

고급C프로그래밍

10 기타 추가 문법들

공지사항 – Quiz(오프라인 코딩테스트)

- 진행일시 : 12/3(수), 수업시간
- 진행방식
 - ✓ 일주일 전 11/26(수) 문제 공개
 - ✓ 일주일간 공개된 문제에 대해 준비
 - ✓ 12/3(수) 오프라인 코딩테스트 후, 부족한 부분을 보완하여 12/10까지 과제보고서(설계보고서, 소스코드) 제출
 - ✓ 결과보고서는 제출한 소스코드와 일치해야 함
 - ✓ 과제보고서 점수 : 보고서 점수 * 오프라인 코딩테스트 점수
 - ✓ 부정행위 적발 시 F학점 처리

(1) main 함수

main() 함수의 매개변수 (1/4)

- **매개변수를 받아서 main() 함수를 만들고 빌드한 후 실행하는 형식**

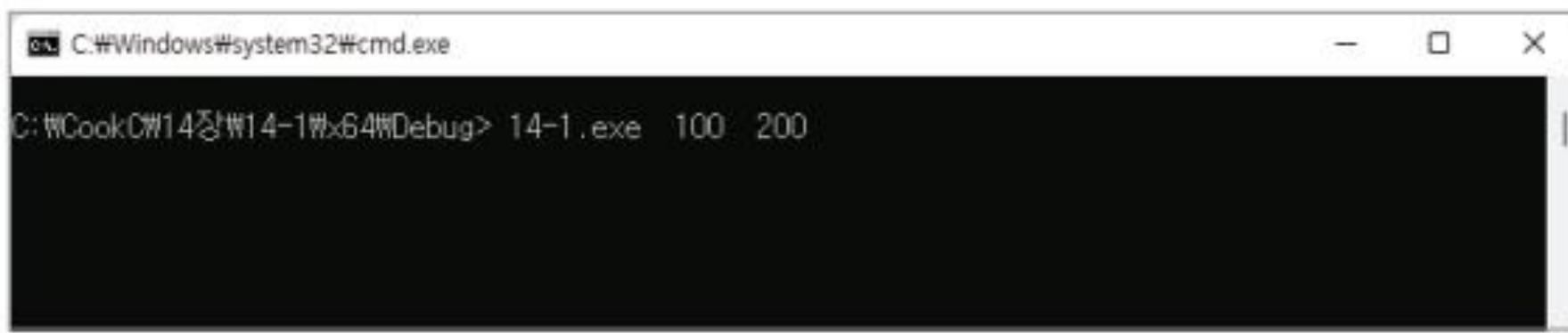


그림 14-1 main() 함수에서의 매개변수 사용

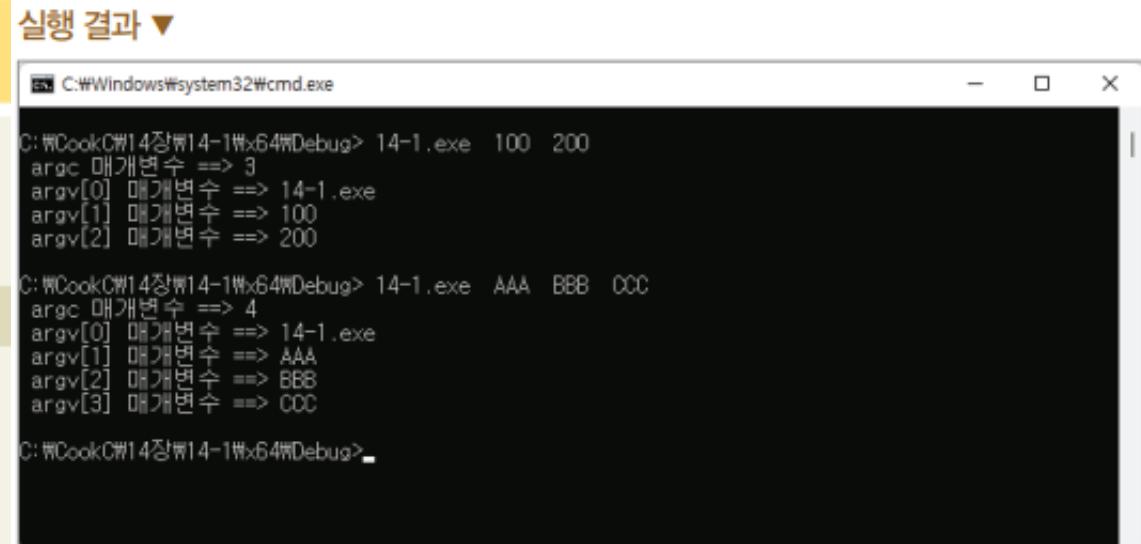
main() 함수의 매개변수 (2/4)

- **main() 함수의 매개변수를 사용하는 대표적인 예**

- ✓ argc : 명령행 인수(command line arguments)의 수
- ✓ argv[] : 명령행 인수, 첫번째는 프로그램의 이름

기본 14-1 main() 함수에서의 매개변수 사용 예 1

```
01 #include <stdio.h>
02
03 void main(int argc, char* argv[])
04 {
05     int i;
06
07     printf(" argc 매개변수 ==> %d \n", argc);
08
09     for(i=0; i < argc; i++)
10         printf(" argv[%d] 매개변수 ==> %s \n", i, argv[i]);
11 }
```



---- 매개변수의 개수를 출력한다.

---- 매개변수의 개수만큼 반복하여
매개변수의 내용을 출력한다.

main() 함수의 매개변수 (3/4)

• void main(int argc, char* argv[])의 다른 사용예

응용 14-2 main() 함수에서의 매개변수 사용 예 2

14-2.c

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <stdio.h>
03 void main(int argc, char* argv[])
04 {
05     char str[200];
06     FILE *rfp;
07
08     if(__1__ != 2)           ---- 매개변수가 하나가 아니면 메시지를 출력한 후
09     {                      프로그램을 종료한다.
10         printf("\n - 매개변수를 1개 사용하세요 -\n");
11         return;
12     }
13
14     rfp=fopen(__2__, "r");   ---- 첫 번째 매개변수로 넘어온 것을 읽기 모드로 연다.
15
16     for( ; ; )              ---- 무한 루프이다.
17     {
18         fgets(str, 199, rfp); ---- 파일의 내용을 한 줄 읽어들인다.
19
20         if(feof(rfp))       ---- 파일의 끝이면 무한 루프를 종료한다.
```

main() 함수의 매개변수 (4/4)

- void main(int argc, char* argv[])의 다른 사용예

```
21     break;
22
23     printf("%s", str);      ----- 읽을 내용을 출력한다.
24 }
25
26 fclose(rfp);           ----- 파일을 닫는다.
27 }
```

선택 1 argc 2 argv[1]

실행 결과 ▼

```
C:\Windows\system32\cmd.exe
C:\Cook\14장\14-2\Debug> 14-2.exe
-- 매개변수를 1개 사용하세요 --
C:\Cook\14장\14-2\Debug> 14-2.exe AAA BBB
-- 매개변수를 1개 사용하세요 --
C:\Cook\14장\14-2\Debug> 14-2.exe C:\Windows\win.ini
: for 16-bit app support
[fonts]
[extensions]
[mci extensions]
[files]
[Mail]
MAP=1

C:\Cook\14장\14-2\Debug>
```

(2) 헤더 파일

헤더 파일의 이해 (1/2)

- *.h라는 확장자를 사용하는 파일

- ✓ 파일에는 함수의 프로토 타입이 선언되어 있으며 구조체 등의 데이터 구조가 정의되어 있음

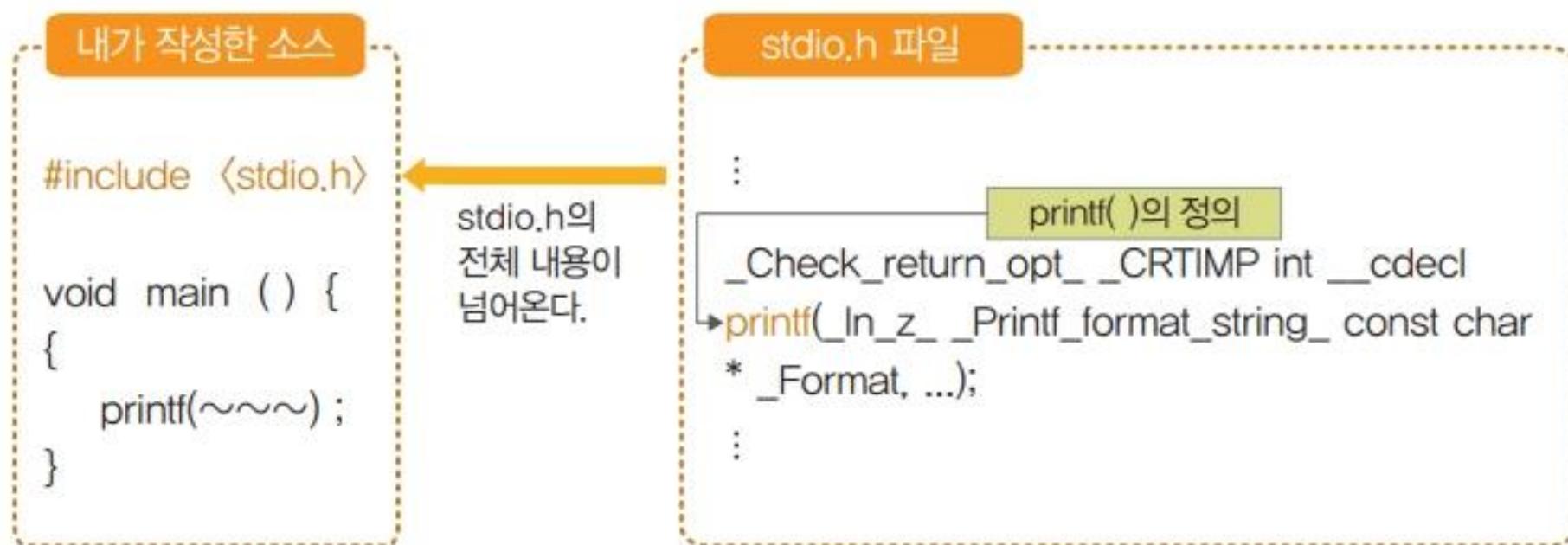


그림 14-2 #include의 개념

헤더 파일의 이해 (2/2)

• 헤더 파일과 관련된 함수의 종류

표 14-1 주요 헤더 파일과 관련 함수

헤더 파일	설명	관련 함수
stdio.h	표준 입출력 관련	printf() scanf() puts() gets() fopen() fclose() ...
string.h	문자열 관련	strcat() strcmp() strlen() strcpy() ...
math.h	수학 관련	sin() cos() abs() pow() sqrt() log10() ...
malloc.h	메모리 관련	malloc() realloc() calloc() free() ...
stdlib.h	C 표준 라이브러리 관련	exit() rand() system() ...
time.h	시간 관련	clock() time() localtime() ...

사용자가 만드는 헤더 파일 (1/2)

- 사용자가 만든 헤더 파일의 이름이 ‘myheader.h’ 일 경우
 - ✓ Visual Studio를 실행한 뒤 ‘14_3’이라는 이름으로 프로젝트를 생성하고 ‘14_3.c’ 파일 생성
 - ✓ 왼쪽 솔루션 탐색기의 [14_3]-[헤더 파일]에서 마우스 오른쪽 버튼 클릭 후 [추가(D)]-[새 항목]을 선택



그림 14-3 헤더 파일 추가 1

사용자가 만드는 헤더 파일 (2/2)

- 사용자가 만든 헤더 파일의 이름이 ‘myheader.h’ 일 경우 (계속)

- ✓ [헤더 파일]을 선택하고 이름에 “myHeader.h”를 입력한 후 <추가>를 클릭

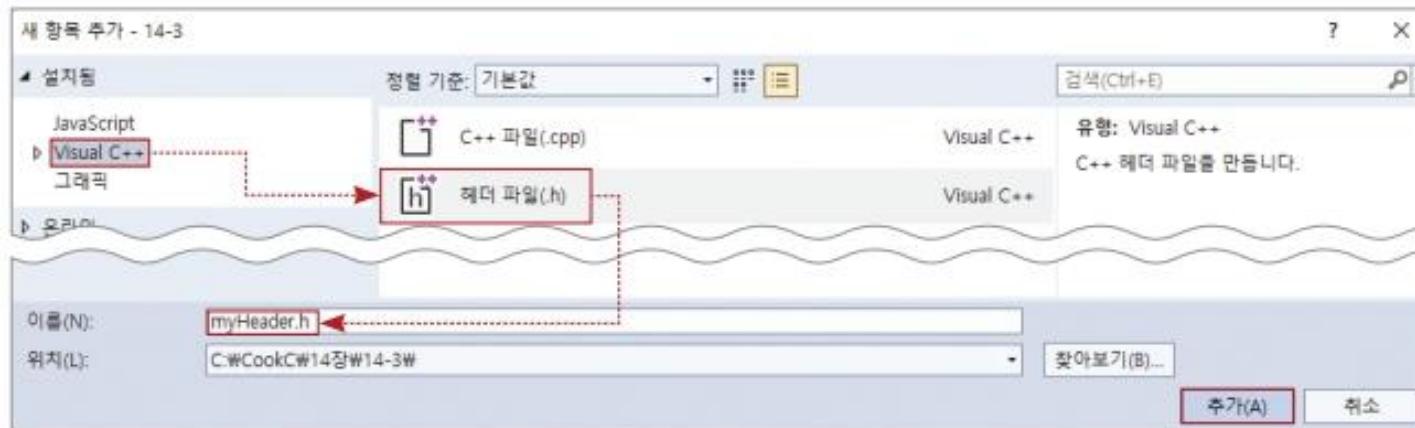


그림 14-4 헤더 파일 추가 2

- ✓ 헤더 파일의 내용에는 필요한 내용을 기술

- ✓ 예)

```
void func(int);
```

복수의 파일로 구성된 프로그램 만들기 (1/2)

- **다수의 헤더 파일과 다수의 소스 파일로 프로그램 구성 가능**
 - ✓ 다수의 파일을 빌드해서 하나의 실행 파일을 생성
- **어떤 파일을 헤더 파일로 만들고 어떤 파일을 소스 파일로 만들 것인가?**
 - ✓ 헤더 파일 : 복수의 프로그램에서 공통으로 사용
 - ✓ 소스 파일 : 특정 프로그램의 기능을 기술

복수의 파일로 구성된 프로그램 만들기 (2/2)

복수의 파일로 구성된 프로그램 만들기 예

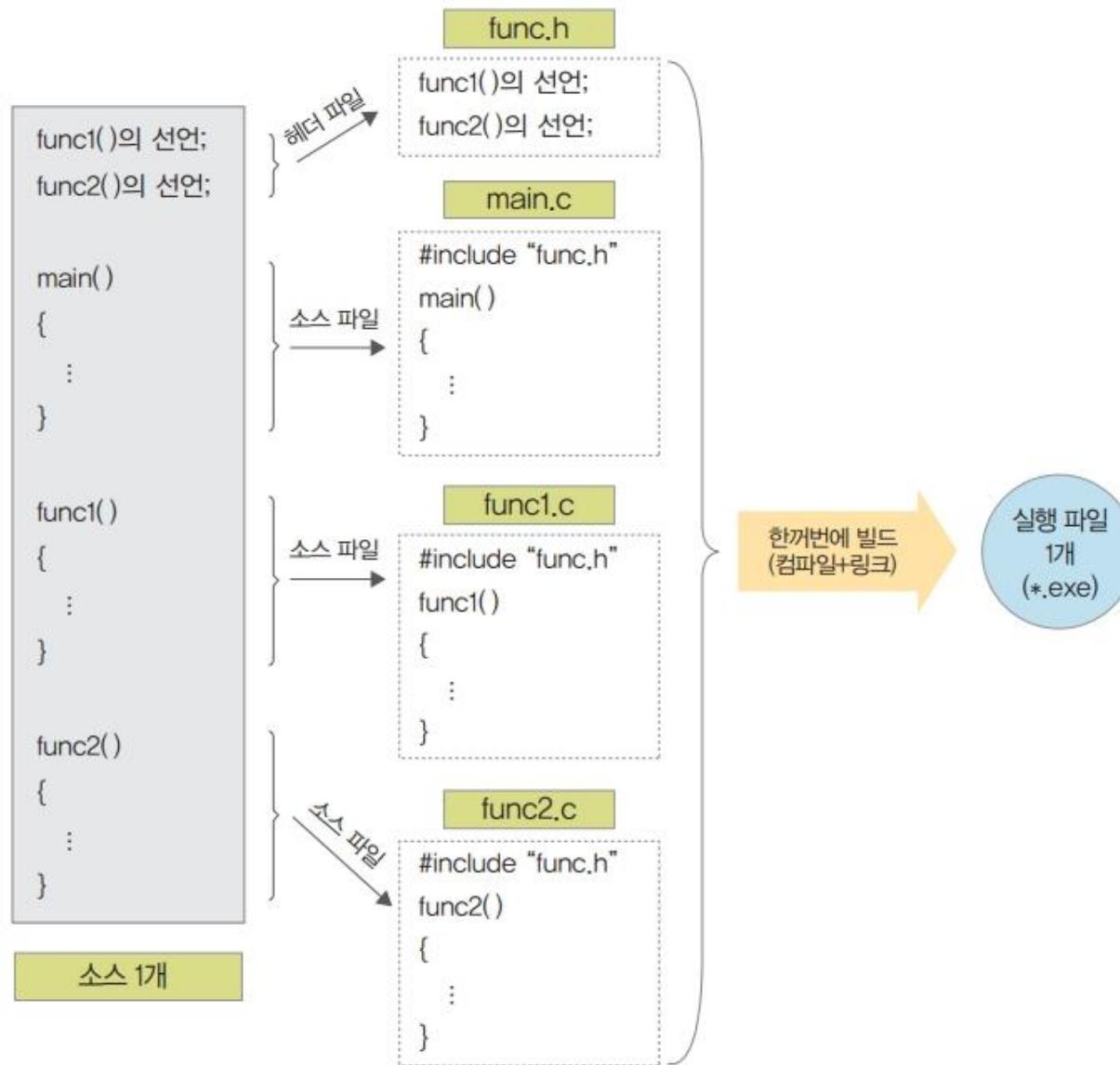


그림 14-5 파일 분리의 개념

(3) 전처리문과 예약어

전처리문 (1/2)

- **실제 컴파일 전에 미리 처리되는 문장**

- ✓ 기존의 방대한 소스를 건드리지 않은 상태에서 부분적인 컴파일 수행
- ✓ 소스의 시작부분에 위치하며, #으로 시작
- ✓ #include, #define, #ifdef, #undef 등

전처리문 (2/2)

- **#define 문**

- ✓ 소스 코드에 사용할 숫자나 문자열, 함수의 이름이 너무 길거나 복잡할 때 한 눈에 파악하도록 쉬운 기호로 표현

```
#define [기호] [숫자 또는 문자열 또는 함수]
```

- ✓ 예) 원주율 표시 – 3.1415926535

```
void main( ){  
...  
area = rad * rad * 3.1415926535;  
...  
}
```

```
#define PI 3.1415926535  
void main( ){  
...  
area = rad * rad * PI;  
...  
}
```

예약어 (1/6)

- 특별한 기능을 수행하도록 프로그래밍 언어에서 미리 정의한 것
 - ✓ 변수나 함수 이름에 사용할 수 없음
- **const** 예약어
 - ✓ #define과 비슷한 기능 (상수 정의 시 const 사용을 더 권장함)
 - ✓ const로 변수를 선언 → 상수 역할을 함

```
const 변수형 변수_이름 = 변수값;
```

예약어 (2/6)

- **const** 예약어 (계속)

- ✓ **const**의 사용 방법

```
const int a = 100;      ⇒ a를 100으로 고정한다.
```

```
int b;
```

```
b = a + 200;          ⇒ (o) a는 100이므로 b는 300이 된다.
```

```
a = 200;              ⇒ (x) a는 더 이상 변수가 아니며 고정된 값으로만 사용해야 한다.
```

- ✓ **const**를 사용한 위의 예를 **#define**으로 바꿈

```
#define a 100           ⇒ 100을 기호 a로 정의한다.
```

```
int b;
```

```
b = a + 200;          ⇒ (o) a는 100과 동일하므로 b는 300이 된다.
```

```
a = 200;              ⇒ (x) 100=200이라고 쓴 것이므로 문법적으로 틀린 문장이다.
```

예약어 (3/6)

• static 예약어

- ✓ 한번 초기화하면, 다시 초기화 되지 않도록 함

```
static 변수형 변수_이름 = 변수값;
```

기본 14-6 static 예약어 사용 예

14-6.c

```
01 #include <stdio.h>
02
03 void myfunc( );           —— myfunc( ) 함수의 프로토타입을 선언한다.
04
05 void main( )
06 {
07     myfunc( );            —— myfunc( ) 함수를 두 번 호출한다.
08     myfunc( );
09 }
10
11 void myfunc( )
12 {
13     static int a = 0;      —— static 예약어로 a 변수를 선언하고 초기화한다.
14
15     a = a + 100;          —— a에 100을 증가시킨 후 출력한다.
16     printf("a의 값 ==> %d\n", a);
17 }
```

실행 결과

```
a의 값 ==> 100
a의 값 ==> 200
```

예약어 (4/6)

• **extern** 예약어

- ✓ 다른 소스 파일에 선언된 전역 변수를 현재의 소스 파일에 가져와서 사용하고 싶을 때 사용
- ✓ 다수의 소스 파일로 프로그램을 구성할 때 사용

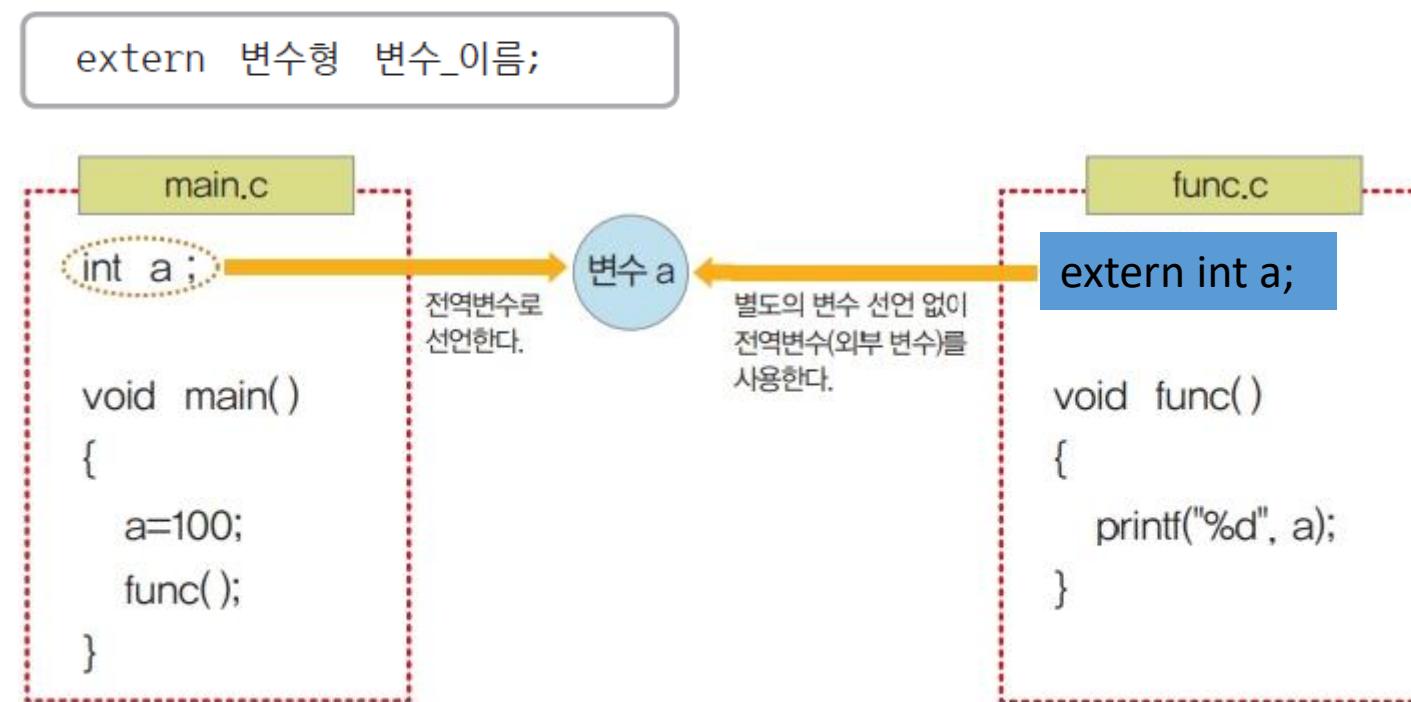


그림 14-7 **extern** 예약어 사용

예약어 (5/6)

• **extern** 예약어 사용 예

기본 14-7 extern 예약어 사용 예 1

14-7.c

main.c

```
01 #include <stdio.h>
02
03 void func( );
04
05 int a;
06
07 void main( )
08 {
09     a = 100;
10
11     func( );
12 }
```

—— func() 함수의 프로토타입을 선언한다.

—— 전역변수 a를 선언한다.

—— 전역변수에 값을 대입한다.

—— 함수를 호출한다.

예약어 (6/6)

- **extern** 예약어 사용예

func.c

01 #include <stdio.h>

02

03 extern int a;

---- 변수 a를 외부 변수로 선언한다.

04

05 void func()

06 {

07 printf("extern int a값 ==> %d\n", a);

---- 변수 a의 값을 출력한다.

08 }

3 실행 결과

extern int a값 ==> 100

실습

[실습 1] 도서관 프로그램(9주차 [실습 3]) 확장

- **도서관 관리자 모드 추가 구현**

- ✓ 관리자 모드에서는 현재 도서의 보유 재고 현황을 출력함

- **프로그램의 동작**

- ✓ 파일 실행 시 “admin”를 입력하면 관리자 모드로 진입
→ void main(int argc, char * argv[]) 사용
 - ✓ 파일 실행 시 “admin”를 입력하면 관리자 모드로 진입
 - ✓ 관리자 모드에서는 관리자 ID/PW를 입력 받아 로그인 처리
→ 관리자 ID/PW는 “admin_auth.txt”에 저장되어 있음
 - ✓ 관리자 인증을 받으면, 현재 보유 도서의 재고 현황을 화면에 출력
→ 이후 서비스 모드로 전환(또는 종료) 여부는 사용자에게 물어보고 결정

[실습 1] 도서관 프로그램(9주차 [실습 3]) 확장

- **프로그램의 구성**

- ✓ main () : 관리자 모드와 서비스 모드 관련 사용자와의 상호작용 및 필요한 함수 호출
- ✓ initial_booklist () : 보유 도서 현황을 초기화
- ✓ manage_booklist () : 사용자 요청에 따라 보유 도서 현황(또는 대여 도서 현황)을 업데이트
- ✓ admin_check () : 관리자 로그인 처리
- ✓ print_booklist () : 현재 보유 도서 현황을 출력

- **데이터 구조는 9주차 실습 3를 활용**

[실습 2] POS 프로그램(7주차 [실습 3]) 확장

- 주문 가능한 메뉴를 확장 : 음료와 디저트 → 두 개의 구조체 변수 필요
 - ✓ 각 메뉴는 ID, 이름, 가격으로 구성
 - ✓ 음료와 디저트 메뉴는 drink.txt, dessert.txt에서 정보를 획득
- 프로그램의 동작
 - ✓ 사용자에게 주문 메뉴를 출력 : 1.음료 주문 2.디저트 주문 0.주문 종료
 - ✓ 음료 주문 시 음료 메뉴 리스트(ID, 이름, 가격) 제공
 - ✓ 사용자는 ID 입력을 통해 음료를 주문
 - ✓ 음료를 주문하면 다시 주문 메뉴를 출력, 사용자는 반복을 통해 복수개의 음료, 디저트를 주문할 수 있음
 - ✓ 주문 종료 선택 시 현재까지 주문한 메뉴의 영수증(주문 리스트와 총액) 출력
→ receipt.txt에 저장 또는 화면에 출력

[실습 2] POS 프로그램(7주차 [실습 3]) 확장

- **프로그램의 구성**

- ✓ main () : 사용자에게 주문 메뉴를 출력하고, 필요한 함수들을 호출함
- ✓ Initial_menulist () : 음료/디저트 메뉴 리스트를 초기화
- ✓ drink_count () : 음료 주문을 처리
- ✓ dessert_count () : 디저트 주문을 처리
- ✓ print_receipt () : 영수증을 출력

[실습 2] POS 프로그램(8주차 [실습 3]) 확장

• 결과화면 예시

```
1. 음료 주문하기  
2. 디저트 주문하기  
0. 종료 :: 1  
주문하실 음료를 선택하세요  
  
drink menues  
[1] cappuccino      2.99  
[2] espresso        0.99  
[3] strawberry_milk 1.99  
[4] strawberry_ade   1.99  
[5] vanilla_latte   2.99  
5  
  
1. 음료 주문하기  
2. 디저트 주문하기  
0. 종료 :: 2  
주문하실 디저트를 선택하세요  
  
dessert menues  
[1] scone            3.55  
[2] bread             7.50  
[3] pie               9.88  
[4] chicken           8.88  
[5] noodle            3.80  
1  
  
1. 음료 주문하기  
2. 디저트 주문하기  
0. 종료 :: 1  
주문하실 음료를 선택하세요  
  
drink menues  
[1] cappuccino      2.99  
[2] espresso        0.99  
[3] strawberry_milk 1.99  
[4] strawberry_ade   1.99  
[5] vanilla_latte   2.99  
5  
  
1. 음료 주문하기  
2. 디저트 주문하기  
0. 종료 :: 0  
영수증이 Receipt.txt에 저장되었습니다.
```

```
☰ Receipt.txt  
1 =====  
2 | Receipt  
3 =====  
4 | vanilla_latte    2          5.98  
5 | scone            1          3.55  
6 =====  
7 | Total ::          9.53  
8 |
```

[실습 3] 도서관 프로그램([실습 1]) 확장

- **재고 확장 기능 추가**

- ✓ 관리자 모드로 진입 시, 창고에서 도서를 가져와서 재고를 확장하는 기능 추가
- ✓ storage.txt가 창고 역할을 수행하며, 여분의 도서가 보관되어 있음
- ✓ 관리자 모드로 들어가면 보유 도서가 0인 도서에 대해서 창고에서 도서를 가져와서 보유 도서 숫자를 3권으로 확장

→ 창고에서는 잔여 도서 숫자를 감소시켜야 함

- ✓ 정리하면, 관리자 모드 진입 시 보유 도서 현황을 출력하고, 창고에서 여분의 도서를 가져와서 보유 도서 현황을 업데이트 한 후, 창고에 남아있는 도서 현황을 출력

★ 프로그램 테스트를 위해 storage.txt의 보유 도서 숫자를 인위적으로 감소시세요!

Q & A