

쉽게 풀어쓴 C언어 Express[개정판] 천인국 저, 생능출판사 2012

2장. 프로그래밍 작성 과정

성결대학교 컴퓨터공학부
임 상 순

강의 목표 및 내용

▶ 강의 목표

- Visual Studio를 이용한 프로그램 작성 과정을 살펴본다.
- 예제 프로그램을 작성하고 컴파일, 실행하여 본다.
- 예제 프로그램을 이해한다.

▶ 내용

- 통합 개발 환경
- Visual Studio를 이용한 프로그램 작성
- 예제 프로그램의 설명
- 예제 프로그램의 활용
- 오류 수정 및 디버깅

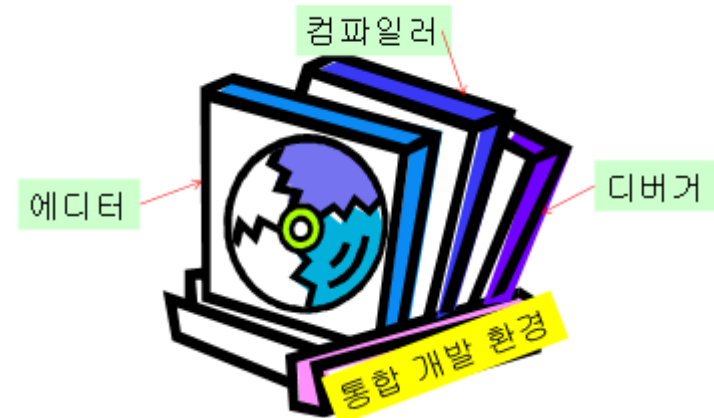
통합 개발 환경

▶ 통합 개발 환경

- IDE(Integrated Development Environment)
- 에디터 + 컴파일러 + 디버거
- 프로그램 개발과 관련된 전체 과정을 쉽게 해줌

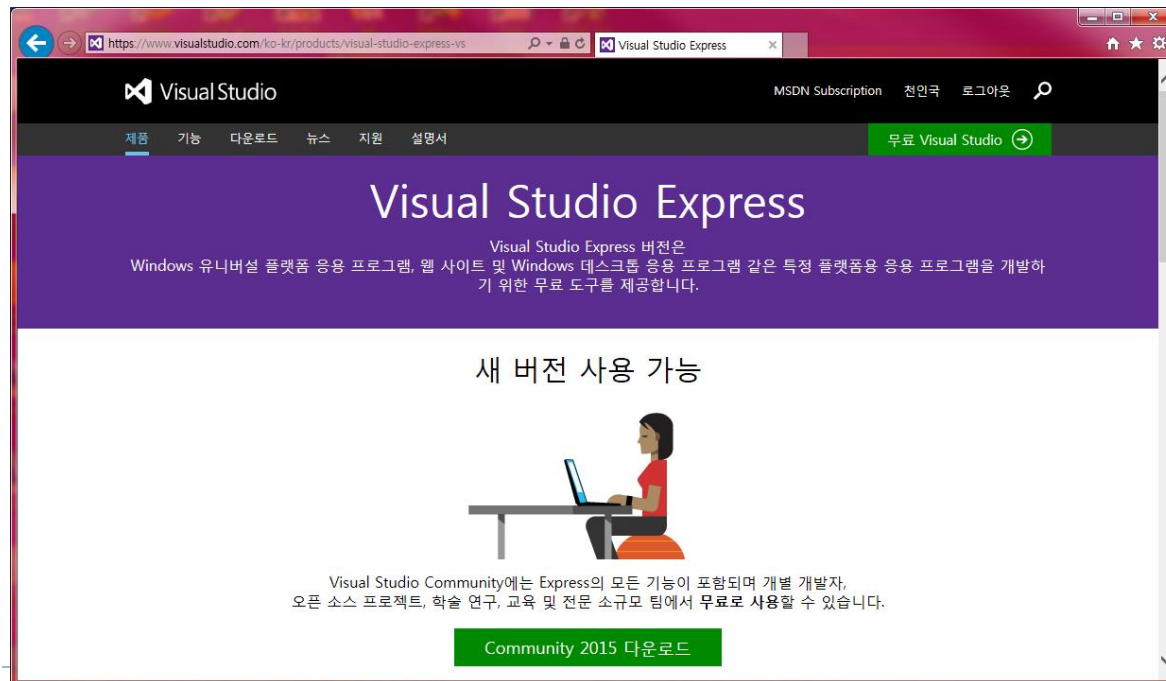
▶ 통합 개발 환경의 예

- Visual Studio
 - ▶ 마이크로소프트
- 이클립스(eclipse)
 - ▶ 오픈 소스 프로젝트
- Dev-C++
 - ▶ 오픈 소스 프로젝트



통합 개발 환경의 종류[1/3]

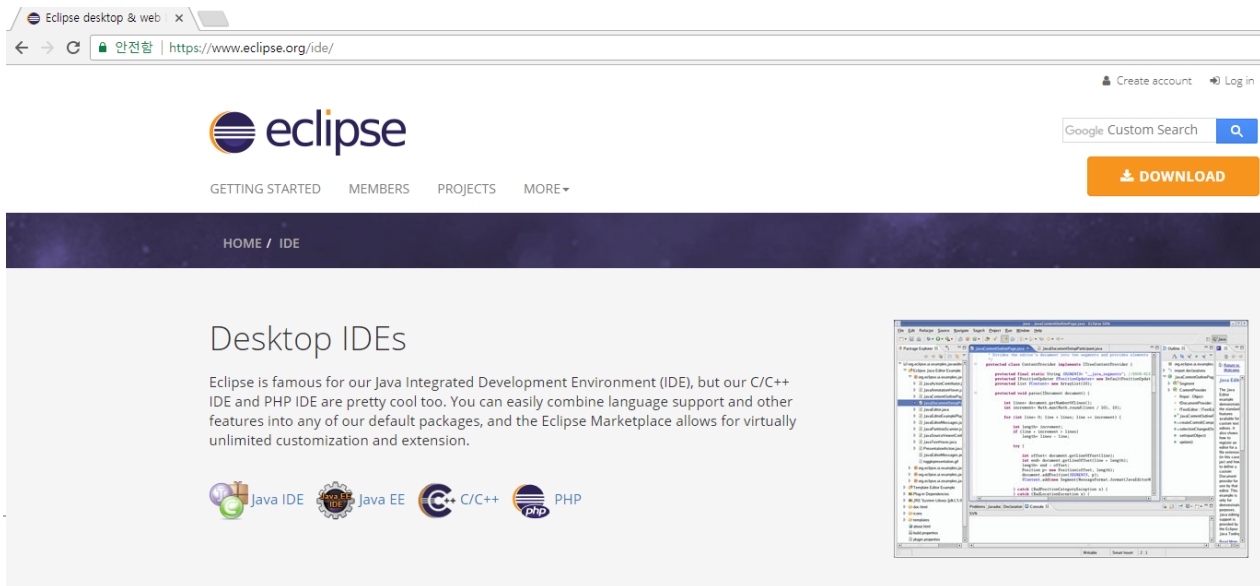
- ▶ 비주얼 스튜디오(Visual Studio)
 - 마이크로소프트사의 제품
 - 윈도우 기반의 거의 모든 형태의 응용 프로그램 제작 가능
 - 최신 버전: Visual Studio 2017
 - 우리가 사용할 버전: Visual Studio 2015



통합 개발 환경의 종류[2/3]

▶ 이클립스(Eclipse)

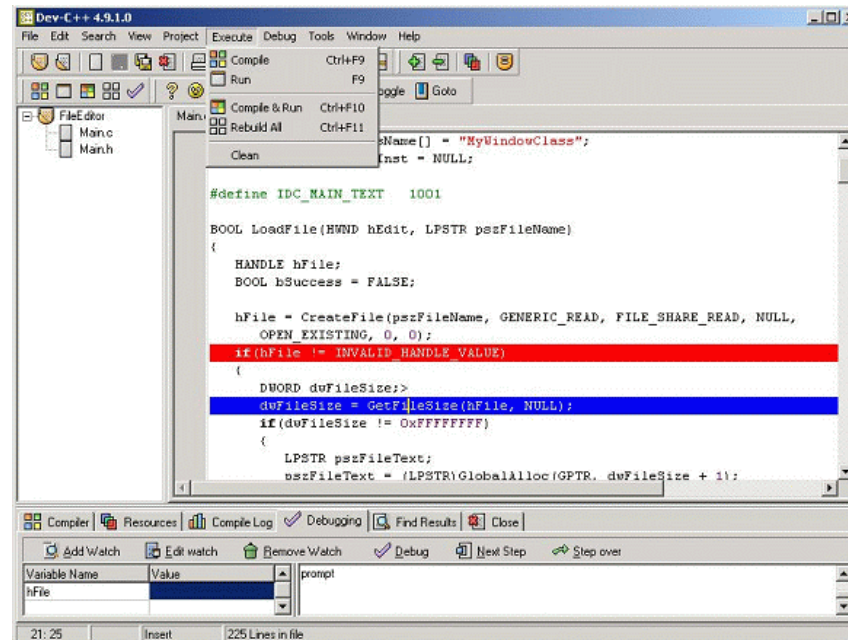
- 오픈 소스 프로젝트
- IBM을 비롯한 많은 회사들의 지원으로 개발
- 자바 언어를 위한 JDT(Java Development Tools) 지원
- C언어를 위한 CDT(C/C++ Development Tools) 지원
- 공개 컴파일러인 GNU의 gcc 컴파일러 사용
- 윈도우즈, 리눅스, 솔라리스 등의 다양한 플랫폼에서 사용



통합 개발 환경의 종류[3/3]

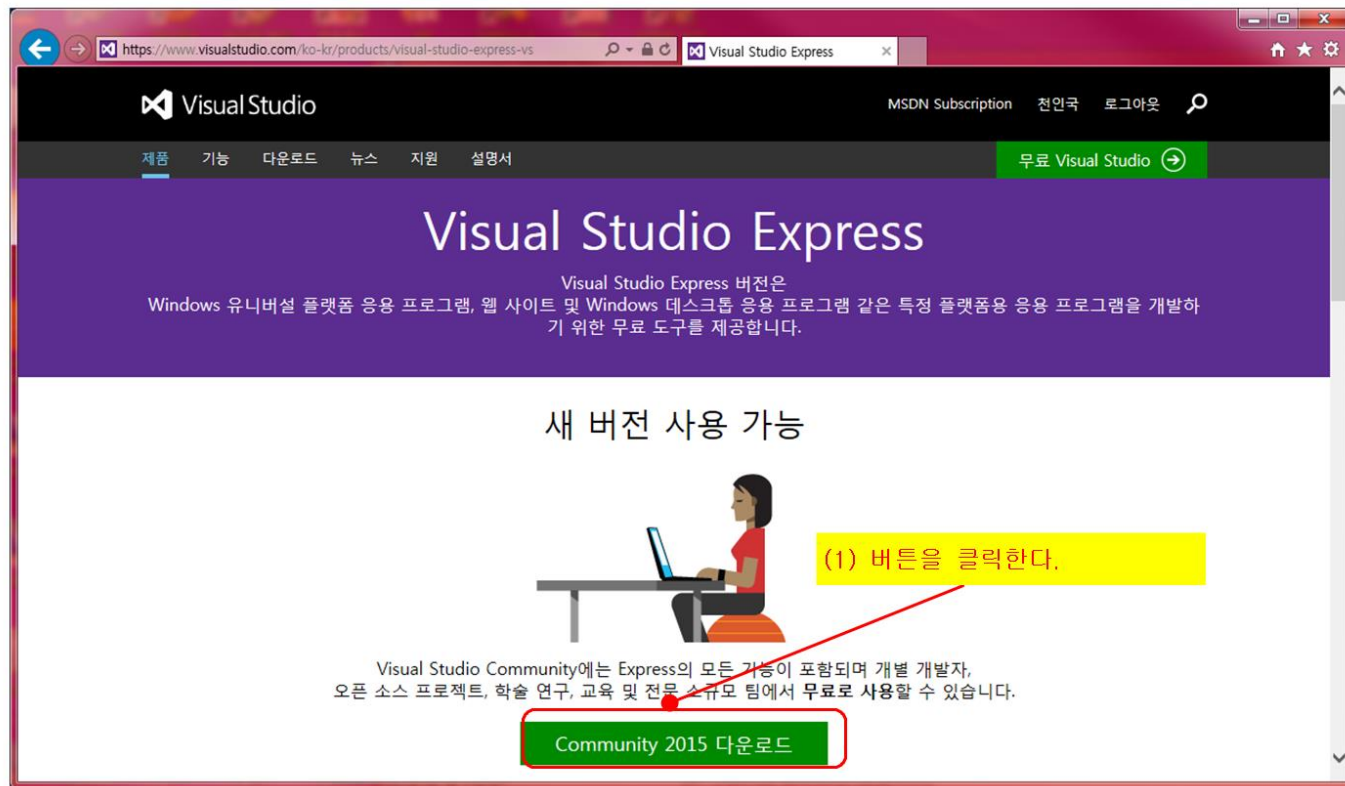
▶ Dev-C++

- 오픈 소스 프로젝트
- C/C++ 통합 개발 환경
- GCC 컴파일러 이용
- 무료



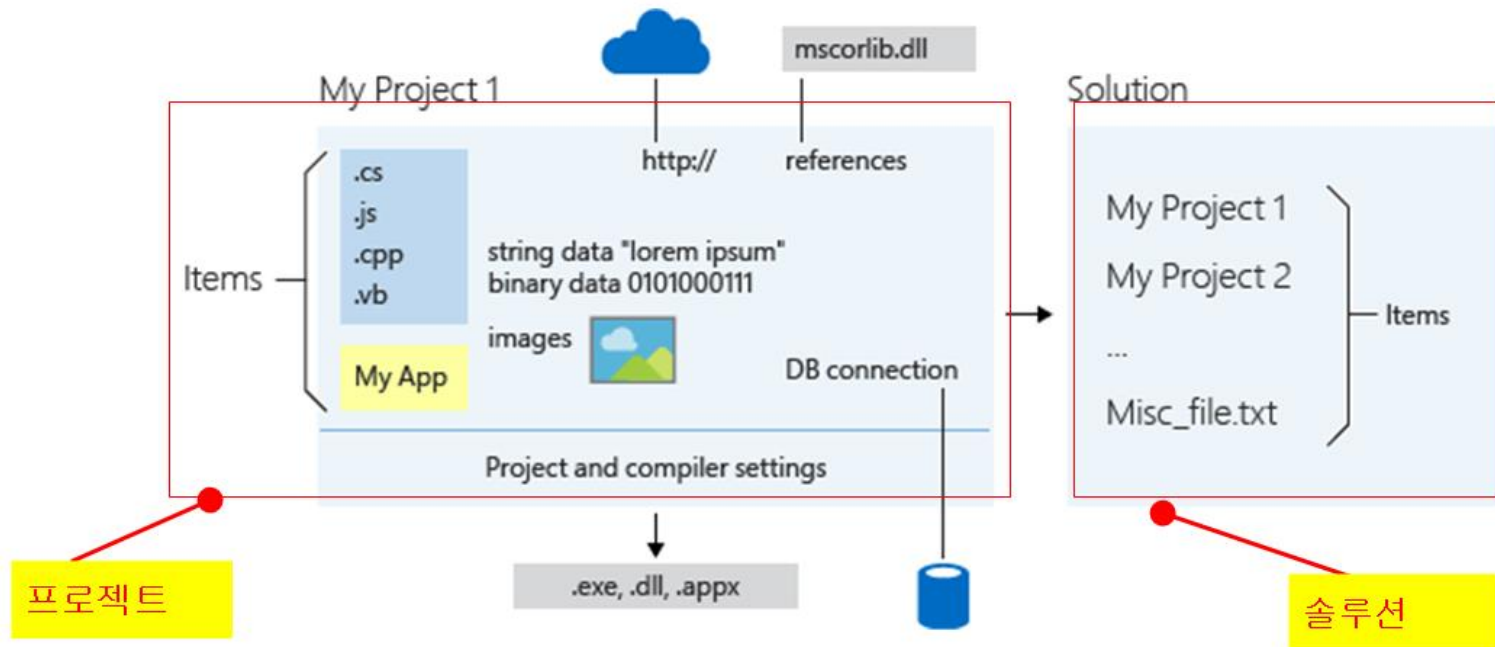
비주얼 스튜디오 설치

- ▶ <https://www.visualstudio.com/ko/vs/>
 - 설치 시 1시간 정도 소요!



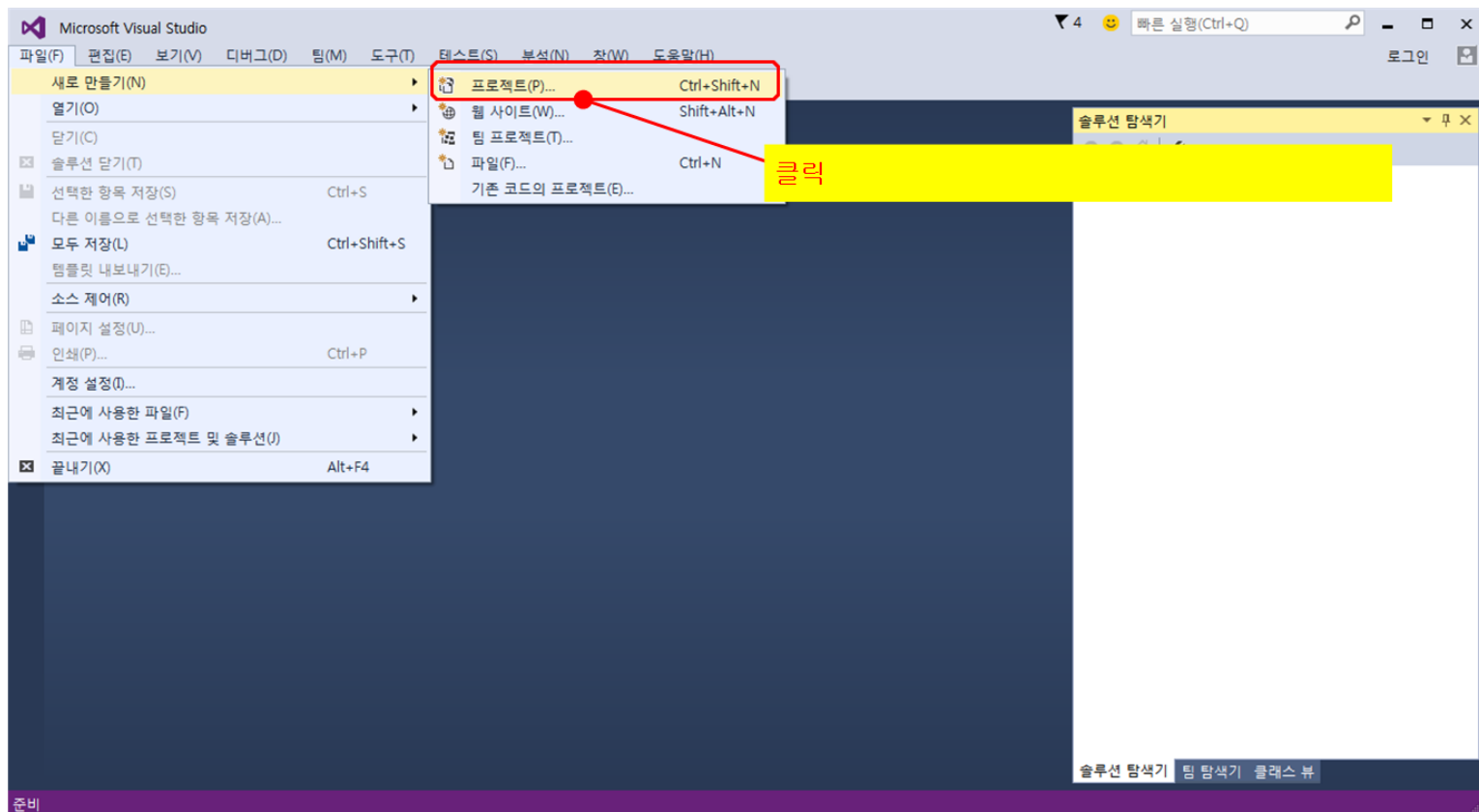
솔루션과 프로젝트

- ▶ 솔루션(Solution)
 - 문제 해결에 필요한 프로젝트가 들어 있는 컨테이너
- ▶ 프로젝트(Project)
 - 하나의 실행 파일을 만드는데 필요한 여러 항목들이 들어 있는 컨테이너

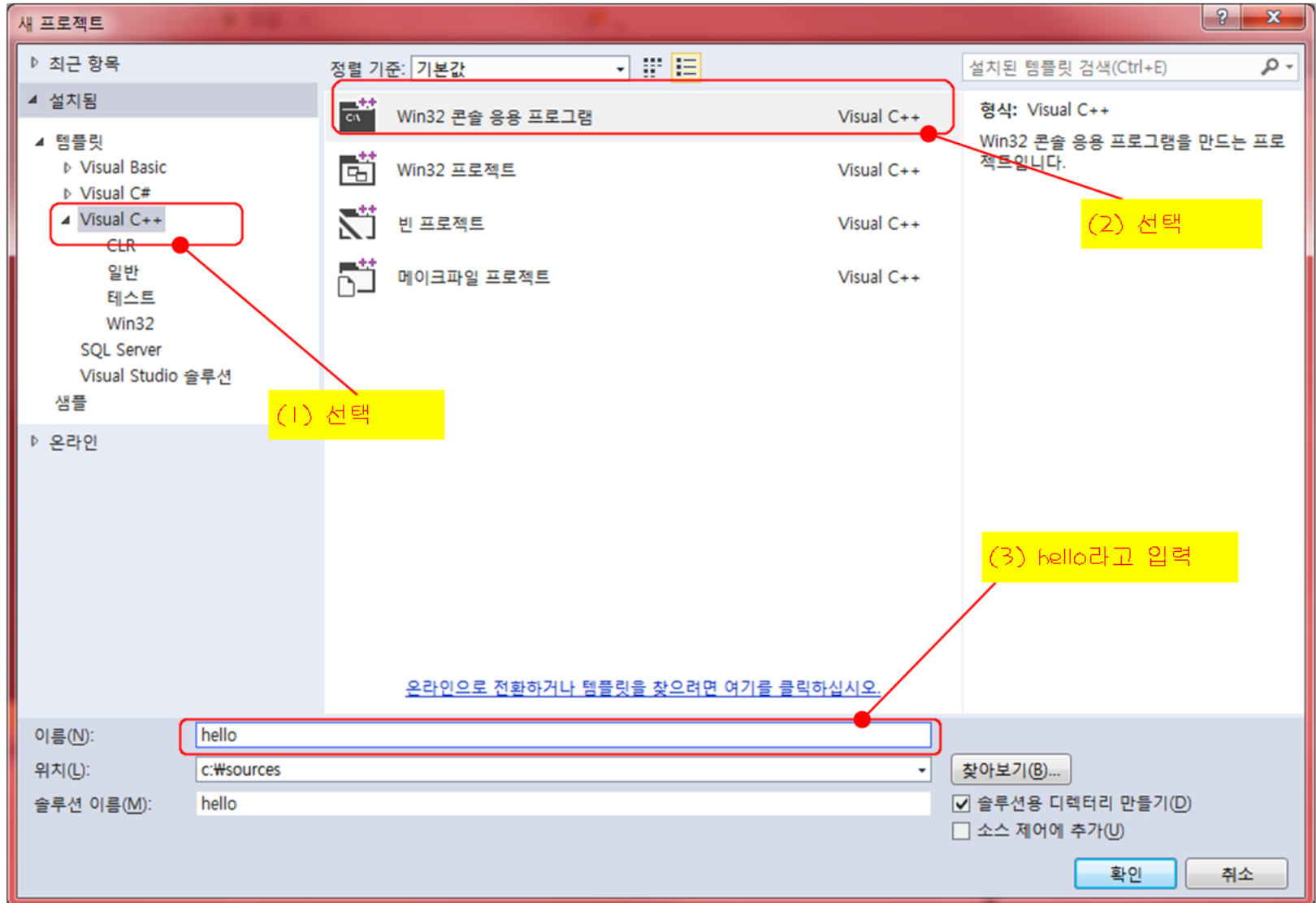


프로젝트 생성하기[1/3]

- ▶ [파일] → [새로 만들기] → [프로젝트]
 - 사용자가 새로운 프로젝트를 만들면 자동으로 솔루션 생성



프로젝트 생성하기[2/3]



프로젝트 생성하기[3/3]

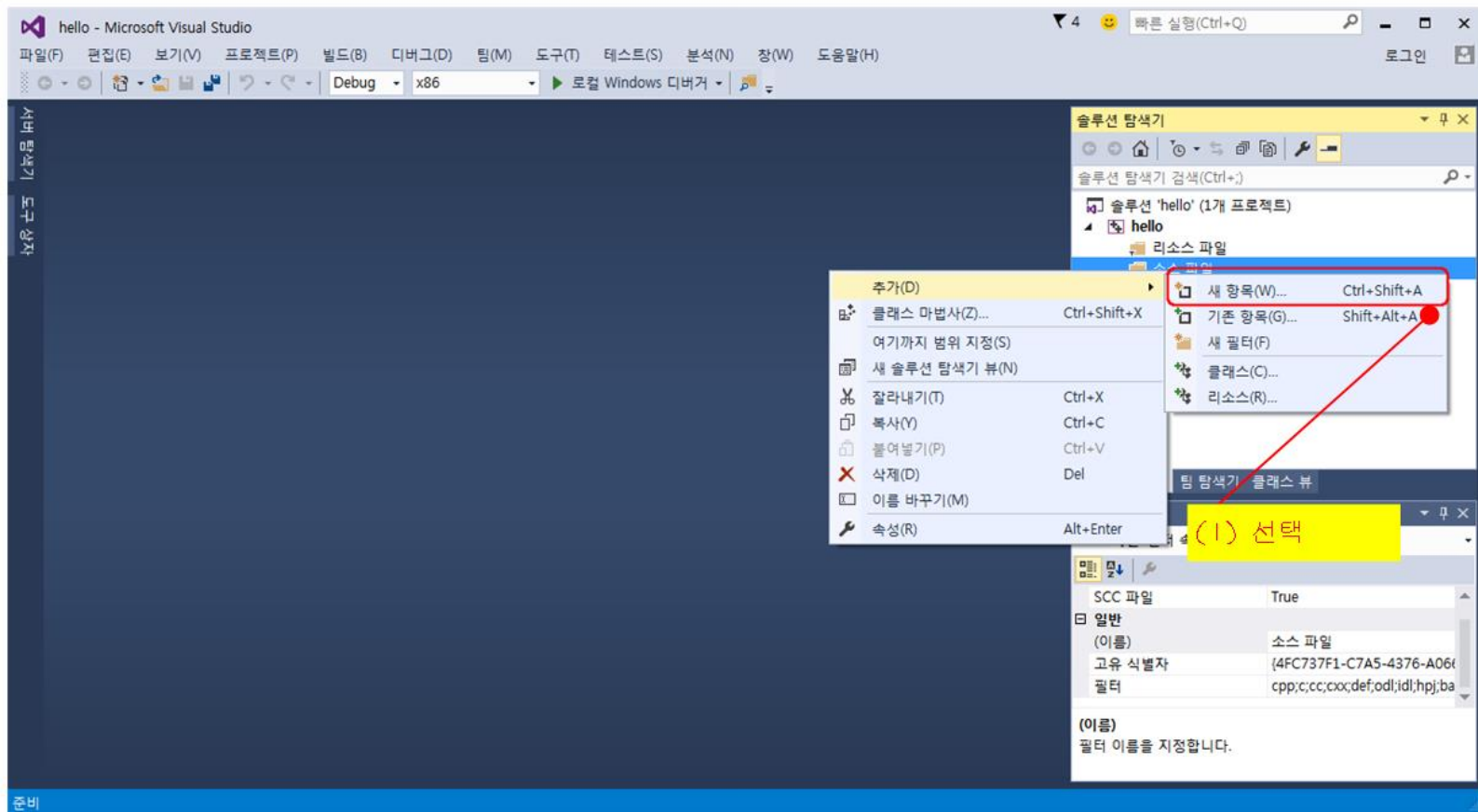
▶ SDL 검사 체크를 반드시 해제

- SDL 검사 체크가 해제되지 않으면 scanf() 함수 사용 시 오류



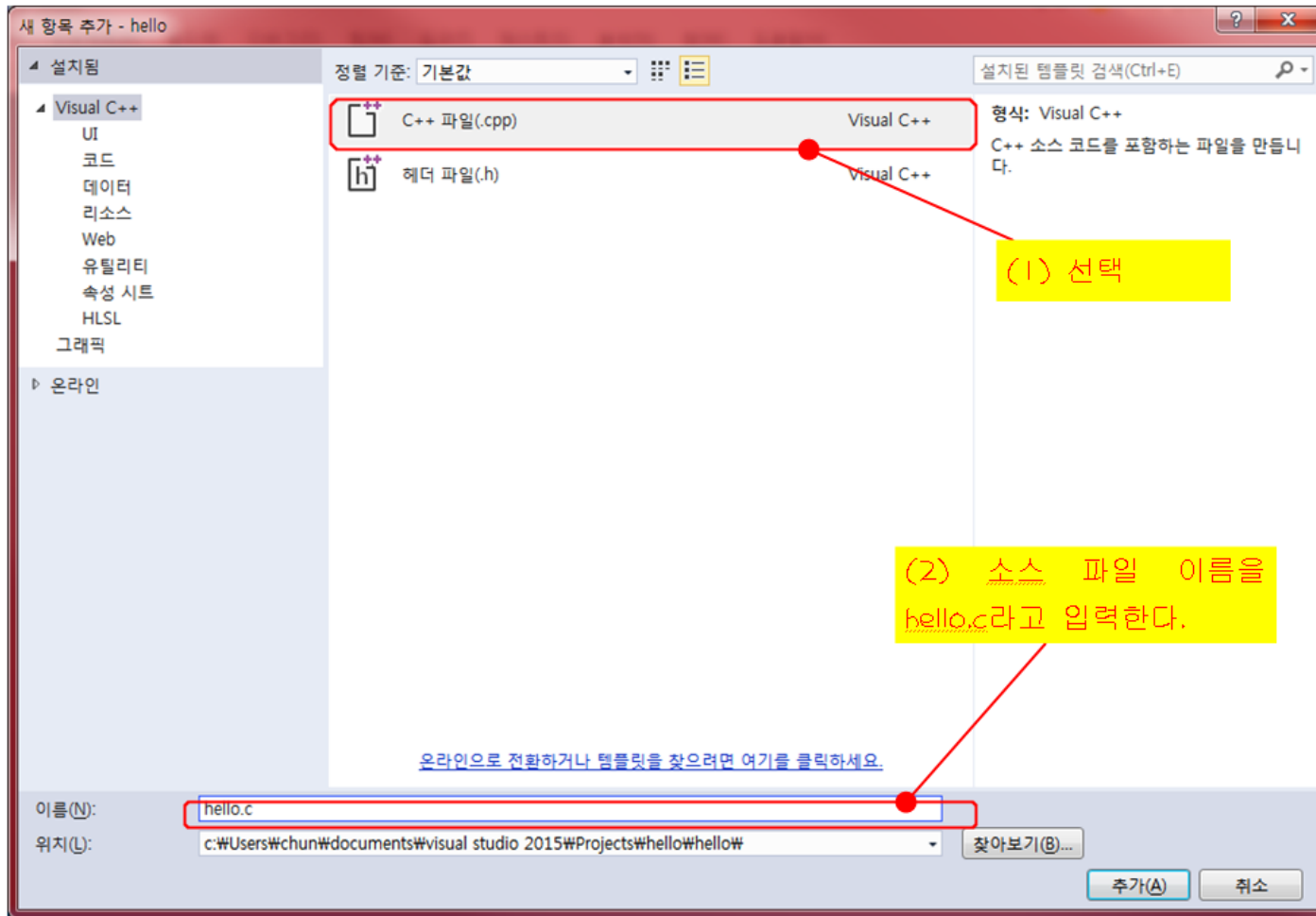
소스 파일 생성하기[1/2]

- ▶ 솔루션 탐색기의 [소스 파일]에서 마우스 우클릭
 - [추가] → [새 항목]



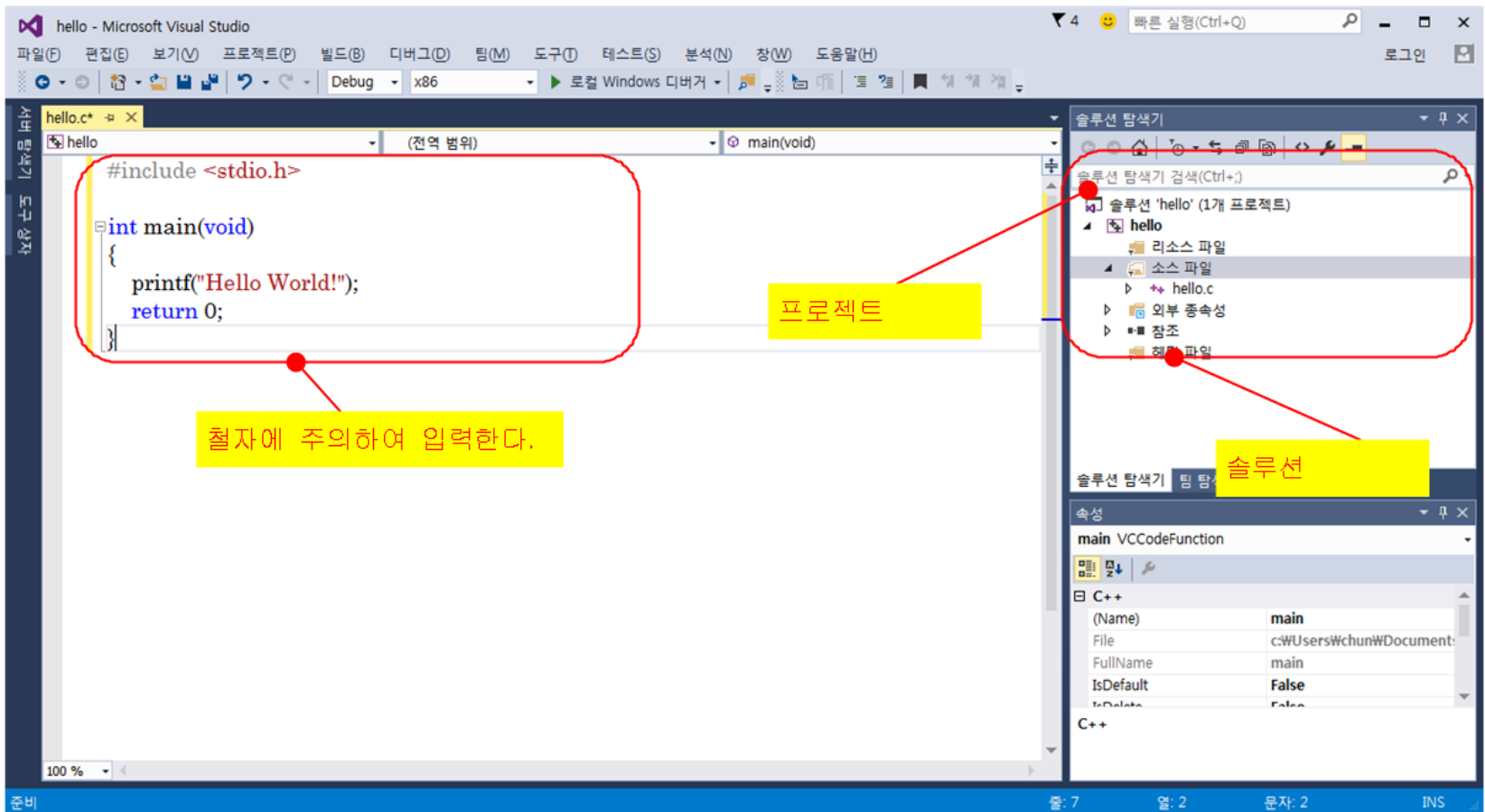
소스 파일 생성하기[2/2]

▶ 파일 이름의 확장자를 .c로 생성



프로그램 입력 [1/2]

- ▶ 솔루션 탐색기의 소스 파일 클릭
 - hello.c 더블 클릭하면 hello.c 작업 환경 진입



프로그램 입력[2/2]

- ▶ 교재 56쪽의 소스 코드 입력 주의 사항 정독!
- ▶ 에디터 화면의 단어들이 파란색, 붉은색, 검은색으로 나타남
 - 교재와 동일한지 비교

include나 stdio는 붙여쓴다.

```
# i n c l u d e    < s t d i o . h >
i n t    m a i n ( v o i d )
{
    p r i n t f ( " H e l l o    W o r l d ! " ) ;
    r e t u r n    0 ;
}
```

기호는 정확하게 입력

문장의 끝에는 세마콜

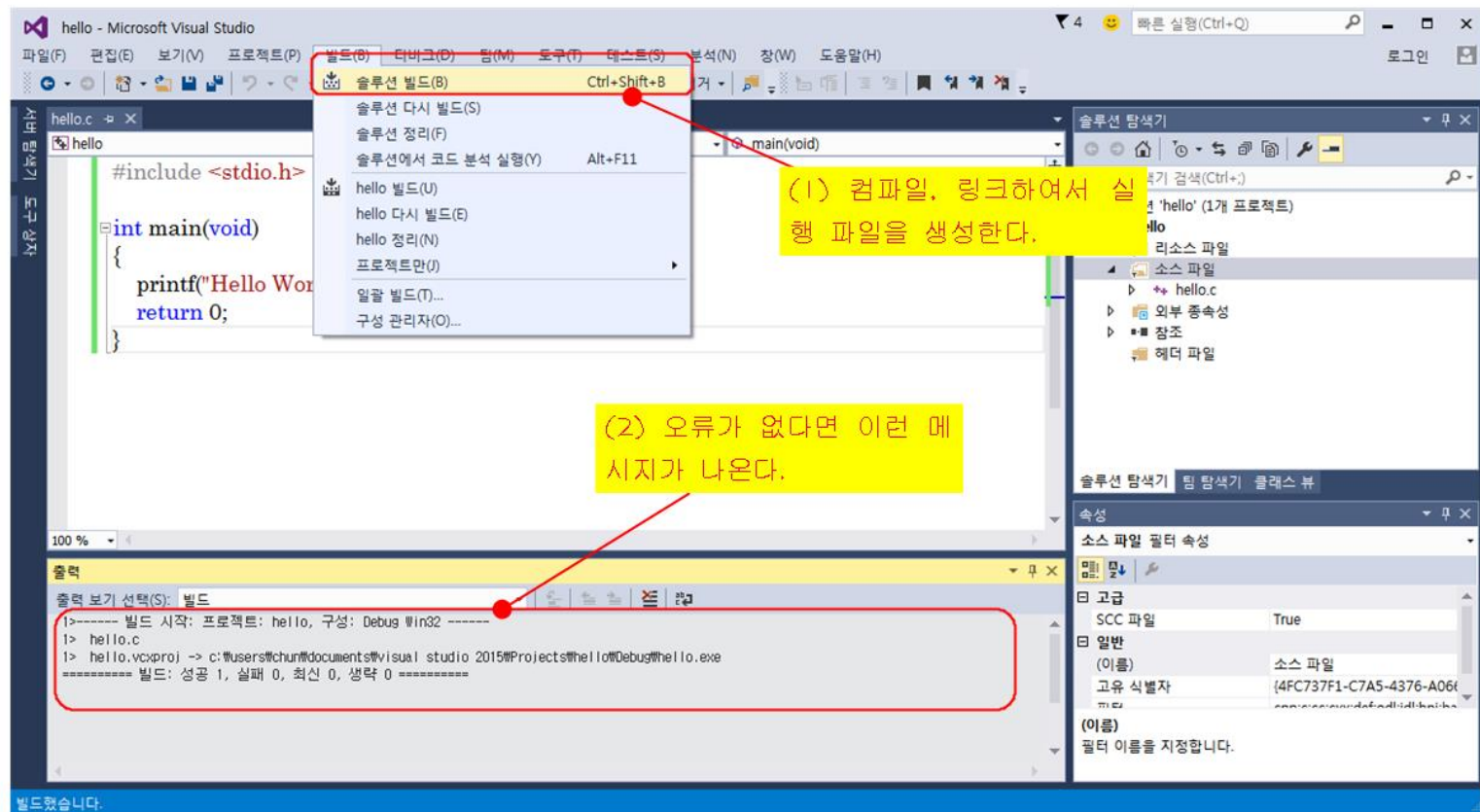
들여쓴다.

그림 2.3 소스 코드를 위와 같이 입력하여야 한다.

컴파일하기

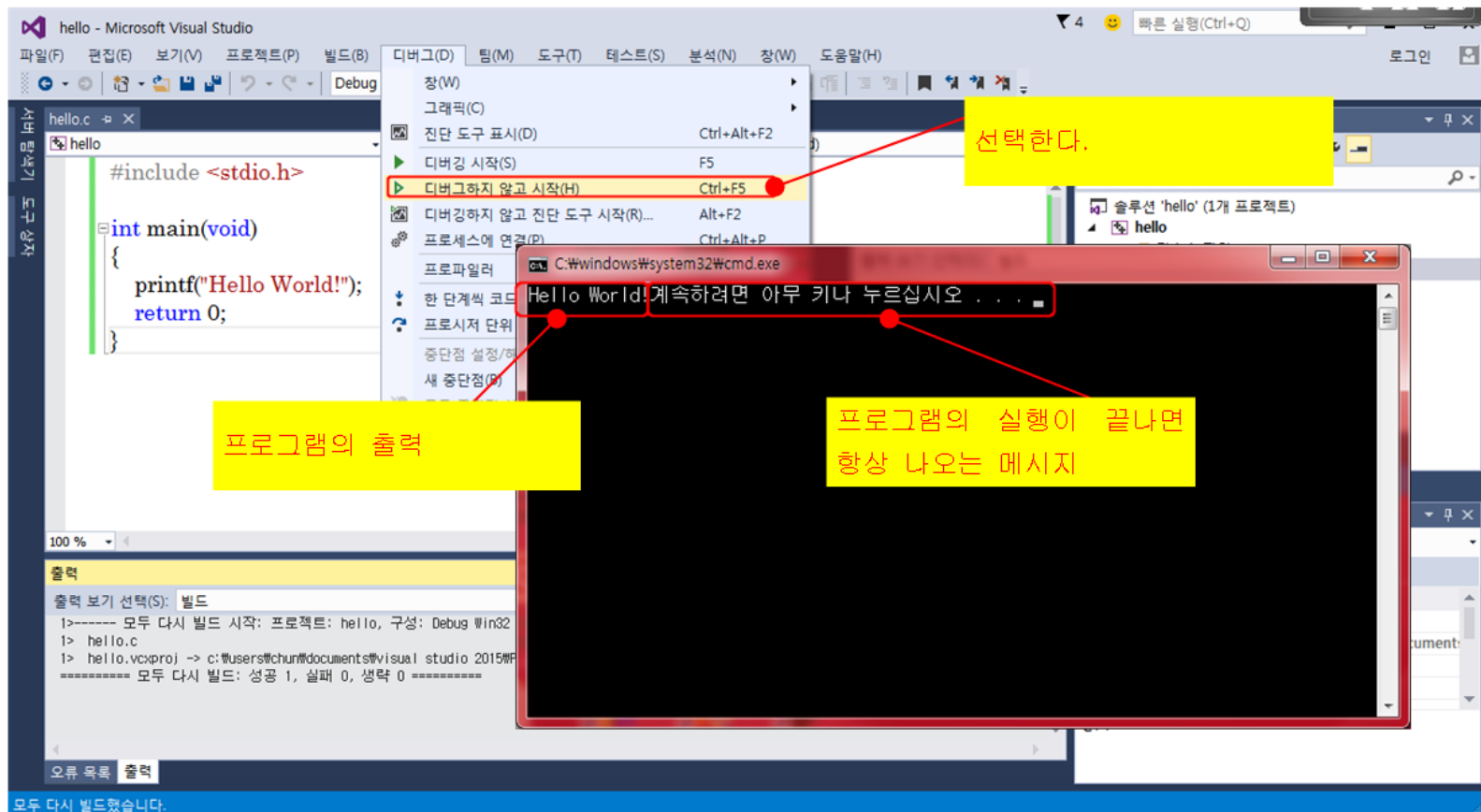
▶ 빌드(Build)

- 컴파일과 링크를 통해 완전한 실행 파일 생성
- [빌드] → [솔루션 빌드]



프로그램 실행 하기

- ▶ 작성한 프로그램 실행
 - [디버그] → [디버깅하지 않고 시작]





중간 점검

- ▶ 에디터, 컴파일러, 링커, 실행, 디버깅 등의 기능이 하나의 프로그램 안에 들어 있는 것을 무엇이라고 하는가?
- ▶ Visual C++에서 새로운 프로젝트를 생성하는 메뉴는 무엇인가?
- ▶ Visual C++에서 프로젝트에 속하는 소스 파일을 컴파일 하여 실행 파일을 생성하는 메뉴는?
- ▶ C 언어에서는 대문자와 소문자를 구별하는가?
- ▶ Visual C++를 이용하여서 sample.c라는 소스 파일을 컴파일하였을 때 생성되는 파일들은 무엇인가?
- ▶ Visual C++를 사용하여 소스 프로그램을 편집하는 경우, 메모장같은 다른 텍스트 에디터를 사용하여도 되는가?

첫 번째 프로그램의 설명

- ▶ 대표적인 C언어 입문 예제 프로그램
- ▶ 콘솔 화면에 "Hello World!"를 출력하는 프로그램

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

```
int main(void)
```

```
{
```

```
    printf("Hello World!");
```

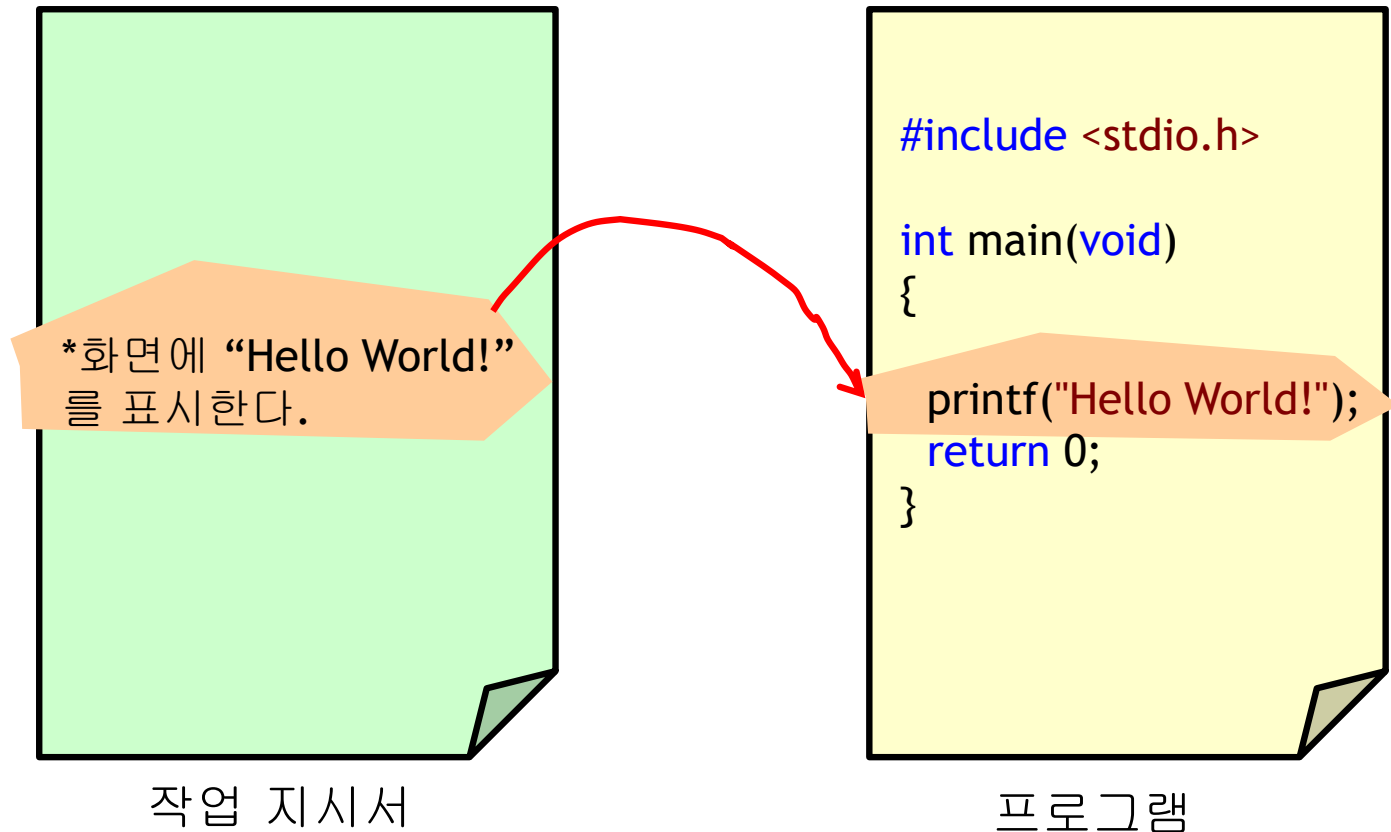
```
    return 0;
```

```
}
```



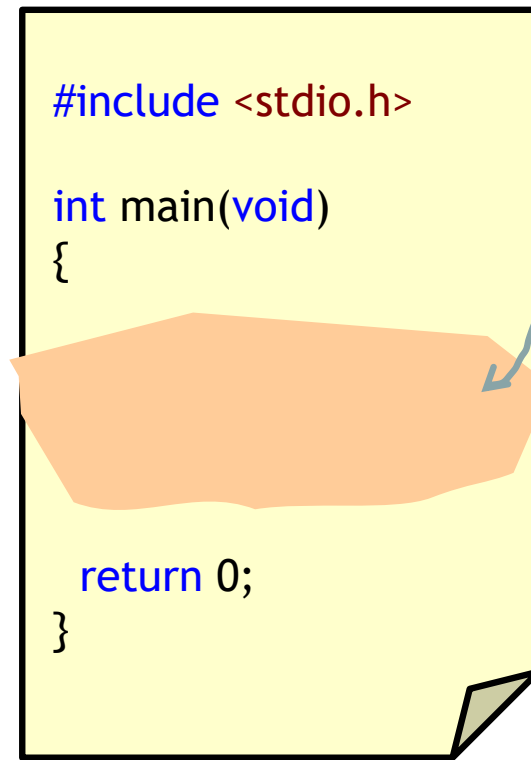
프로그램 == 작업 지시서

- ▶ 작업 지시 문장 이외의 다양한 문장들
 - C언어 기반 프로그램 구동을 위해 필수적인 문장



작업을 적어주는 위치

- ▶ 컴퓨터에게 C언어로 일(작업)을 시키기 위한 기본적인 틀
 - #include ...
 - main()



여기다가 원하는 작업을 수행하는 문장을 적어준다.

프로그램

C프로그램의 기본 구조



헤더 파일 포함

- ▶ #으로 시작하는 문장은 전처리기 지시자
 - 전처리기는 컴파일하기 전에 사전 작업을 수행
 - #include는 특정 파일을 현재 위치에 포함

• #include는 소스 코드 안에 특정 파일을 현재의 위치에 포함

• 주의!: 전처리기 지시자 문장 끝에는 세미콜론(;)을 붙이면 안 된다.

#include <stdio.h>

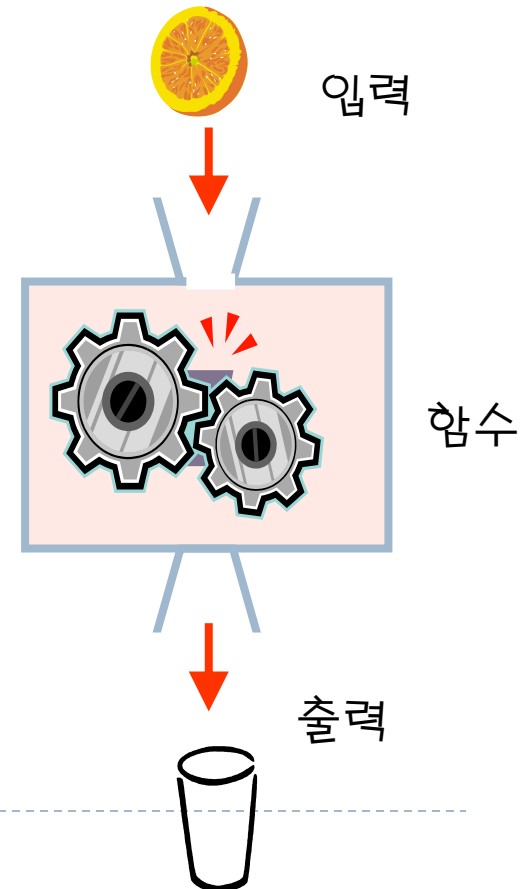
- 헤더 파일(header file): 컴파일러가 필요로 하는 정보를 가지고 있는 파일
- stdio.h: standard input output header file

함수

▶ 함수(function)

- 특정한 작업을 수행하기 위하여 작성된 독립적인 코드
- 입력이 주어지면 출력을 만들어내는 처리 수행
- (참고) 수학적 함수 $y = x^2 + 1$

▶ 프로그램 = 함수의 집합



main() 함수

- ▶ main() 함수는 C 프로그램에서 가장 먼저 실행
 - 프로그램의 시작점
 - C 프로그램에는 반드시 하나의 main() 함수만 존재

• 함수의 반환값

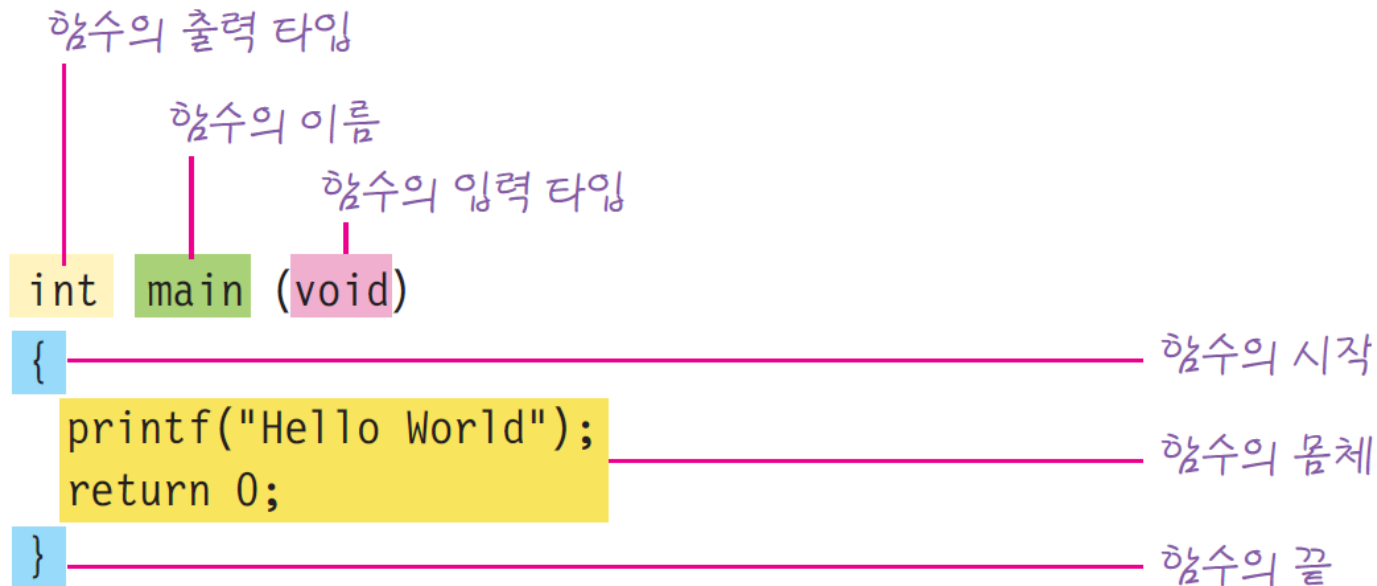
• 함수의 입력은 없음!

int main(void)

- 함수의 이름
- main()은 가장 먼저 수행되는 함수

함수 구조 설명

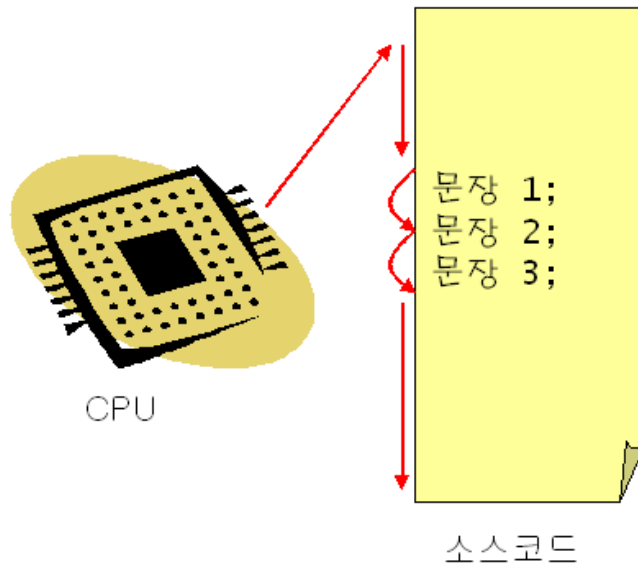
- ▶ int
 - 함수가 출력하는 값의 타입이 정수
- ▶ void는 비어있다는 의미
 - 함수의 입력이 없음을 의미
- ▶ 함수의 시작과 끝은 { 과 }로 나타냄
- ▶ 함수의 몸체에는 함수의 기능을 수행하는 문장들이 들어감



문장

- ▶ 함수는 여러 개의 문장으로 이루어짐
- ▶ 문장들은 순차적으로 실행
- ▶ 문장의 끝에는 반드시 ;이 있어야 함

I



printf() 호출

- ▶ printf() 함수를 호출하여 모니터 화면에 "Hello World!" 출력
 - 함수 호출은 이미 작성되어 있는 함수를 사용하는 것
 - 큰 따옴표 "" 안에 둘러싸인 텍스트를 문자열(string)이라고 함

- printf()는 컴파일러가 제공하는 함수로서 출력을 담당한다

printf("Hello World!");

- 큰따옴표 안의 문자열이 화면에 출력된다.



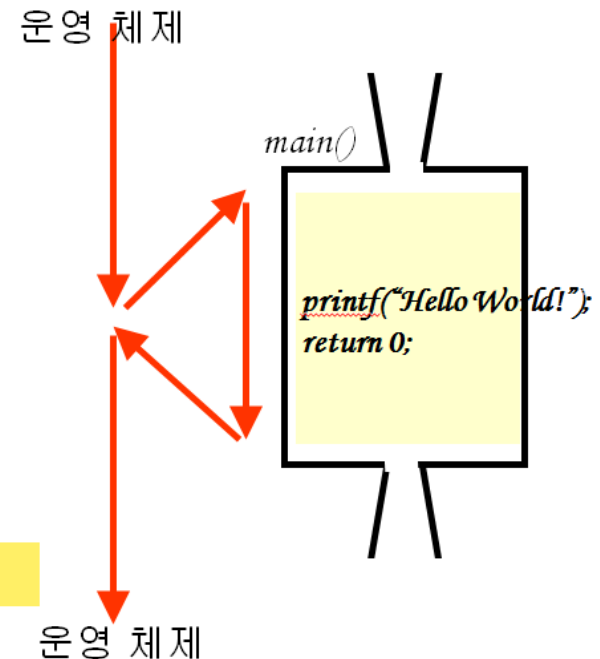
함수의 반환값

- ▶ return은 함수가 작업을 끝내고 작업 결과를 반환할 때 사용
- ▶ main()의 반환값은 프로그램을 실행시킨 운영체제에 반환하는 값
 - 보통 0의 값은 프로그램이 정상적으로 종료했음을 나타냄

- return은 함수의 결과값을 외부로 반환

return 0;

- 반환값은 0





중간 점검

- ▶ 문장의 끝에 추가하여야 하는 기호는?
- ▶ printf()가 하는 기능은 무엇인가?

응용 프로그램 #1

- ▶ 다음과 같은 출력을 가지는 프로그램 제작
- ▶ hello.c를 수정



첫 번째 버전

- ▶ 문장들은 순차적으로 실행된다는 사실 이용

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

```
int main(void)
```

```
{
```

```
    printf("Hello World!");
```

```
    printf("Kim ChulSoo");
```

```
    return 0;
```

```
}
```

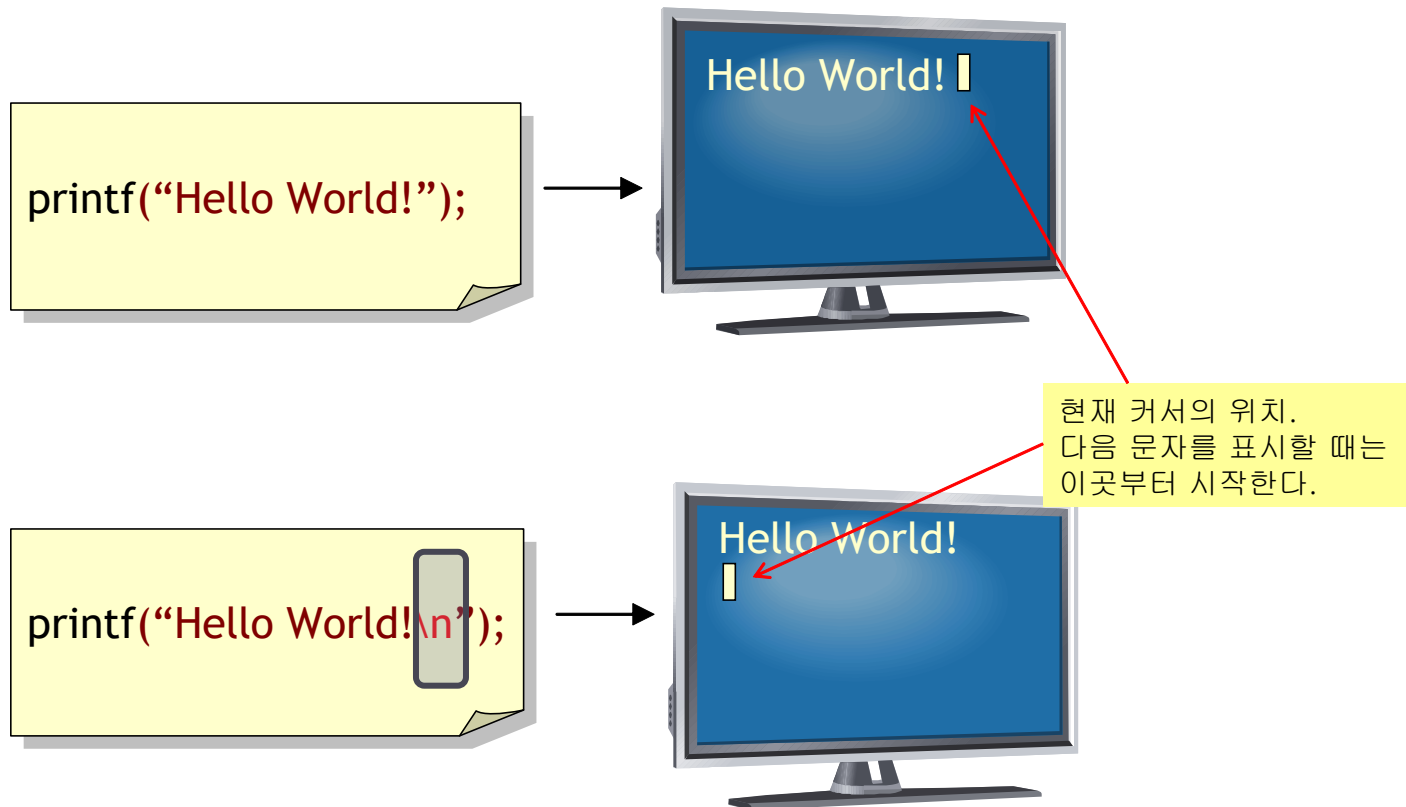
- 2개의 문장은 순차적으로 실행된다



Hello World! Kim ChulSoo

줄바꿈 문자 \n

- ▶ C언어에서 역슬래시(\)가 먼저 나타나면 특별한 의미가 있음
 - 줄바꿈 문자인 \n은 화면에서 커서는 다음 줄로 이동하게 함



줄바꿈 문자 2개를 사용하면?

```
printf("Hello \nWorld! \n");
```



변경된 프로그램

- ▶ 줄바꿈 문자를 포함하면 우리가 원하던 결과 도출

```
#include <stdio.h>

int main(void)
{

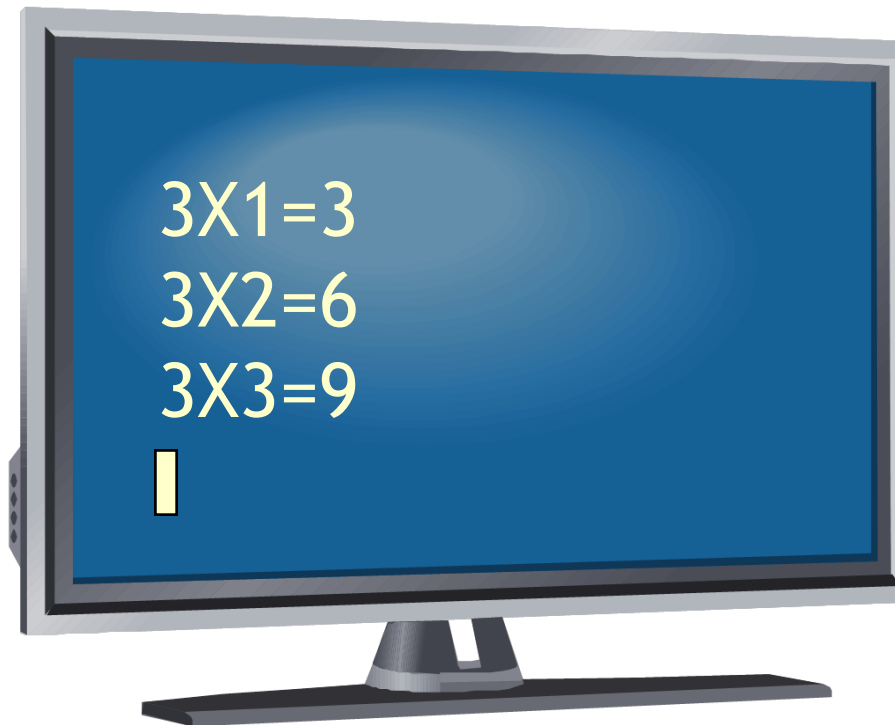
    printf("Hello World!\n");
    printf("Kim ChulSoo \n");

    return 0;
}
```



응용 프로그램 #2

- ▶ 다음과 같은 출력을 가지는 프로그램 제작



응용 프로그램

- ▶ 문장들은 순차적으로 수행된다는 점을 이용

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

```
int main(void)
```

```
{
```

```
    printf("3 X 1 = 3\n");
```

```
    printf("3 X 2 = 6\n");
```

```
    printf("3 X 3 = 9\n");
```

```
    return 0;
```

```
}
```

3개의 문장은 순차적으로 실행된다.



중간 점검

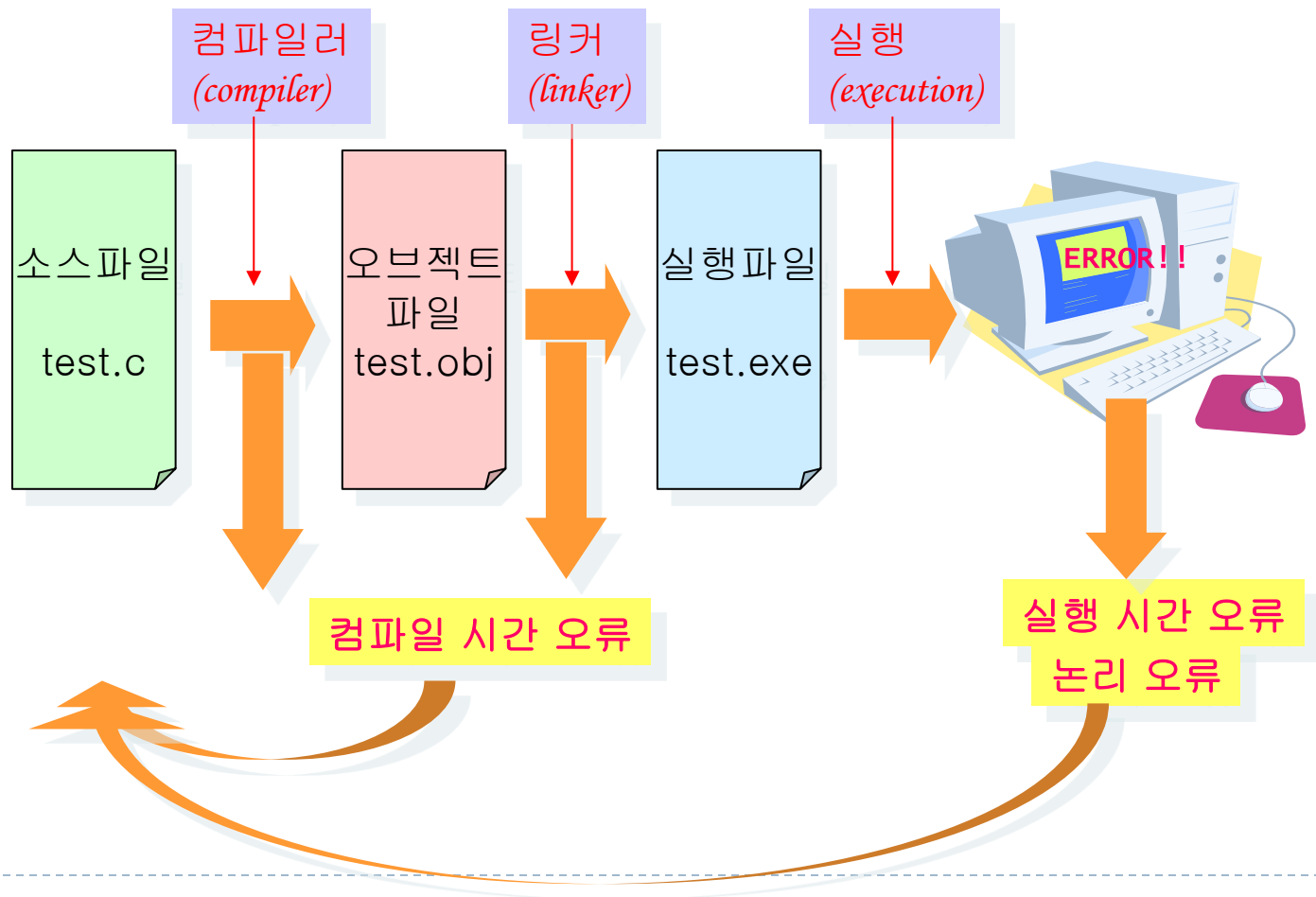
- ▶ 화면에 새로운 줄을 만드는데 사용되는 특수한 기호는?
- ▶ "사과", "오렌지", "포도"를 한 줄에 하나씩 출력하는 프로그램을 작성하여 보자.
- ▶ 구구단 3단 전체를 출력하는 프로그램을 작성하여 보자.

오류 수정 및 디버깅

- ▶ 컴파일이나 실행 시에 오류가 발생할 수 있음
- ▶ 에러와 경고
 - 에러(error): 심각한 오류
 - 경고(warning): 경미한 오류
- ▶ 오류의 종류
 - 컴파일 시간 오류
 - ▶ 대부분 문법적인 오류
 - 실행 시간 오류
 - ▶ 실행 중에 0으로 나누는 연산 같은 오류
 - 논리 오류
 - ▶ 논리적으로 잘못되어서 결과가 의도했던 대로 나오지 않는 오류



오류의 종류



컴파일 시간 오류[1/3]

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

```
int main(void)
```

```
{
```

```
    printf("Hello World!\n")
```

```
    return 0;
```

```
}
```

문장의 끝
에 ;이 없
음!!

오류가 발견된
소스파일

오류가 발견된 줄 번
호

return 앞에 ;을 빠뜨
렸다는 의미이다.

1>----- 모두 다시 빌드 시작: 프로젝트: hello, 구성: Debug Win32 -----

1> hello.c

1>c:\users\chun\documents\visual studio 2010\projects\hello\hello\hello.c(7): error C2143: 구문 오류 :
';'이(가) 'return' 앞에 없습니다.

===== 모두 다시 빌드: 성공 0, 실패 1, 생략 0 =====

컴파일 시간 오류[2/3]

```
/* 에러가 발생하는 프로그램 */  
#include <stdio.h>  
  
int main(void)  
{  
    printf("Hello World!\n")  
    return 0;  
}
```

*과 /이 떨어져 있음
-> 전체가 주석처리됨

```
I>----- 빌드 시작: 프로젝트: hello, 구성: Debug Win32 -----  
I> hello.c  
I>c:\users\chun\documents\visual studio 2010\projects\hello\hello\hello.c(9): fatal error  
C1071: 주석에서 예기치 않은 파일의 끝이 나타났습니다.  
===== 빌드: 성공 0, 실패 1, 최신 0, 생략 0 =====
```

주석은 프로그램에 대한 설명글로서 /* */ 안에 표시한다.

컴파일 시간 오류[3/3]

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

```
int main(void)
```

```
{
```

```
    print("Hello World!");
```

```
    return 0;
```

```
}
```

print가 아니
란 printf임



```
I>----- 빌드 시작: 프로젝트: hello, 구성: Debug Win32 -----
```

```
I> hello.c
```

```
I>c:\users\chun\documents\visual studio 2010\projects\hello\hello\hello.c(6): warning  
C4013: 'print'이(가) 정의되지 않았습니다. extern은 int형을 반환하는 것으로 간주합니  
다.
```

```
I>hello.obj : error LNK2019: _print 외부 기호(참조 위치: _main 함수)에서 확인하지 못  
했습니다.
```

```
I>c:\Users\chun\Documents\Visual Studio 2010\Projects\hello\Debug\hello.exe : fatal error  
LNK1120: 1개의 확인할 수 없는 외부 참조입니다.
```

```
===== 빌드: 성공 0, 실패 1, 최신 0, 생략 0 =====
```

논리 오류[1/3]

- ▶ 다음과 같은 출력을 가지는 프로그램을 작성



논리 오류[2/3]

- ▶ 논리 오류가 존재하는 프로그램

```
#include <stdio.h>

int main(void)
{
    printf("Hey!");
    printf("Good Morning");
    return 0;
}
```

줄이 바뀌지
않았음!

Hey!Good Morning
|

논리 오류[3/3]

- ▶ 논리 오류가 수정된 프로그램

```
#include <stdio.h>

int main(void)
{
    printf("Hey! \n");
    printf("Good Morning \n");
    return 0;
}
```

논리 오류
수정!!



디버깅

▶ 논리 오류를 찾는 과정

- 수사관들이 범죄 현장을 분석하여 범인을 찾는 것과 같음
- 실행 결과를 보고 원인을 분석

아무래도 이
부분이 수상
해..



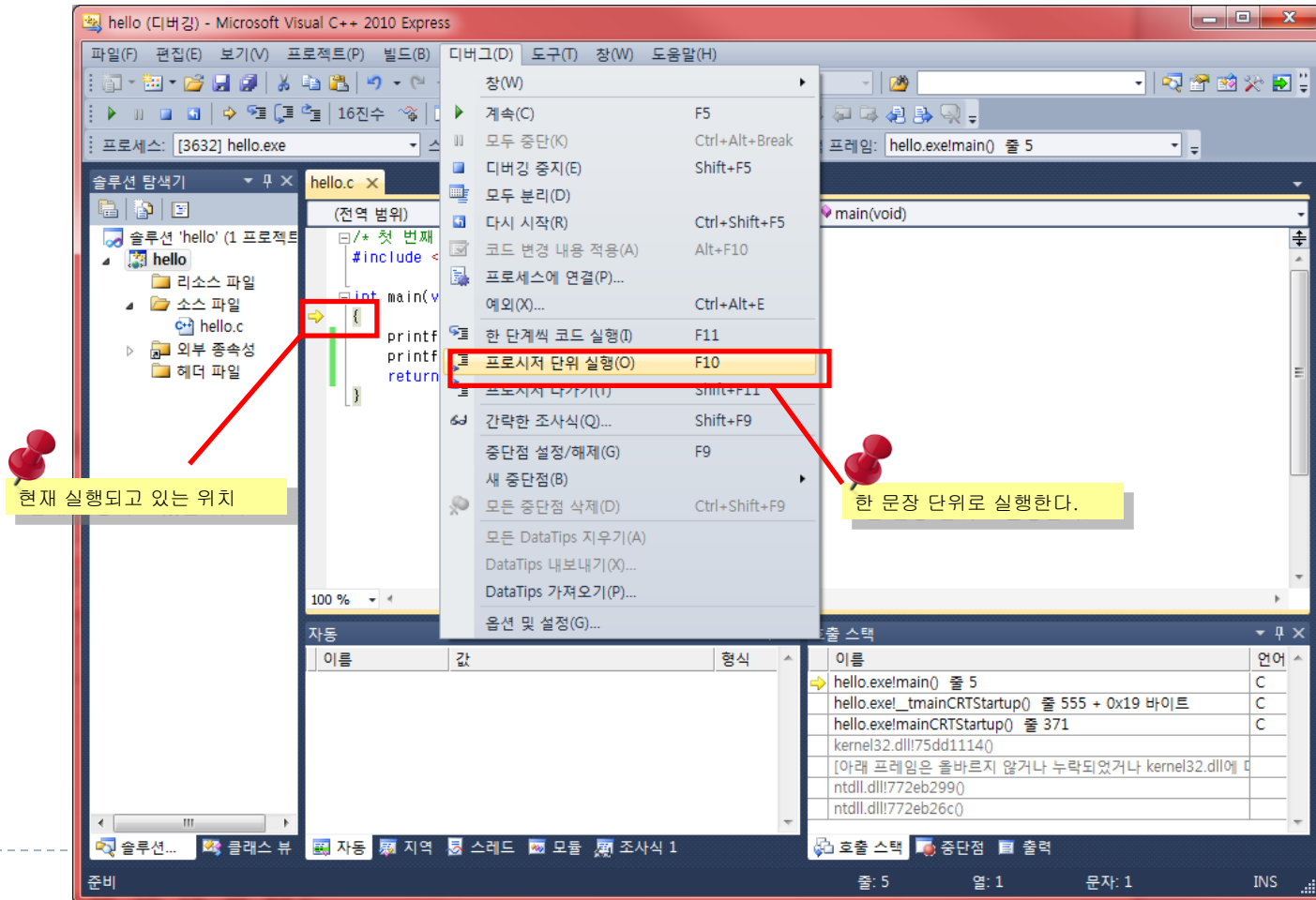
논리 에러를 발견
하는 것은 수사관
이 범죄 흔적을
이용하여 범인을
찾는 것과 같습니
다.



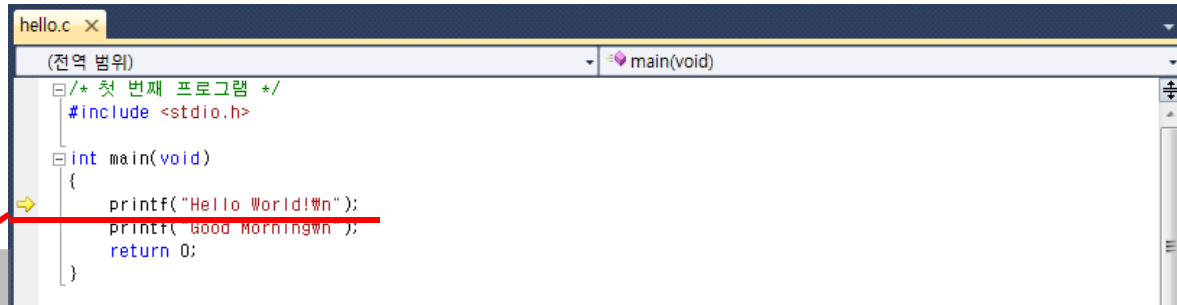
프로그램의
실행결과

디버거(debugger)

- ▶ 디버깅 지원 도구
- ▶ [디버그] → [프로시저 단위 실행]

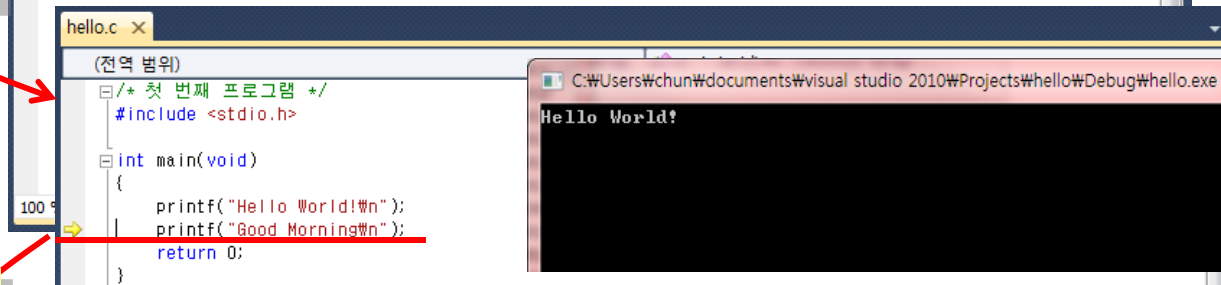


디버거 실행 과정



```
hello.c x
(전역 범위) main(void)
/* 첫 번째 프로그램 */
#include <stdio.h>

int main(void)
{
    printf("Hello World!\n");
    printf("Good Morning\n");
    return 0;
}
```

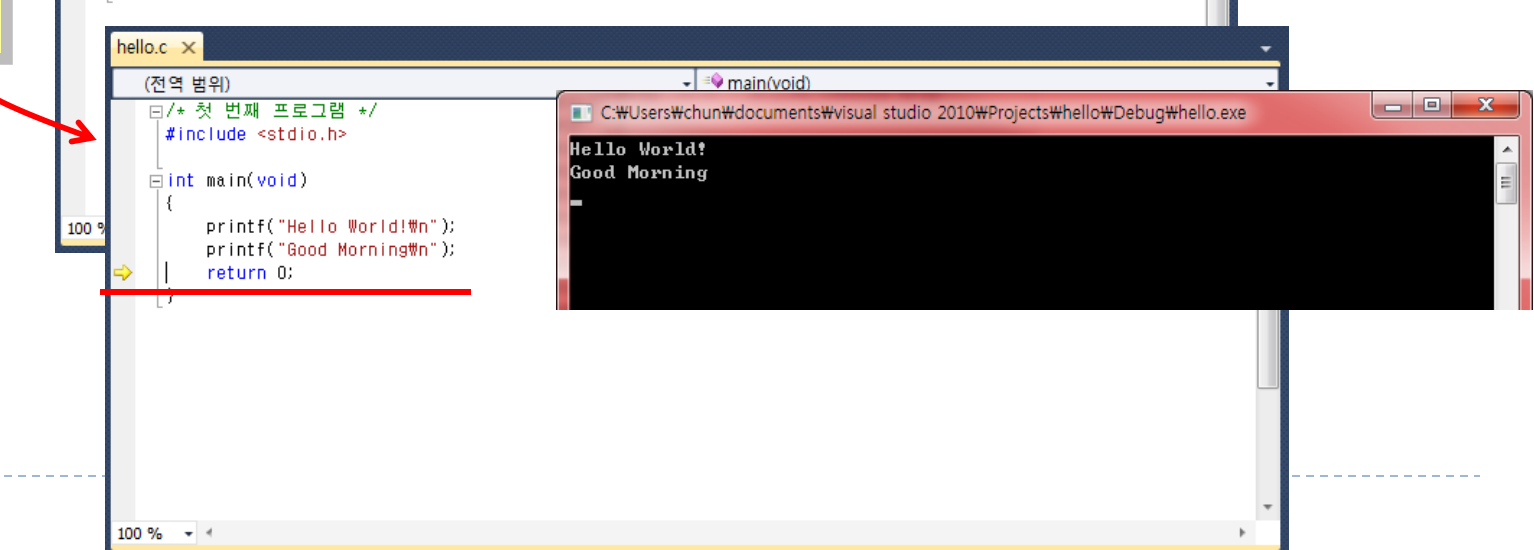


```
hello.c x
(전역 범위) main(void)
/* 첫 번째 프로그램 */
#include <stdio.h>

int main(void)
{
    printf("Hello World!\n");
    printf("Good Morning\n");
    return 0;
}
```

C:\Users\chun\documents\visual studio 2010\Projects\hello\Debug\hello.exe

Hello World!



```
hello.c x
(전역 범위) main(void)
/* 첫 번째 프로그램 */
#include <stdio.h>

int main(void)
{
    printf("Hello World!\n");
    printf("Good Morning\n");
    return 0;
}
```

C:\Users\chun\documents\visual studio 2010\Projects\hello\Debug\hello.exe

Hello World!
Good Morning

디버거 명령어 정의

- ▶ F5 (Go)
 - 실행
- ▶ F10 (Step Over)
 - 한 문장씩 실행(함수도 하나의 문장 취급)
- ▶ F11 (Step Into)
 - 한 문장씩 실행(함수 안으로 진입)
- ▶ F9 (Breakpoint)
 - 현재 문장에 중단점을 설정



중간 점검

- ▶ 프로그램을 편집하여 컴파일, 링크를 한 다음, 실행시켰는데 자신이 기대한 대로 결과가 나오지 않았다. 이때는 어떻게 하여야 하는가?
- ▶ 비교적 경미한 오류를 무엇이라고 하는가?

Q & A

