

고급C프로그래밍

10 기타 추가 문법들

공지사항 – Quiz(오프라인 코딩테스트)

- **진행일시 : 12/3(수), 수업시간**

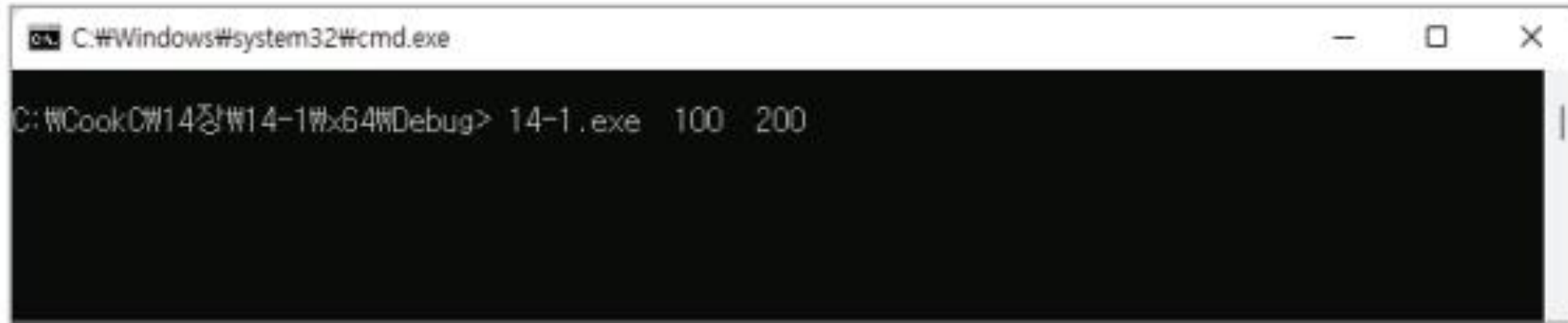
- **진행방식**

- ✓ 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



```
C:\Windows\system32\cmd.exe
C:\Cook\14장\14-1\64\Debug> 14-1.exe 100 200
```

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 }
```

실행 결과 ▼

```
C:\Windows\system32\cmd.exe
C:\CookCW14장\14-1\Debug> 14-1.exe 100 200
argc 매개변수 ==> 3
argv[0] 매개변수 ==> 14-1.exe
argv[1] 매개변수 ==> 100
argv[2] 매개변수 ==> 200
C:\CookCW14장\14-1\Debug> 14-1.exe AAA BBB CCC
argc 매개변수 ==> 4
argv[0] 매개변수 ==> 14-1.exe
argv[1] 매개변수 ==> AAA
argv[2] 매개변수 ==> BBB
argv[3] 매개변수 ==> CCC
C:\CookCW14장\14-1\Debug>
```

----- 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 읽어들이다.
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 }
```

[1] argc 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000

실행 결과 ▼

```
C:\Windows\system32\cmd.exe
C:\Cook\14장\14-2\64\Debug> 14-2.exe
-- 매개변수를 1개 사용하세요 --
C:\Cook\14장\14-2\64\Debug> 14-2.exe AAA BBB
-- 매개변수를 1개 사용하세요 --
C:\Cook\14장\14-2\64\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\64\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

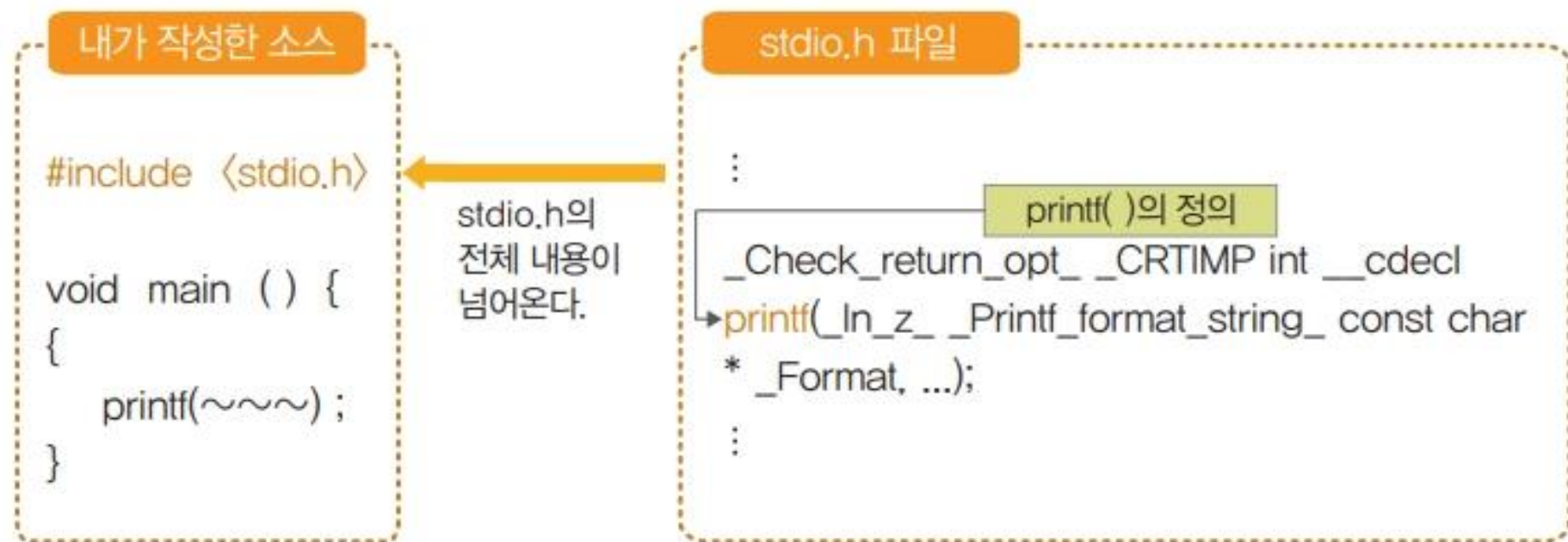


그림 14-2 #include의 개념

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

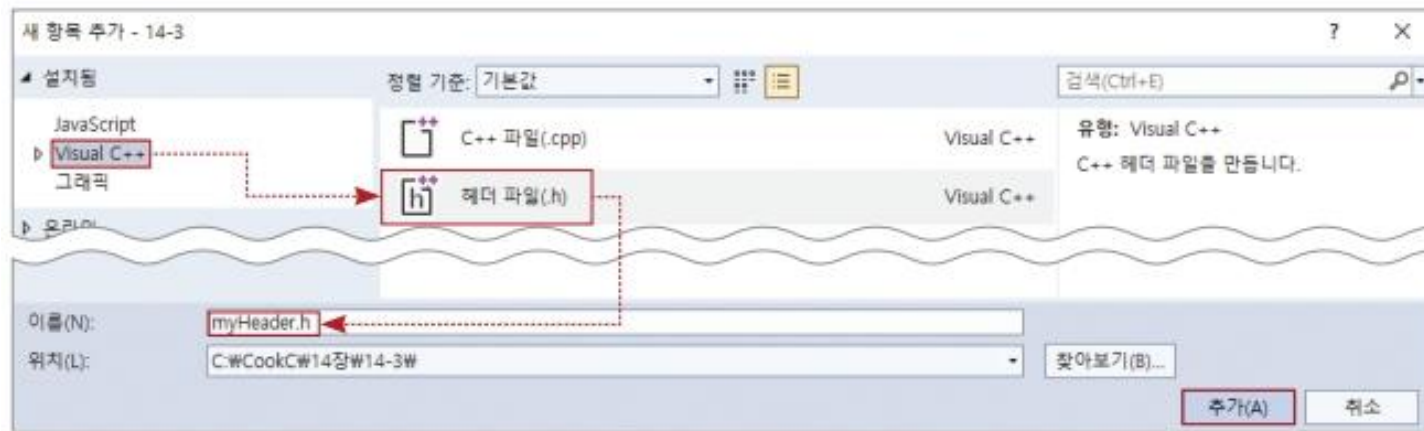


그림 14-4 헤더 파일 추가 2

- ✓ 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

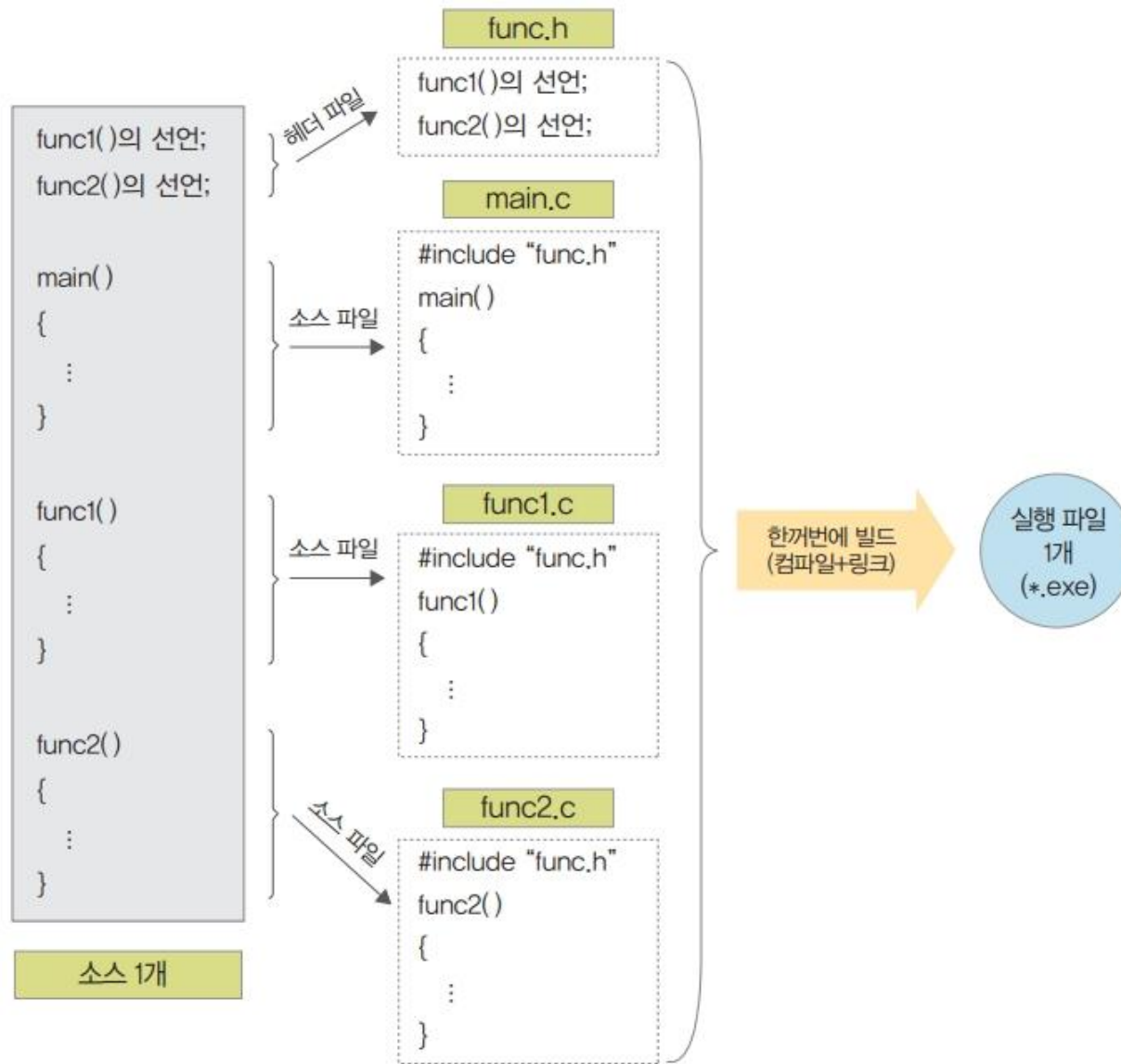


그림 14-5 파일 분리의 개념

(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)**

- ✓ const의 사용 방법

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

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

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

- ✓ const를 사용한 위의 예를 #define으로 바꿈

```
#define a 100          ⇒ 100을 기호 a로 정의한다.  
int b;
```

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

```
a = 200;              ⇒ (×) 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

extern 변수형 변수_이름;

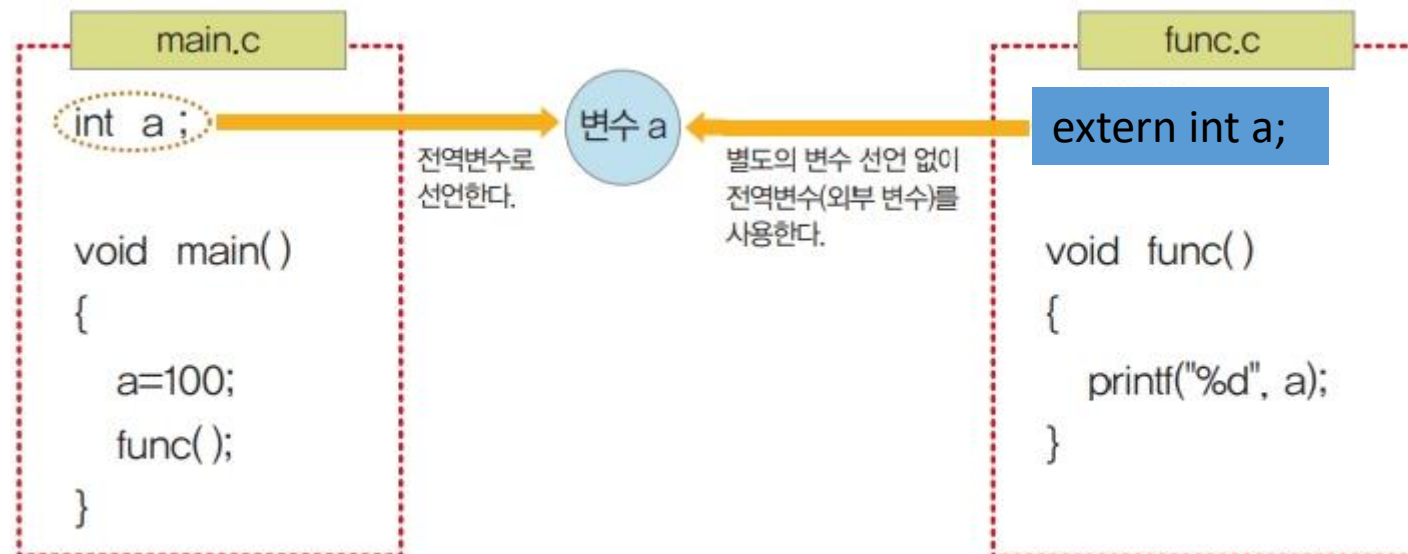


그림 14-7 extern 예약어 사용

예약어 (5/6)

- **extern** 예약어 사용예

기본 14-7 extern 예약어 사용 예 1

14-7.c

main.c

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

```
02
```

```
03 void func( );
```

—— func() 함수의 프로토타입을 선언한다.

```
04
```

```
05 int a;
```

—— 전역변수 a를 선언한다.

```
06
```

```
07 void main( )
```

```
08 {
```

1

```
09     a = 100;
```

—— 전역변수에 값을 대입한다.

```
10
```

```
11     func( );
```

—— 함수를 호출한다.

```
12 }
```

예약어 (6/6)

- **extern** 예약어 사용예

func.c

01 #include <stdio.h>

02

2 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 }

실행 결과

3 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 menus
[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 menus
[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 menus
[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 감소시켜주세요!

Q & A