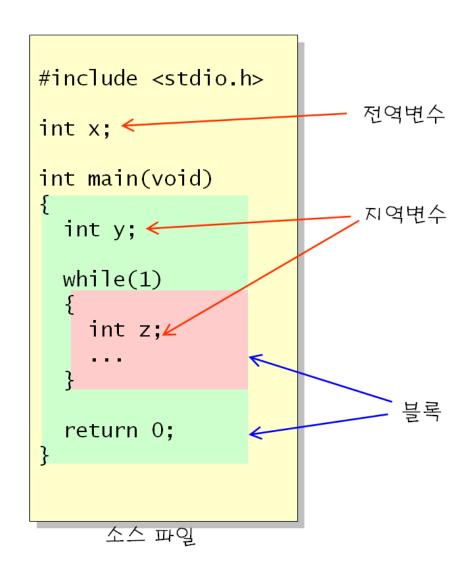
변수의 범위 scope of variables

2023 국민대학교 소프트웨어학부

변수의 범위

- 범위(scope): 변수가 접근되고 사용되는 범위
- 변수의 범위는 변수가 선언되는 위치에 따라서 결정된다.
- 범위의 종류
 - 전역 변수: 함수의 외부에서 선언,
 - 지역 변수: 블록 안에서 선언

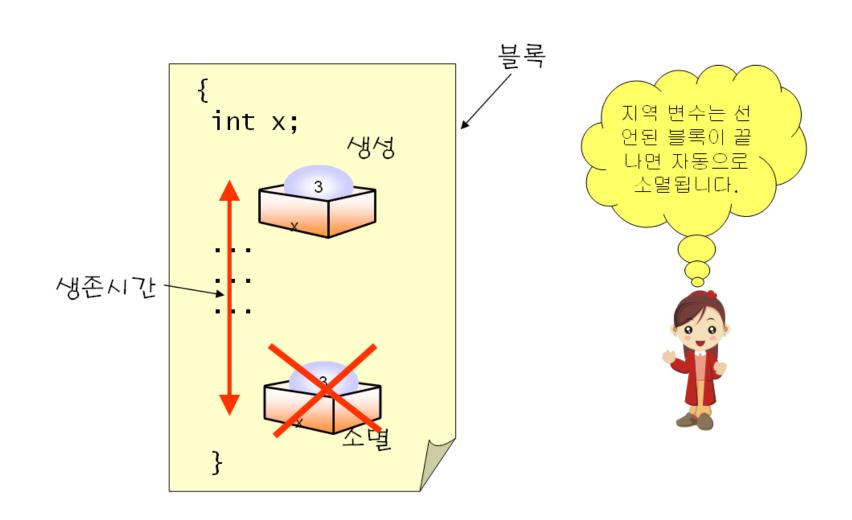


지역 변수의 scope

• 지역 변수(local variable): 블록 안에서 선언되는 변수

```
void sub1()
                                 지역 변수 x
의 범위
 int x;
  int y;
                                 지역 변수 y
                                   의 범위
                                 범위가 다르
면 이름이 같
void sub2()
                                   아도 됨
 int x;
```

지역 변수의 생존 기간 lifetime



퀴즈 문제 중에서...

```
for (int i = 0; i<4; i++){
    if (i%2) continue;
    else    cout << i << endl;
}</pre>
```

```
13. for (int i = 0; i<4; i++){
    if (i%2) continue;
    cout << i << endl;
}</pre>
```

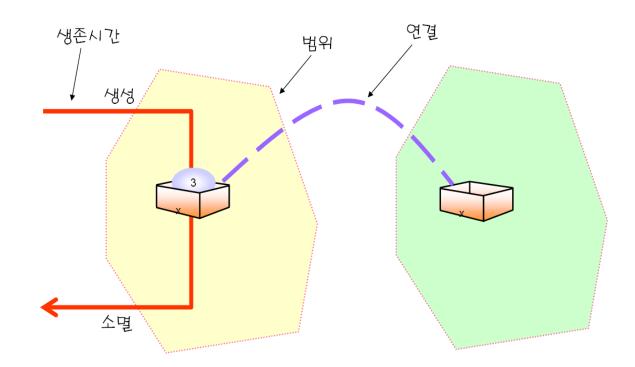
```
for (int i = 0; i<4; i++){
    if (i%2) break;
    else    cout << i << endl;
}</pre>
```

```
15. for (int i = 0; i<4; i++){
    if (i%2) continue;
    else    cout << i << endl;
}
cout << i << endl;</pre>
```

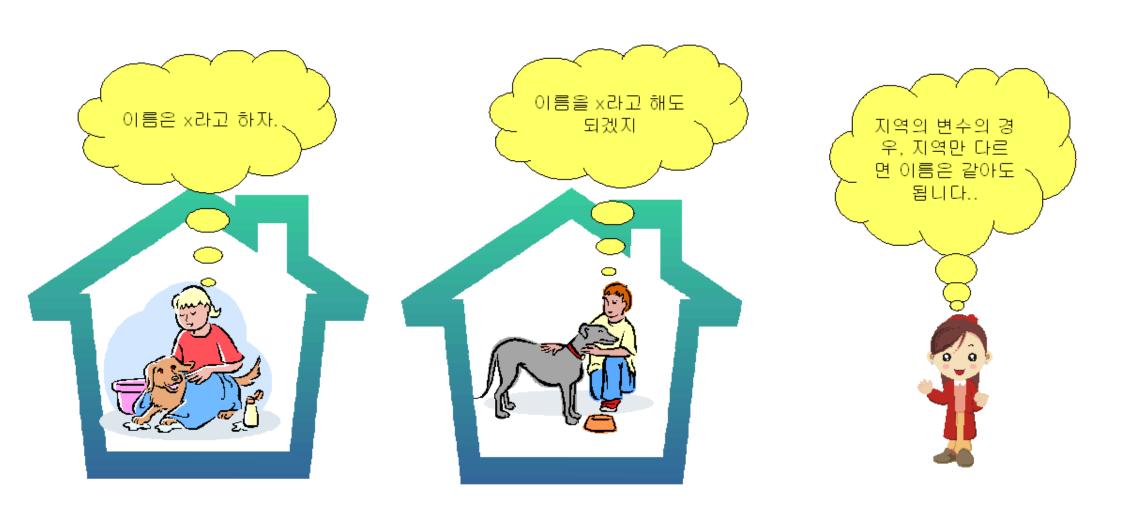
변수의 속성

- 변수의 속성 : 이름, 타입, 크기, 값 + <mark>범위, 생존 시간, 연결</mark>
 - 범위(scope): 변수가 사용 가능한 범위, 가시성

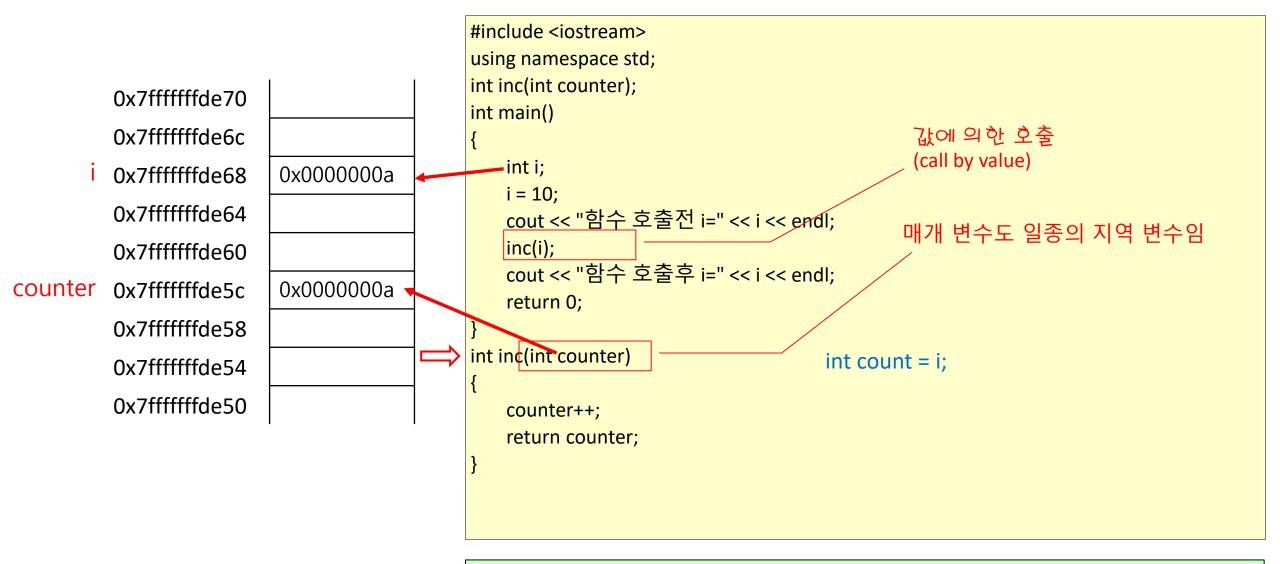
 - 생존시간(lifetime): 메모리에 존재하는 시간
 연결(linkage): 다른 영역에 있는 변수와의 연결 상태



같은 이름의 지역 변수

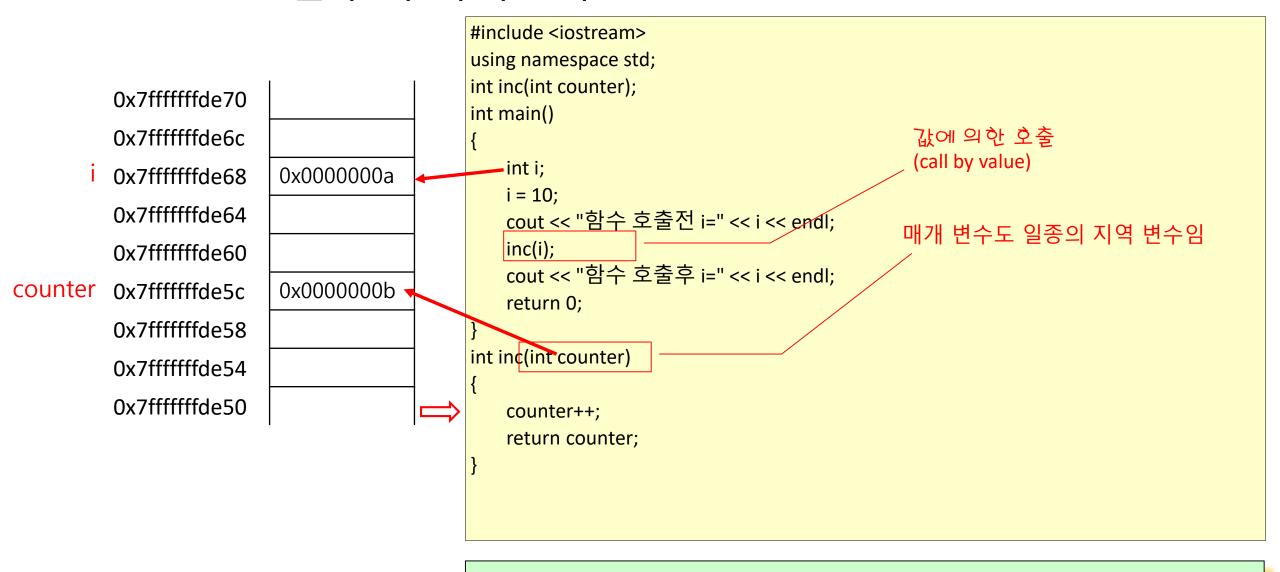


함수의 매개 변수도 지역 변수



함수 호출전 i=10

함수의 매개 변수



함수 호출전 i=10

함수의 매개 변수

```
#include <iostream>
                                               using namespace std;
                                               int inc(int counter);
          0x7fffffffde70
                                               int main()
                                                                                             값에 의한 호출
          0x7fffffffde6c
                                                                                             (call by value)
                                                  int i;
                           0x0000000a
          0x7fffffffde68
                                                  •i = 10;
          0x7fffffffde64
                                                   cout << "함수 호출전 i=" << i << endl;
                                                                                            매개 변수도 일종의 지역 변수임
                                                  inc(i);
          0x7fffffffde60
                                                  cout << "함수 호출후 i=" << i << endl;
                           0x0000000b
counter
          0x7fffffffde5c
                                                   return 0;
          0x7fffffffde58
                                               int inc(int counter)
          0x7fffffffde54
          0x7fffffffde50
                                                   counter++;
                                                   return counter;
```

함수 호출전 i=10 함수 호출후 i=10

전역 변수

• 전역 변수(global variable): 함수의 외부에서 선언되는 변수



전역 변수의 초기값과 생존 기간



```
// 전역 변수의 초기값과 생존 기간
#include <iostream>
                                                         counter=0
using namespace std;
                                                          counter=100
                   <del>// 전역 변수</del>
int counter;
                                                         counter=20
void set counter(int i)
                             // 직접 사용 가능
   counter = i;
                                                 * 전역 변수의 초기값은 0
int main(void)
                                                 *생존 기간은 프로그램 시작부터 종료
   cout << "counter=" << counter << endl;</pre>
                            // 직접 사용 가능
   counter = 100;
   cout << "counter=" << counter << endl;</pre>
   set_counter(20);
   cout << "counter=" << counter << endl;</pre>
   return 0;
```

전역 변수의 사용



```
// 전역 변수를 사용하여 프로그램이 복잡해지는 경우
#include <iostream>
using namespace std;
void f(void);
int i;
int main(void)
    for(i = 0;i < 5; i++)
        f();
    return 0;
void f(void)
    for(i = 0;i < 10; i++)
        cout << "#";
```



##########

같은 이름의 전역 변수와 지역 변수



```
// 동일한 이름의 전역 변수와 지역 변수
#include <iostream>
using namespace std;
                                          지역 변수가 전역 변수를 가린다.
                          // 전역 변수
int sum = 1;
int main()
   int i = 0;
   int sum = 0;
   for(i = 0; i <= 10; i++)
                         // 지역 변수가 전역 변수를 가린다.
        sum += i;
   cout << "sum = " << sum << endl;
   return 0;
```



```
(gdb) p g1
    int g1;
                                            $1 = 0
    int garray[3];
                                            (gdb) p garray
                                            $2 = \{0, 0, 0\}
                                            (gdb) p i 프로그램이 수행되기 전에는 지역 변수는 볼 수 없다.
    int main(){
                                            No symbol "i" in current context.
           i=100, a[3];
9
      int
                                            (gdb) b 14
10
                                            Breakpoint 1 at 0x99f: file scope1.cpp, line 14.
                                            (gdb) r
         int i
12
                                            Starting program: /home/ejim/C2020/scope1
         i = 1;
                                            Breakpoint 1, main () at scope1.cpp:14
14
         a[0] = 20;
                                                         a[0] = 20;
                                            14
                                            (gdb) info locals local 변수들을 모두 보여줌
      cout << "i = " << i << endl;
16
                                             = 1
                                             = 100
                                            a = \{21845, -8320, \overline{32767}\}
18
      for (int i=0; i<3; i++) {}
                                            (gab) pi
19
                                            $3 = 1
20
      cout << "i = " << i << endl:
                                            (gdb) p &i
21
      f1(i);
                                            $4 = (int *) 0x7fffffffde88
                                            (gdb) n
22
      return 0;
                                                      cout << "i = " << i << endl;
                                            16
23
                                            (gdb) info locals
24
    void f1(int a, int b){
                                             = 100
25
      int c = 100;
                                            a = \{20, -8320, 32767\}
26
                                            (qdb) p &i
      a = a+b+c:
                                            $5 = (int *) 0x7fffffffde84
```

void f1(int a, int b=1);

Reading symbols from scope1...done.

```
void f1(int a, int b=1);
                                                = 100
    int g1;
                                                18
                                                          for (int i=0; i<3; i++) {}
    int garray[3];
                                                (gdb)
                                                20
                                                          cout << "i = " << i << endl;
                                                (gdb)
    int main(){
                                                = 100
9
      int
          i=100, a[3];
                                                          f1(i);
                                                21
10
                                                (gdb) s
                                                         step
                                                f1 (a=100, b=1) at scope1.cpp:25
11
                                                         int c = 100;
12
         int i;
                                                (gdb) info locals
         i = 1;
                                                c = 32767
14
         a[0] = 20;
                                               (gdb) bt
                                               #0 f1 (a=100, b=1) at scope1.cpp:25
15
                                                #1 0x0000555555554a38 in main () at scope1.cpp:21
16
      cout << "i = " << i << endl;
                                                (gdb) p g1
17
                                                $1 = 0
18
      for (int i=0; i<3; i++) {}
                                                (gdb) p i
                                                No symbol "i" in current context.
19
                                                (gdb) p &a
20
      cout << "i = " << i << endl:
                                                $2 = <del>(int *)</del> 0x7fffffffde5c
21
      f1(i);
                                                (gdb) disp a
                                                                          0x7fffffffde84 (&i) 와 다름!
      return 0;
                                                1. a = 100
                                                (gdb) n
23
                                                          a = a+b+c;
                                                26
24
    void f1(int a, int b){
                                   int a=i;
                                               1: a = 100
      int c = 100;
                                               (gdb)
26
      a = a+b+c;
                                                1: a = 201
```

(gdb) n

```
int g1;
                                     1: a = 201
                                      (gdb) n
    int garray[3];
                                      main () at scope1.cpp:22
                                      22
                                               return 0;
    int main(){
                                      (gdb) p a
                                      $3 = {20, -8320, 32767}
      int
           i=100, a[3];
                                      (qdb) p i
10
                                      $4 = 100
                                      (gdb) c
                                      Continuing.
12
         int i:
                                      [Inferior 1 (process 6375) exited normally]
         i = 1;
                                      (gdb) pi
                                                        프로그램 수행이 끝난 뒤 지역 변수는 볼 수 없다.
14
         a[0] = 20;
                                      No symbol "i" in current context.
                                      (gdb) p garray
                                      $5 = {0, 0, 0}
16
                                      (qdb) help s
17
                                      Step program until it reaches a different source line.
                                      Usage: step [N]
      for (int i=0; i<3; i++) {}
18
                                      Argument N means step N times (or till program stops for another reason).
19
                                      (gdb) help disp
                      << i << endl;
20
      cout <<
                                     Print value of expression EXP each time the program stops.
                                      /FMT may be used before EXP as in the "print" command.
      f1(i)
21
                                      /FMT "i" or "s" or including a size-letter is allowed,
       return 0;
                                      as in the "x" command, and then EXP is used to get the address to examine
                                      and examining is done as in the "x" command.
24
    void f1(int a, int b){
                                      With no argument, display all currently requested auto-display expressions
      int c = 100;
                                      Use "undisplay" to cancel display requests previously made.
26
      a = a+b+c:
```

void f1(int a, int b=1);

변수의 생존 기간

- 정적 할당(static allocation):
 - 프로그램 실행 시간 동안 계속 유지
 - 전역 변수는 정적 할당
- 자동 할당(automatic allocation):
 블록에 들어갈때 생성

 - 블록에서 나올때 소멸
 - 지역 변수는 자동 할당
- 생존 기간을 결정하는 요인
 - 변수가 선언된 위치
 - 저장 유형 지정자
- 저장 유형 지정자
 - static
 - extern



변수 생성



변수 소멸

저장 유형 지정자 static (for local variables)



```
auto_count=1
#include <iostream>
                                                   static_count=1
using namespace std;
                                                   auto_count=1
void sub(void);
                                                    static count=2
int main()
                                                   auto_count=1
                                                   static_count=3
   int i;
   for(i = 0; i < 5; i++)
          sub();
                                               정적 지역 변수로서
   return 0;
                                               static을 붙이면 지역 변수가
                                               정적 변수로 된다.
void sub(void)
   int auto count = 0;
   static int static_count = 0;
   auto count++;
   static count++;
   cout << "auto count=" << auto count << endl;</pre>
   cout << "static count=" << static count << endl;</pre>
```

저장 유형 지정자 extern



```
컴파일러에게 변수가 다른 곳에서 선언
extern1.c
                                          되었음을 알린다
#include <iostream>
using namespace std;
          // 전역 변수
int x;
             <mark>/ 현재 소스 파일의 </mark>뒷부분에 선언된 변수
extern int y;
             / 다른 소스 파일의 변수
extern int z;
int main(void)
   extern int x; // 전역 변수 x를 참조한다. 없어도된다.
   x = 10;
  y = 20;
   z = 30;
   return 0;
      // 전역 변수
int y;
```

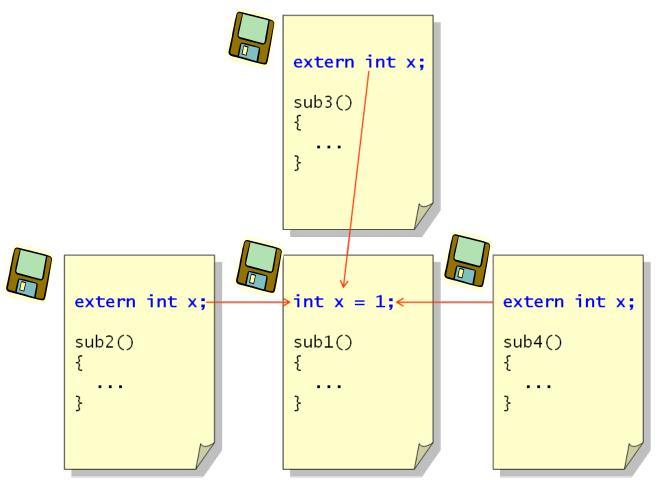


extern2.c

int z;

다중 소스 파일

• 연결은 흔히 다중 소스 파일에서 변수들을 연결하는데 사용된다.



```
// scope2-2.cpp
    #include <iostream>
                                                             _int garray[3];
    using namespace std;
                                                              int i;
    int f1(int a, int b=1);
                                                              void f2(void){
    int g1 = 3;
                                                               extern int g1;
    extern int garray[3]; // = \{4,5,6\};
    int main(){
9
      extern int
10
                   i; •
11
12
      for (int j=0; j<2; j++){
13
         static int i = 1;
14
         i++;
         cout << "inside loop, i = " << i << endl;
15
16
      cout << "outside loop, i = " << i << endl;</pre>
17
18
19
      cout << "f1(i)= " << f1(i) << endl;
20
      cout << "f1(i)= " << f1(1000, 50) << endl:
21
      return 0;
                                  ejim@ejim-VirtualBox:~/C2020$ g++ -g -o scope2 scope2-1.cpp scope2-2.cpp
22 }
                                  ejim@ejim-VirtualBox:~/C2020$ ./scope2
    int fl(int a, int b){
                                  inside loop, i = 2
24
      static int c = 100;
                                  inside loop, i = 3
      c = a+b+c+q1;
25
                                  outside loop, i = 0
                                  f1(i) = 104
26
      return c;
                                  f1(i)= 1157
27 }
```

// scope2-1.cpp

```
// scope2-1.cpp
                                                              // scope2-2.cpp
    #include <iostream>
                                                              #include <iostream>
    using namespace std;
                                                              using namespace std;
    int f1(int a, int b=1);
                                                              int garray[3];
    int g1 = 3;
                                                              int i;
    extern int garray[3]; // = \{4,5,6\};
                                                              void f2(void){
9
    int main(){
10
      extern int i:
                                                                  extern int q1;
11
                                                                  cout << "f1(10,20) = " << f1(10, 20) << endl;
                                                          10
12
      for (int j=0; j<2; j++){
                                                          11 }
         static int i = 1;
13
14
        i++;
        cout << "inside loop, i = " << i << e ejim@ejim-VirtualBox:~/C2020$ g++ -g -o scope2 scope2-1.cpp scope2-2.cpp
15
                                            scope2-1.cpp: In function 'int main()':
16
                                            scope2-1.cpp:21:3: error: 'f2' was not declared in this scope
      cout << "outside loop, i = " << i << end
17
                                               F2();
18
19
      cout << "f1(i)= " << f1(i) << endl;
                                            scope2-1.cpp:21:3: note: suggested alternative: 'f1'
20
      cout << "f1(i)= " << f1(1000, 50) << en
                                               f2();
21
      f2();
                                               ^~
22
      return 0;
                                            scope2-2.cpp: In function 'void f2()':
23
                                            scope2-2.cpp:10:30: error: 'f1' was not declared in this scope
24
                                                cout << "f1(10,20) = " << f1(10, 20) << endl;
25
    int f1(int a, int b){
26
      static int c = 100;
                                            scope2-2.cpp:10:30: note: suggested alternative: 'q1'
27
      c = a+b+c+q1;
                                                cout << "f1(10,20) = " << f1(10, 20) << endl;
28
     return c;
```

q1

29 }

```
extern int garray[3]; // = \{4,5,6\};
     cout << "inside loop, i = " << i << endl;</pre>
  cout << "outside loop, i = " << i << endl;
  cout << "f1(i)= " << f1(i) << endl;
  cout << "f1(i)= " << f1(1000, 50) << endl;
```

// scope2-1.cpp

void f2(void);

int g1 = 3;

int main(){

i++;

f2();

return 0;

int f1(int a, int b){

c = a+b+c+g1;

return c;

static int c = 100;

10

11

12

13

14

15 16

17

18

19

20 21

22

23

24 25

26

27

28 }

#include <iostream>

using namespace std;

extern int i;

for (int j=0; j<2; j++){

static int i = 1;

int f1(int a, int b=1);

```
1 // scope2-2.cpp
    #include <iostream>
    using namespace std;
   int f1(int a, int b=1);
    int garray[3];
    int i;
    void f2(void){
       extern int g1;
       cout << "f1(g1,20) = " << f1(g1, 20) << endl;
10
11 }
```

```
ejim@ejim-VirtualBox:~/C2020$ ./scope2
inside loop, i = 2
inside loop, i = 3
outside loop, i = 0
f1(i)= 104
f1(i)= 1157
f1(g1,20) = 1183
```

```
#include <iostream>
                                                               #include <iostream>
    using namespace std;
                                                               using namespace std;
    void f2(void); _전역 변수와 함수를 static 으로 선언하면?
                                                               int f1(int a, int b=1);
   static int f1(int a, int b=1);
                                                               int garray[3];
   static int g1 = 3;
                                                              int i:
    extern int garray[3]; // = \{4,5,6\};
                                                               void f2(void){
9
    int main(){
                                                                  extern int g1;
                                                           9
10
     extern int i;
                                                                  cout << "f1(g1,20) = " << f1(g1, 20) << endl;
                                                          10
11
     for (int j=0; j<2; j++){
                                                          11 }
12
13
        static int i = 1;
14
        i++;
15
        cout << "inside loop, i = " << i << endl;</pre>
16
17
     cout << "outside loop, i = " << i << endl;</pre>
18
     cout << "f1(i)= " << f1(i) << endl;
19
20
     cout << "f1(i)= " << f1(1000, 50) << endl;
21
     f2();
                           ejim@ejim-VirtualBox:~/C2020$ g++ -g -o scope2 scope2-1.cpp scope2-2.cpp
22
      return 0;
                           /tmp/ccsY0pC3.o: In function `f2()':
23
                           /home/ejim/C2020/scope2-2.cpp:10: undefined reference to [g1'
24
   int f1(int a, int b){
25
     static int c = 100;
                           /home/ejim/C2020/scope2-2.cpp:10: undefined reference to `f1(int, int)'
26
     c = a+b+c+q1;
27
     return c;
28 }
```

1 // scope2-2.cpp

// scope2-1.cpp

저장 유형 지정자 static (for global variables & functions)

- 전역 변수나 함수의 static 선언은 지역 변수 static 선언과 의미가 다르다.
- 선언된 file 외부에서 접근할 수 없다는 뜻이다.

static global variable



```
#include <iostream>
using namespace std;
int all_files; // 다른 소스 파일에서도 사용할 수 있는 전역 변수
static int this_file; // 현재의 소스 파일에서만 사용할 수 있는 전역 변수
extern void sub();
int main()
{
sub();
cout << all_files << endl;
return 0;
}
```

```
linkage2.c
```

```
extern int all_files:

void sub()
{
    all_files = 10;
}
```



static function

main.c

```
#include <iostream>
using namespace std;
extern void f2();
int main(void)
   f2();
   return 0;
                                            static 함수는 파일
                                            안에서만 사용할 수 있다
sub.c
#include <iostream>
using namespace std;
static void f1()
   cout << "f1()이 호출되었습니다.\n";
void f2()
   f1();
   cout << "f2()가 호출되었습니다.\n";
```

```
// scope2-1.cpp
                                                              // scope2-2.cpp
    #include <iostream>
                                                              #include <iostream>
    using namespace std;
                                                              using namespace std;
    int f1(int a, int b=1);
                                                              int garray[3];
    int g1 = 3;
                                                              int i;
    extern int garray[3]; // = \{4,5,6\};
                                                              void f2(void){
9
    int main(){
10
      extern int i:
                                                                  extern int q1;
11
                                                                  cout << "f1(10,20) = " << f1(10, 20) << endl;
                                                          10
12
      for (int j=0; j<2; j++){
                                                          11 }
         static int i = 1;
13
14
        i++;
        cout << "inside loop, i = " << i << e ejim@ejim-VirtualBox:~/C2020$ g++ -g -o scope2 scope2-1.cpp scope2-2.cpp
15
                                            scope2-1.cpp: In function 'int main()':
16
                                            scope2-1.cpp:21:3: error: 'f2' was not declared in this scope
      cout << "outside loop, i = " << i << end
17
                                               F2();
18
19
      cout << "f1(i)= " << f1(i) << endl;
                                            scope2-1.cpp:21:3: note: suggested alternative: 'f1'
20
      cout << "f1(i)= " << f1(1000, 50) << en
                                               f2();
21
      f2();
                                               ^~
22
      return 0;
                                            scope2-2.cpp: In function 'void f2()':
23
                                            scope2-2.cpp:10:30: error: 'f1' was not declared in this scope
24
                                                cout << "f1(10,20) = " << f1(10, 20) << endl;
25
    int f1(int a, int b){
26
      static int c = 100;
                                            scope2-2.cpp:10:30: note: suggested alternative: 'q1'
27
      c = a+b+c+q1;
                                                cout << "f1(10,20) = " << f1(10, 20) << endl;
28
     return c;
```

q1

29 }

```
extern int garray[3]; // = \{4,5,6\};
     cout << "inside loop, i = " << i << endl;</pre>
  cout << "outside loop, i = " << i << endl;
  cout << "f1(i)= " << f1(i) << endl;
  cout << "f1(i)= " << f1(1000, 50) << endl;
```

// scope2-1.cpp

void f2(void);

int g1 = 3;

int main(){

i++;

f2();

return 0;

int f1(int a, int b){

c = a+b+c+g1;

return c;

static int c = 100;

10

11

12

13

14

15 16

17

18

19

20 21

22

23

24 25

26

27

28 }

#include <iostream>

using namespace std;

extern int i;

for (int j=0; j<2; j++){

static int i = 1;

int f1(int a, int b=1);

```
1 // scope2-2.cpp
    #include <iostream>
    using namespace std;
   int f1(int a, int b=1);
    int garray[3];
    int i;
    void f2(void){
       extern int g1;
       cout << "f1(g1,20) = " << f1(g1, 20) << endl;
10
11 }
```

```
ejim@ejim-VirtualBox:~/C2020$ ./scope2
inside loop, i = 2
inside loop, i = 3
outside loop, i = 0
f1(i)= 104
f1(i)= 1157
f1(g1,20) = 1183
```

function prototype 은 헤더파일로 만드는 것이 좋다.

```
// scope2-1.cpp
                                                        // scope2-1.h
                                                                                             // scope2-2.h
    #include <iostream>
                                                                                             void f2(void);
                                                        int fl(int a, int b=1);
    #include "scope2-1.h"
    #include "scope2-2.h"
    using namespace std;
                                                       1 // scope2-2.cpp
                                                           #include <iostream>
    int g1 = 3;
                                                           #include "scope2-1.h"
    extern int garray[3]; // = \{4,5,6\};
                                                           #include "scope2-2.h"
    int main(){
10
                                                           using namespace std;
11
     extern int i;
                                                           int garray[3];
     for (int j=0; j<2; j++){
                                                           int i;
1
        static int i = 1;
        i++;
        cout << "inside loop, i = " << i << endl;</pre>
16
                                                           void f2(void){
17
                                                              extern int gl;
                                                      11
     cout << "outside loop, i = " << i << endl;</pre>
18
                                                      12
                                                              cout << "f1(q1,20) = " << f1(q1, 20) << endl;
19
                                                      13 }
     cout << "f1(i)= " << f1(i) << endl;
20
     cout << "f1(i)= " << f1(1000, 50) ejim@ejim-VirtualBox:~/C2020\$ g++ -g -o scope2 scope2-1.cpp scope2-2.cpp
21
22
     f2();
                                   ejim@ejim-VirtualBox:~/C2020$ ./scope2
23
     return 0;
                                   inside loop, i = 2
24 }
                                   inside loop, i = 3
    int f1(int a, int b){
25
                                   outside loop, i = 0
     static int c = 100;
26
                                   f1(i)= 104
     c = a+b+c+g1;
27
                                   f1(i)= 1157
28
     return c;
                                   f1(g1,20) = 1183
29 }
```