수

- static인 지역 변수는 프로그램이 종료될 때까지 살아있게 된다.

```
void Sub()  
{  
    static int n = 0;  
    cout << "n = " << ++n << "\n";  
}  
  
int main()  
{  
    Sub();  
    Sub();  
    Sub();  
    Sub();  
    Sub();  
    Sub();  
    return 0;  
}
```

- 실행 결과



```
C:\> "d:\한빛\source\W26_scopeand\W06\debug\W06.exe"  
n = 1  
n = 2  
n = 3  
n = 4  
n = 5  
Press any key to continue
```

static으로 정의한 전역 변수

- static인 전역 변수는 다른 파일에서 접근 할 수 없게 된다.

```
// Example.cpp
// 다른 파일에 있는 전역 변수에
// 접근 하기위한 준비
extern int ga;

int main()
{
    // 다른 파일에 있는 전역 변수에 접근
    ga = 200;          // 실패 !!

    return 0;
}
```

```
// A.cpp
// 정적 전역 변수
static int ga = 100;
```

register 변수

- 레지스터 변수는 메모리가 아닌 레지스터에 위치한다.

```
// 레지스터 변수의 정의
register int i = 0;

while ( i < 1000000 )
{
    // 중간 코드 생략
    ++i;
}
```

- 레지스터 변수는 읽고, 쓰는 속도가 빠르므로 자주 접근되는 변수를 레지스터 변수로 만들면 좋다.
- 레지스터 변수를 만든다고 해도 상황에 따라서 메모리에 위치할 수도 있다.
- 요즘의 컴파일러들은 자주 접근되는 변수들을 자동적으로 레지스터 변수로 만들어주기 때문에 일반적으로는 직접 사용할 일이 많지 않다.

다른 파일에서 정의한 함수에 접근하기

- 변수와 마찬가지로 extern 키워드를 사용할 수 있다

```
// Example.cpp
extern void Func();

int main()
{
    Func(); // 성공

    return 0;
}
```

```
// A.cpp
void Func()
{
}
```

- extern 키워드를 생략해도 동일한 의미를 갖는다.

static 으로 정의한 함수

- static으로 정의한 전역 변수와 마찬가지로 다른 파일에서 접근 할 수 없게 된다.

```
// Example.cpp
extern void Func();

int main()
{
    Func();    // 실패!!

    return 0;
}
```

```
// A.cpp
static void Func()
{
}
```

C 언어로 작성한 함수 사용하기

- C언어로 작성한 함수를 사용하는 예

```
// Example.cpp
extern "C" void Func();

int main()
{
    Func();    // 성공

    return 0;
}
```

```
// A.c : c언어로 작성한 파일 (확장자가 .c 다)
void Func()
{
}
```

- 같은 원형의 함수이더라도 C언어와 C++언어에서 해석하는 방식에 차이가 있으므로 extern "C"를 붙여주지 않으면 컴파일 오류가 발생한다.

함수의 원형	C 언어	C++ 언어
void Func()	_Func	?Func@@YAXXZ

Q&A

