


컴퓨터프로그래밍의기초

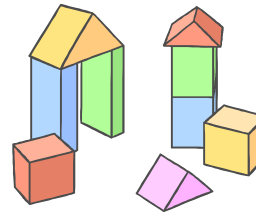
CLASS 2.

CHAPTER 02 프로그램 작성 과정



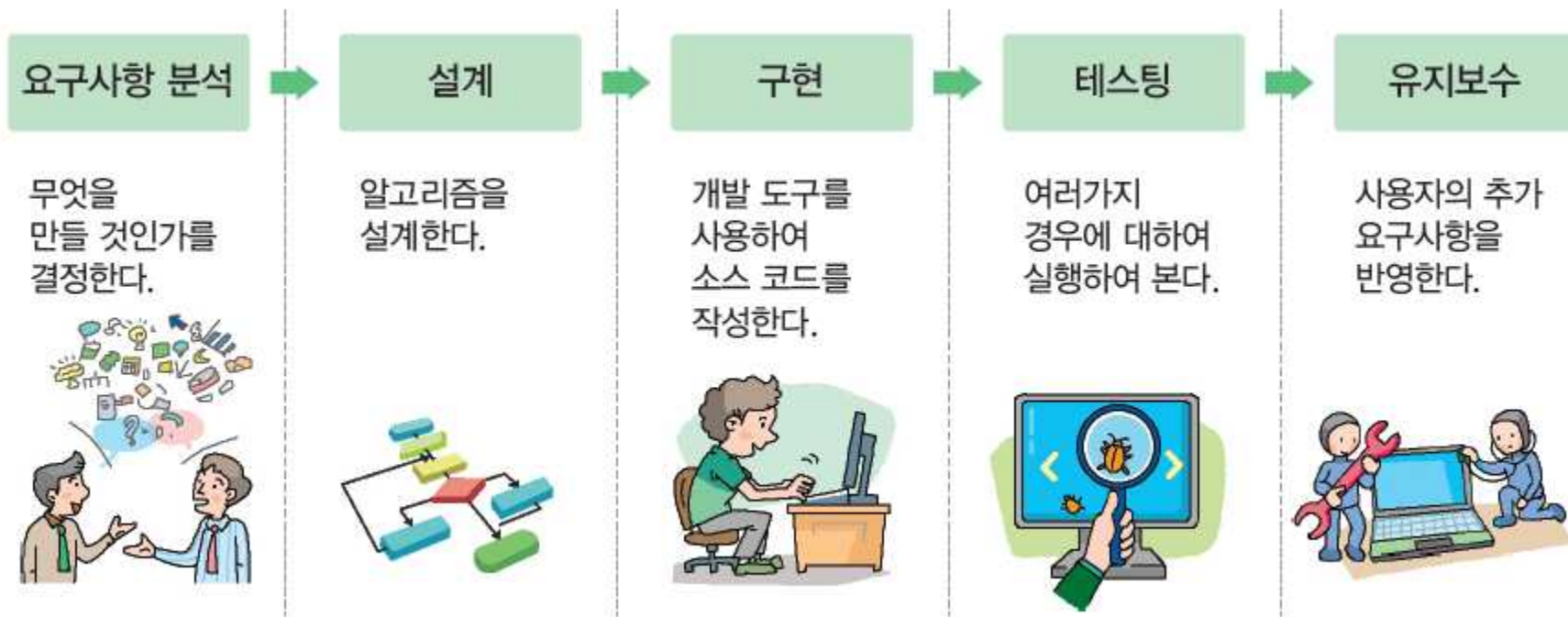
이번 장에서 학습할 내용

- 
- * 프로그램 개발 과정
 - * 컴파일 단계
 - * 개발환경 구축



이번 장에서는
C 프로그램 개발
과정을
살펴봅니다.

프로그램 개발 과정



소스 작성 - source

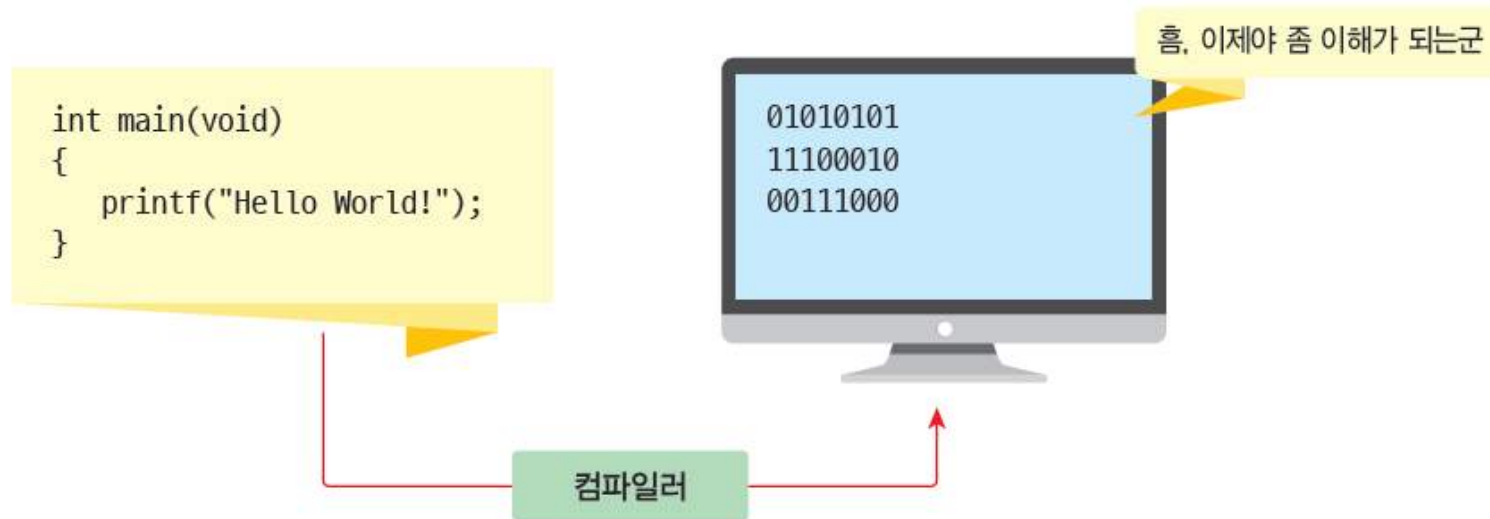
- 소스 source 작성
 - 프로그램은 주로 텍스트 에디터나
 - 통합 개발 환경을 이용하여 작성.
- 소스 파일 이름: (예) test.c
- 통합 개발 환경 (Integrated Development Environment, IDE)
 - Visual studio (C, C++), Eclipse (Java, Android)



```
int main(void)
{
    printf("Hello World!");
}
```

컴파일 - compile *****

- 컴파일: 소스 프로그램을 오브젝트 파일(기계어)로 변환하는 작업
 - 전처리+컴파일+어셈블
- 컴파일은 컴파일러에 의해 수행됨 (C언어: gcc)
- 오브젝트 파일 이름: (예) test.o ; test.obj



오브젝트 파일 – OBJECT *****

- 오브젝트 object 파일, 목적 파일 : test.o
 - 컴파일이 끝나 기계어로 변환된 파일을 오브젝트 파일(object file)

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

```
int main(void)
```

```
{
```

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

```
    return 0;
```

```
}
```

소스 파일



```
010101000001111101010
```

```
101010100000111110101
```

```
010101010000011111010
```

```
101010101000001111101
```

```
010101010100000111110
```

```
101001010100000111110
```

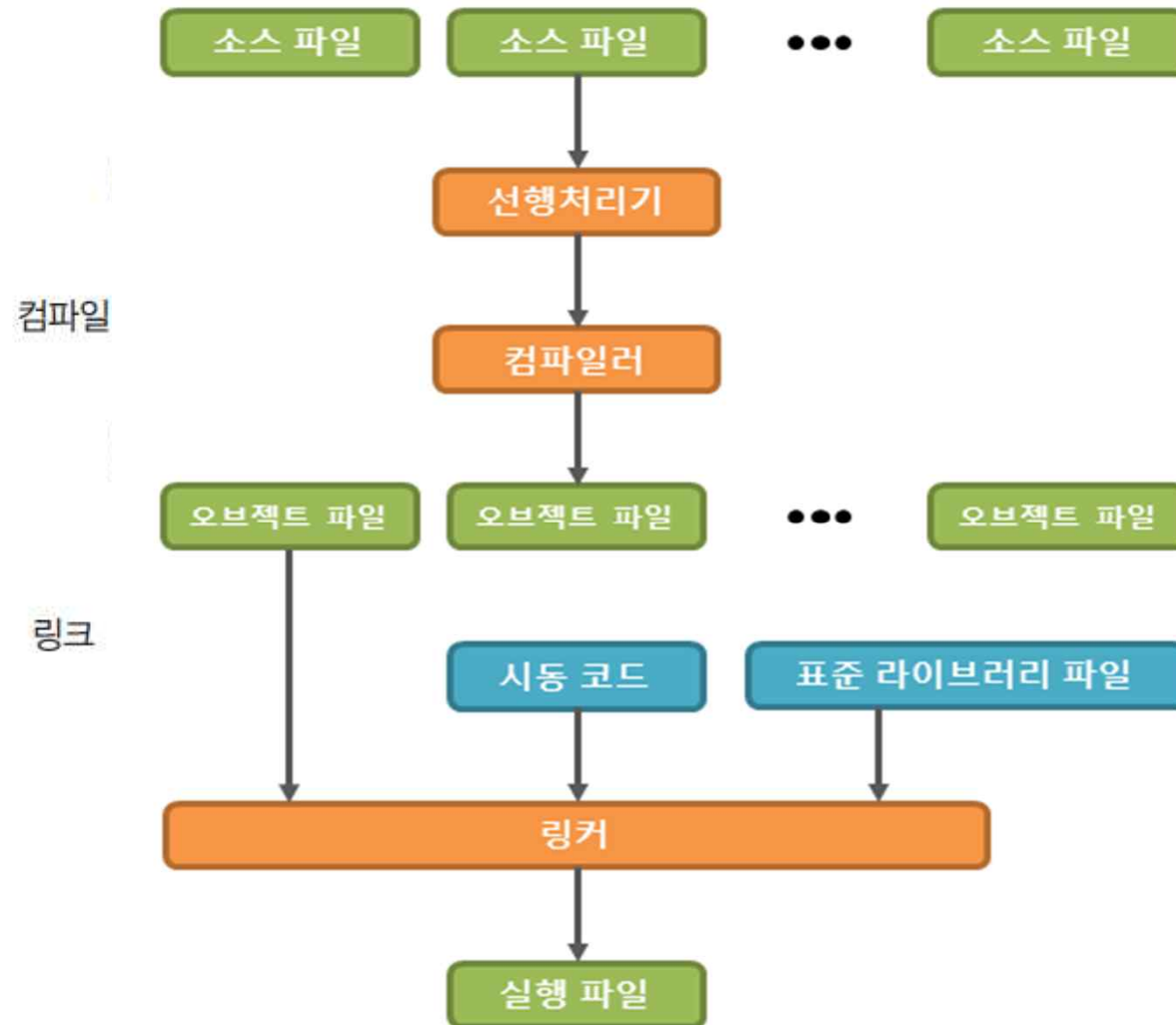
```
10101...
```

오브젝트 파일

링크 – link *****

- 컴파일된 오브젝트(목적) 파일을 라이브러리와 연결하여
- 실행 프로그램을 만드는 것
 - 실행 파일 이름: (예) test.exe
- 라이브러리(library)
- 프로그래머들이 많이 사용되는 함수를 미리 작성해 놓은 것
 - (예) 입출력 기능, 파일 처리, 수학 함수 계산
- 링크를 수행하는 프로그램을 링커(linker)라고 한다.

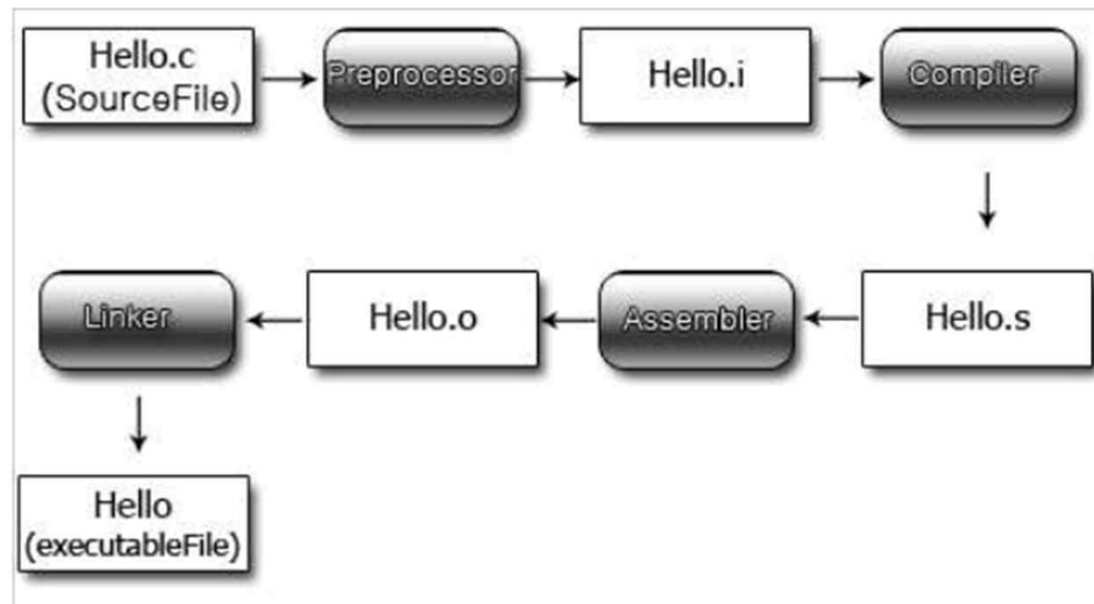
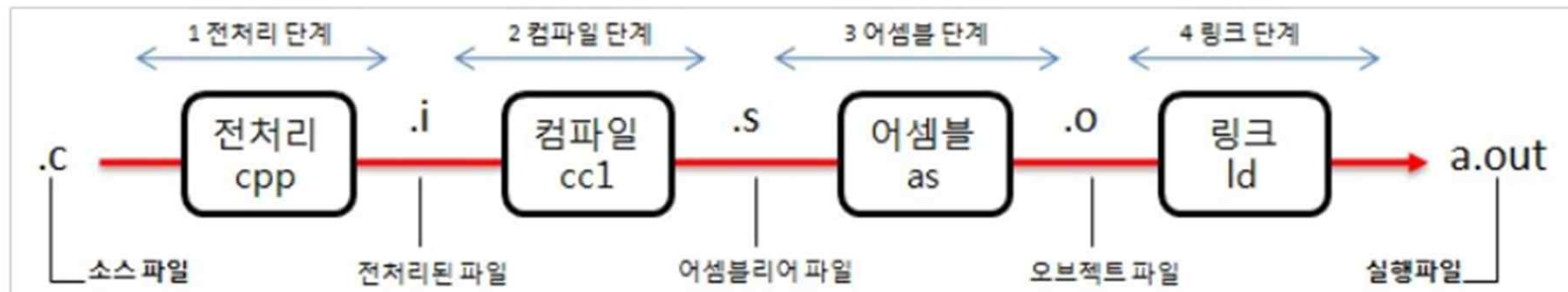
실행파일 - exec



컴파일 단계 *****

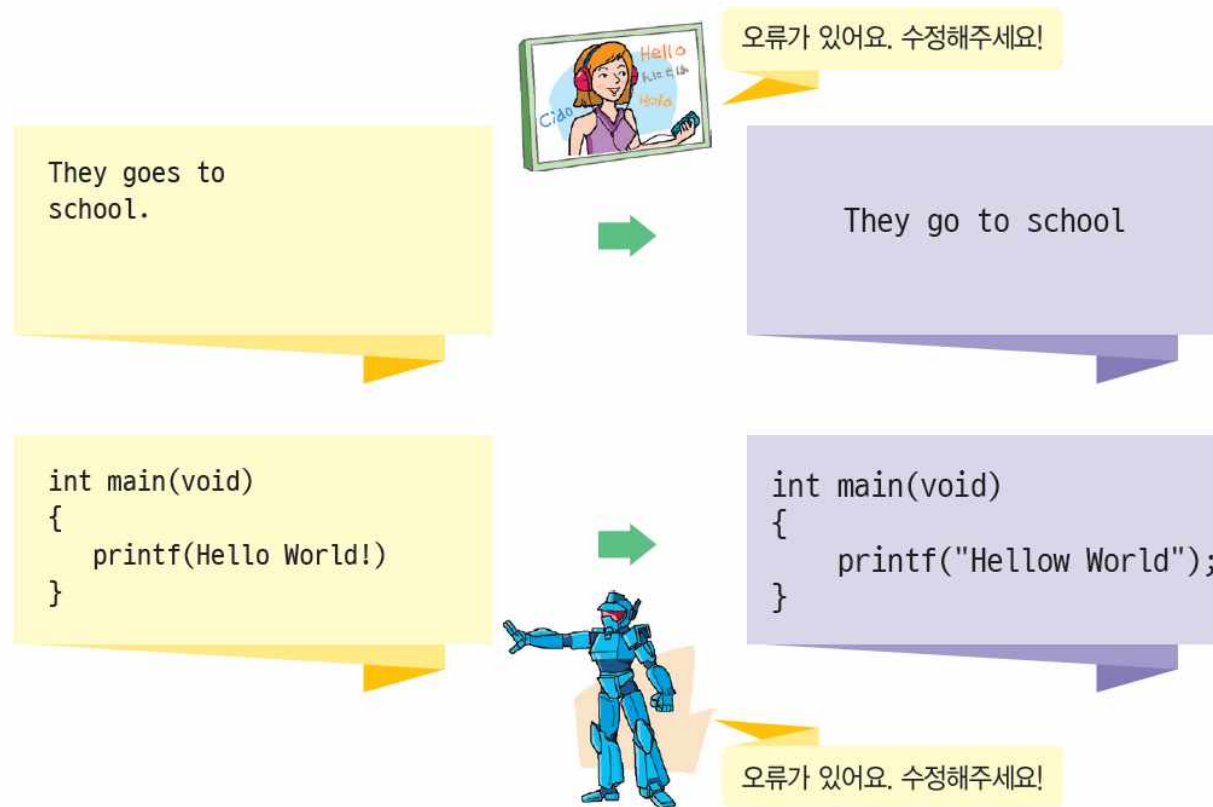
- 컴파일 단계 *****

- 컴파일(전처리(기) + 컴파일(러) + 어셈블(러)) + 링크



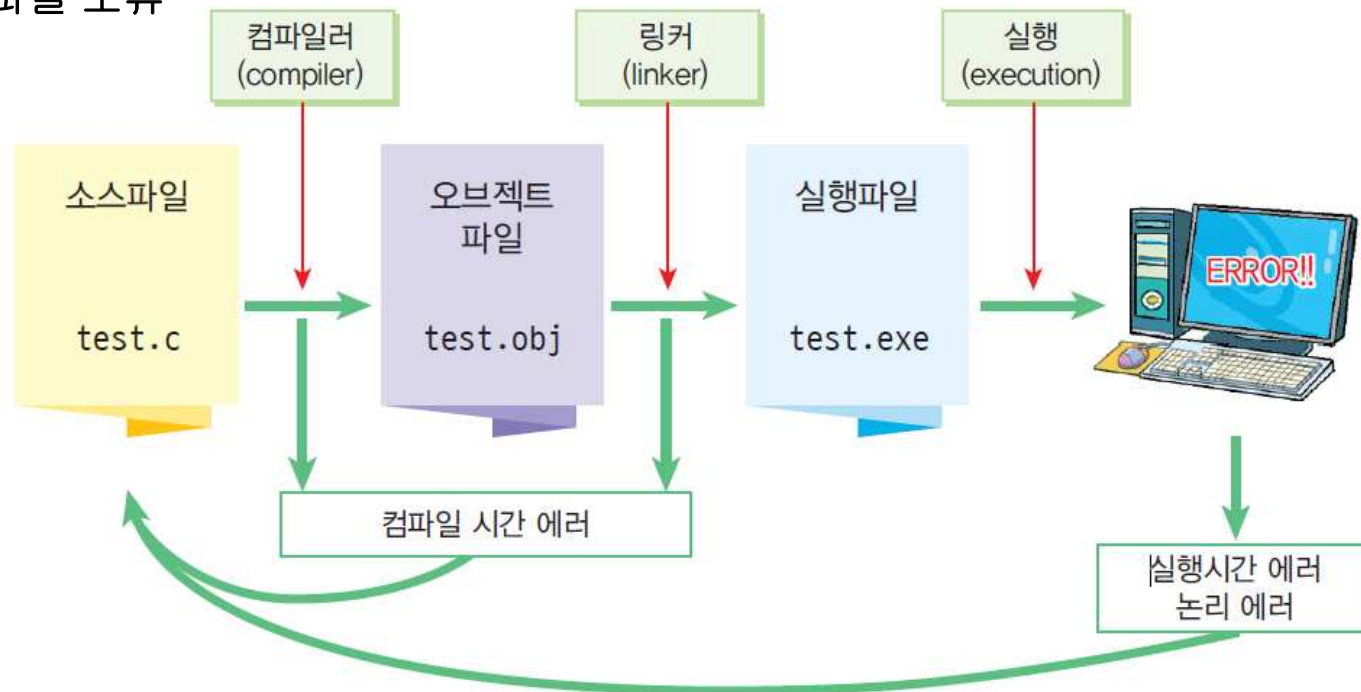
컴파일 오류

- 컴파일 오류(compile error): 문법 오류
 - (예) He go to school;



디버깅 – debug, debugging

- 디버깅: 오류(버그, Bugs)를 잡는 것
- 런타임 에러(runtime error), 실행시간 에러
 - 0으로 나누는 것, 잘못된 메모리 주소에 접근하는 것
- 논리 오류(logical error)
 - 문법은 틀리지 않았으나 논리적으로 정확하지 않는 것
- 컴파일 오류



소프트웨어의 유지 보수

- 소프트웨어의 유지 보수가 필요한 이유
 1. 디버깅 후에도 버그가 남아 있을 수 있기 때문
 2. 소프트웨어가 개발된 다음에 사용자의 요구가 추가될 수 있기 때문



장갑점검

- ① 프로그램 개발 과정을 순서대로 정리하여 보자.
- ② 소스 파일, 오브젝트 파일, 실행 파일의 차이점을 설명하여 보자.
- ③ 소스 파일이 `test.c`라면 컴파일 과정을 거친 후에 생성되는 오브젝트 파일과 실행 파일의 이름은 무엇인가?
- ④ 디버깅(debugging)이란 무엇인가?

통합 개발 환경

- 통합 개발 환경 (IDE: integrated development environment)
 - **에디터 + 컴파일러 + 디버거**
- Visual studio (C, C++) – Microsoft
- Eclipse (Java, Android) – Open Source Project



비주얼 스튜디오 설치

설치 전 주의 사항!

-Visual Studio는 불필요하게 매우 크고 무거운 치명적인 SW!

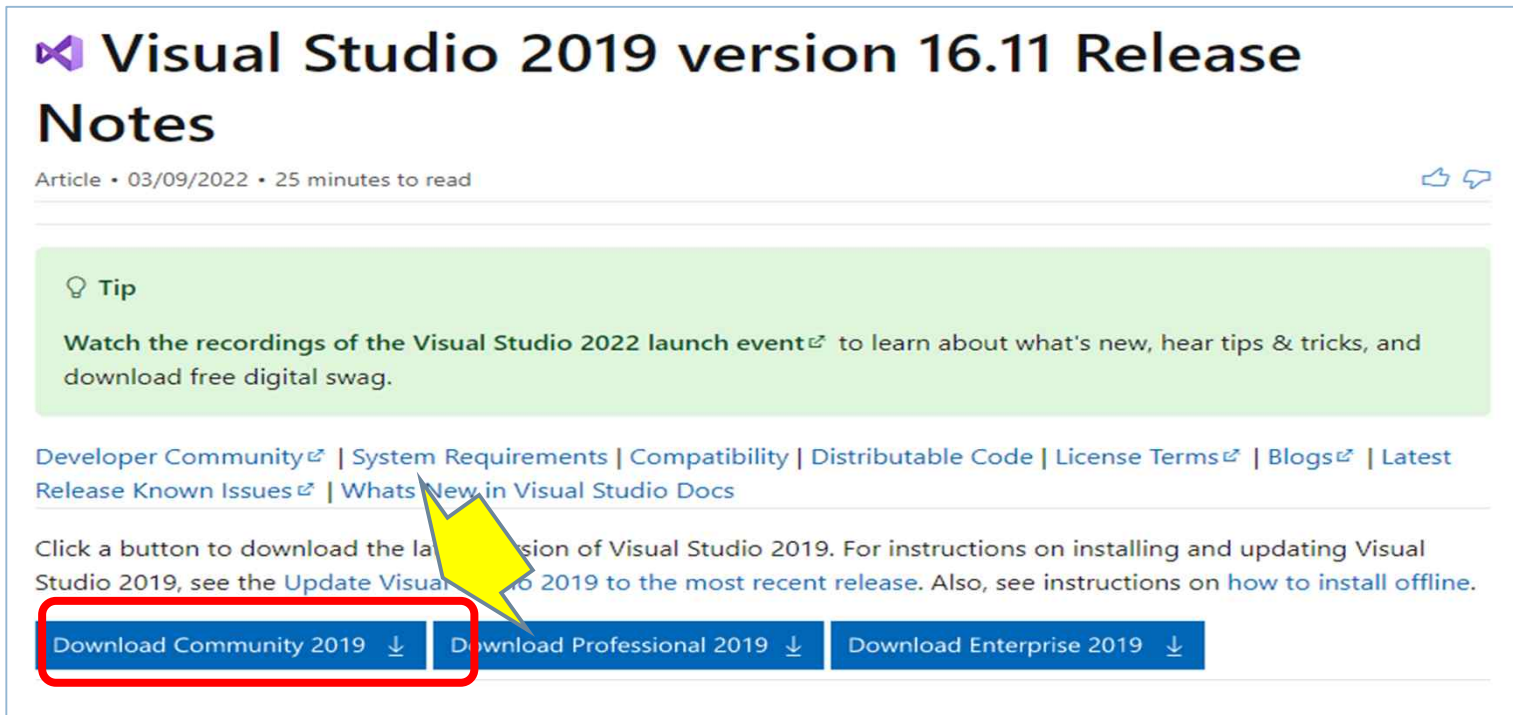
설치 중 윈도우가 깨져서 다시 설치해야 할 수도 있음!

-반드시 재부팅 후; 모든 프로그램을 닫고 실행 하지 않은 상태에서 설치 할 것.

-설치 중에도 아무것도 작업하지 말 것. 설치 완료 후 재부팅 해야 함.

-설치 파일 다운로드 경로:

<https://docs.microsoft.com/en-us/visualstudio/releases/2019/release-notes>



Visual Studio 2019 version 16.11 Release Notes

Article • 03/09/2022 • 25 minutes to read

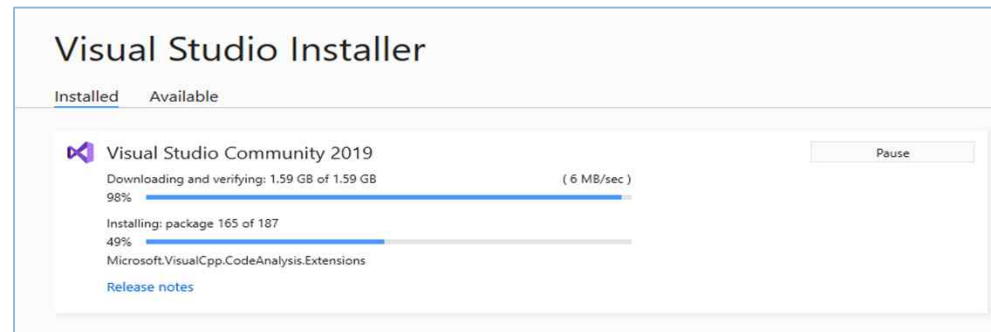
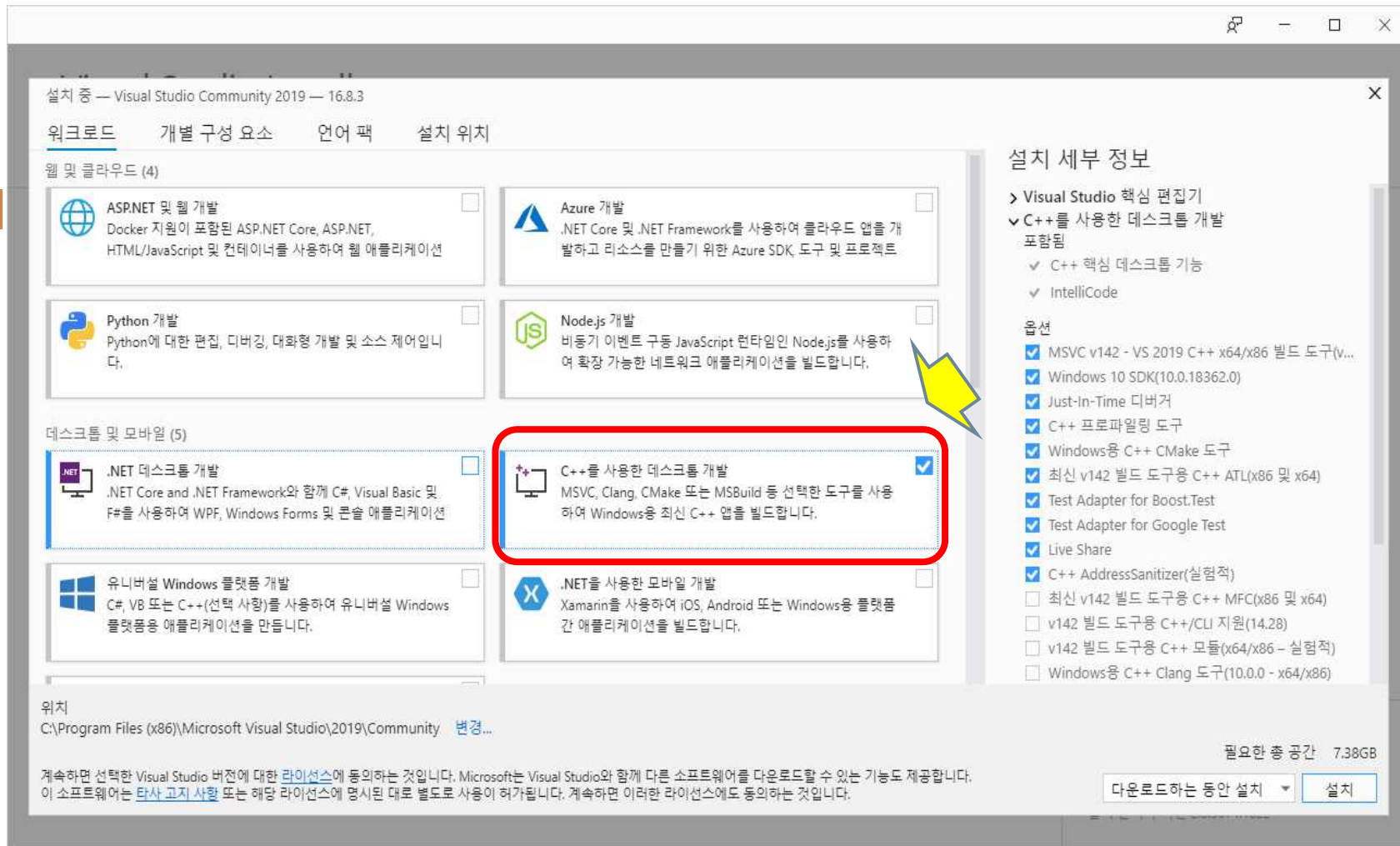
Tip

Watch the recordings of the Visual Studio 2022 launch event to learn about what's new, hear tips & tricks, and download free digital swag.

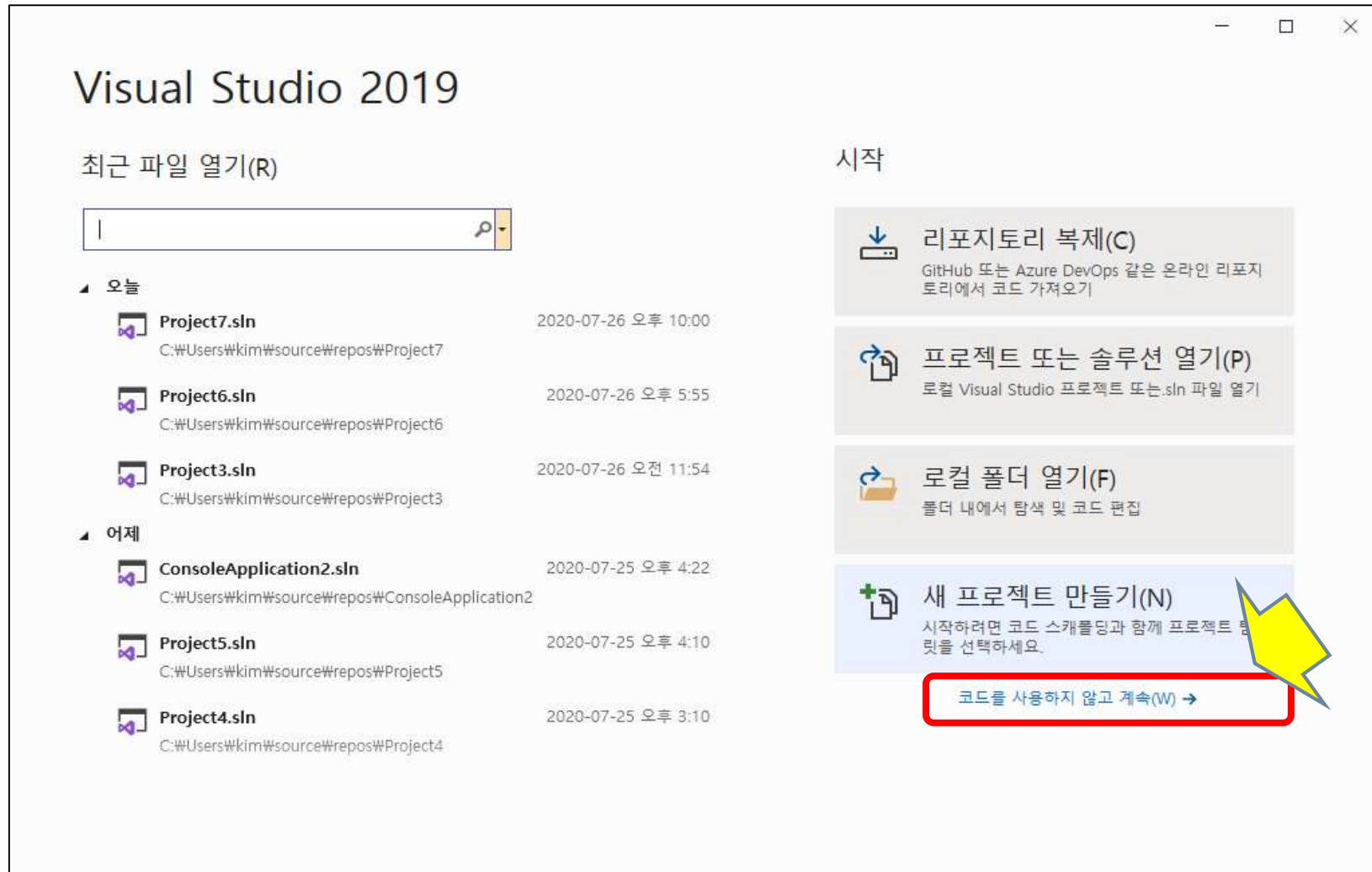
[Developer Community](#) | [System Requirements](#) | [Compatibility](#) | [Distributable Code](#) | [License Terms](#) | [Blogs](#) | [Latest Release](#) | [Known Issues](#) | [What's New in Visual Studio Docs](#)

Click a button to download the latest version of Visual Studio 2019. For instructions on installing and updating Visual Studio 2019, see the [Update Visual Studio 2019 to the most recent release](#). Also, see instructions on [how to install offline](#).

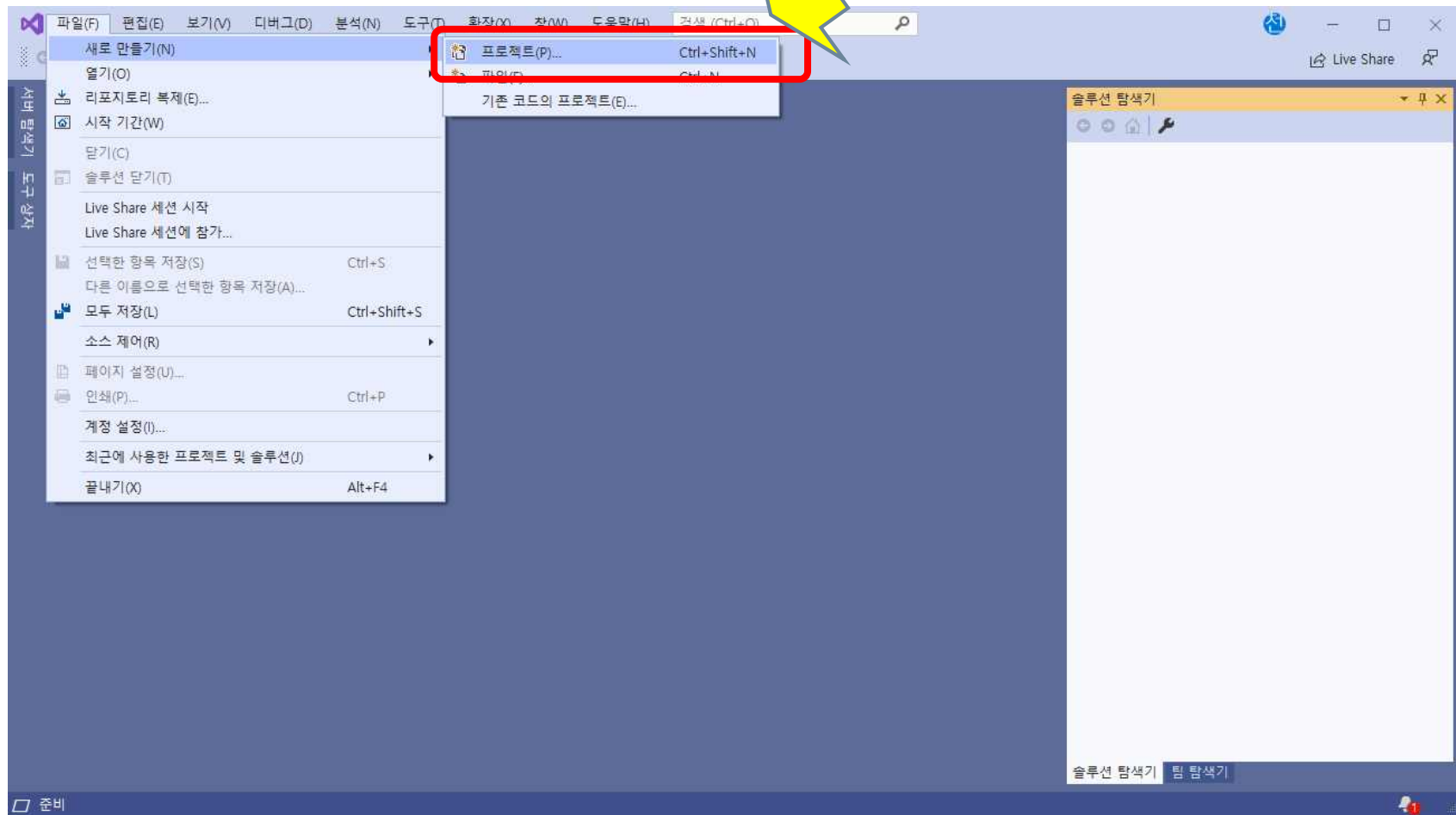
[Download Community 2019](#) | [Download Professional 2019](#) | [Download Enterprise 2019](#)



비주얼 스튜디오 시작



프로젝트 생성하기



프로젝트 생성하기



프로젝트 생성하기

새 프로젝트 구성

빈 프로젝트 콘솔 C++ Windows

프로젝트 이름(N)

hello

위치(L)

C:\Users\Wkim\source\repos

솔루션 이름(M) ⓘ

hello

☒ 솔루션 및 프로젝트를 같은 디렉터리에 배치(D)

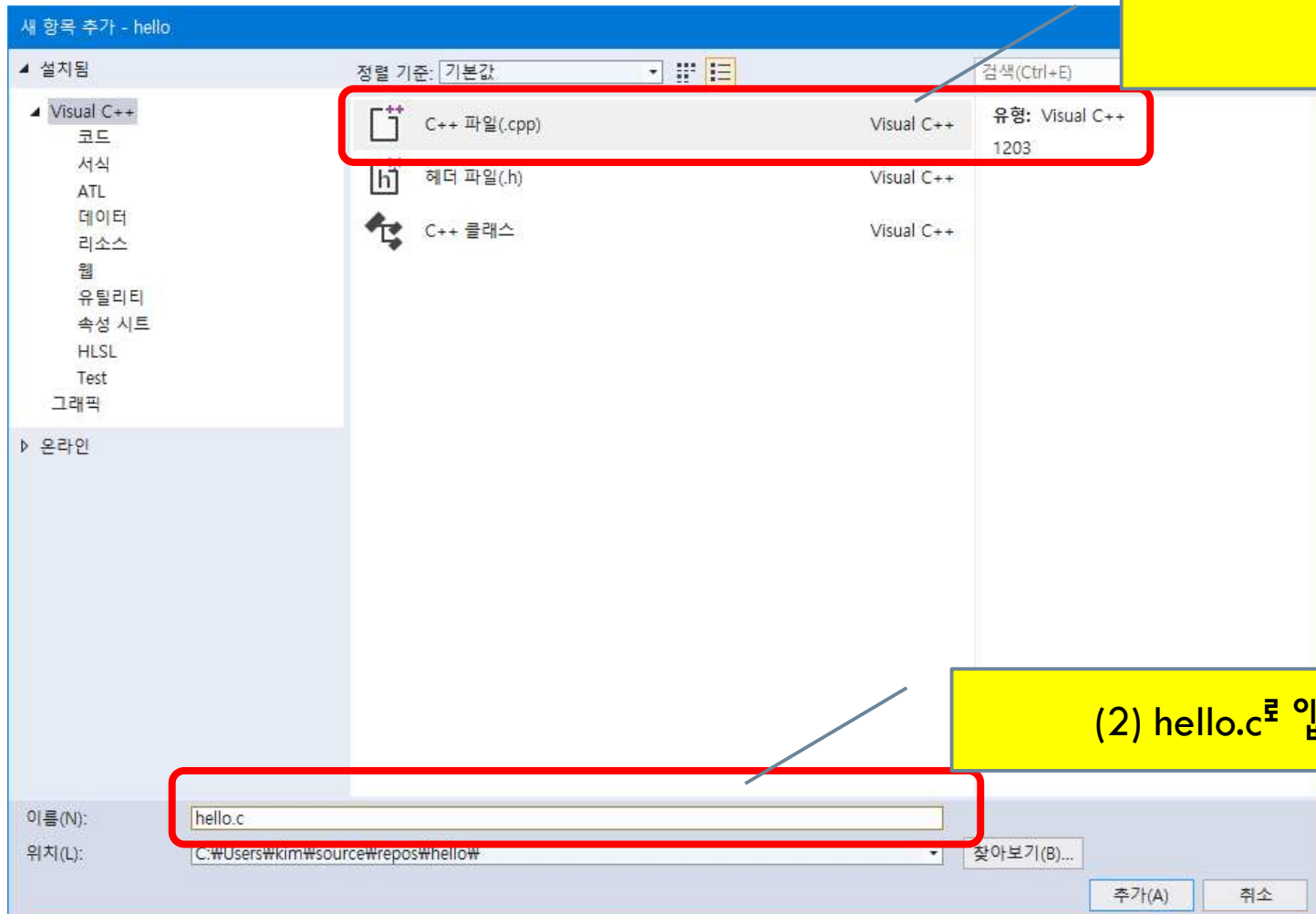
뒤로(B) 만들기(C)

(1) hello로 입력

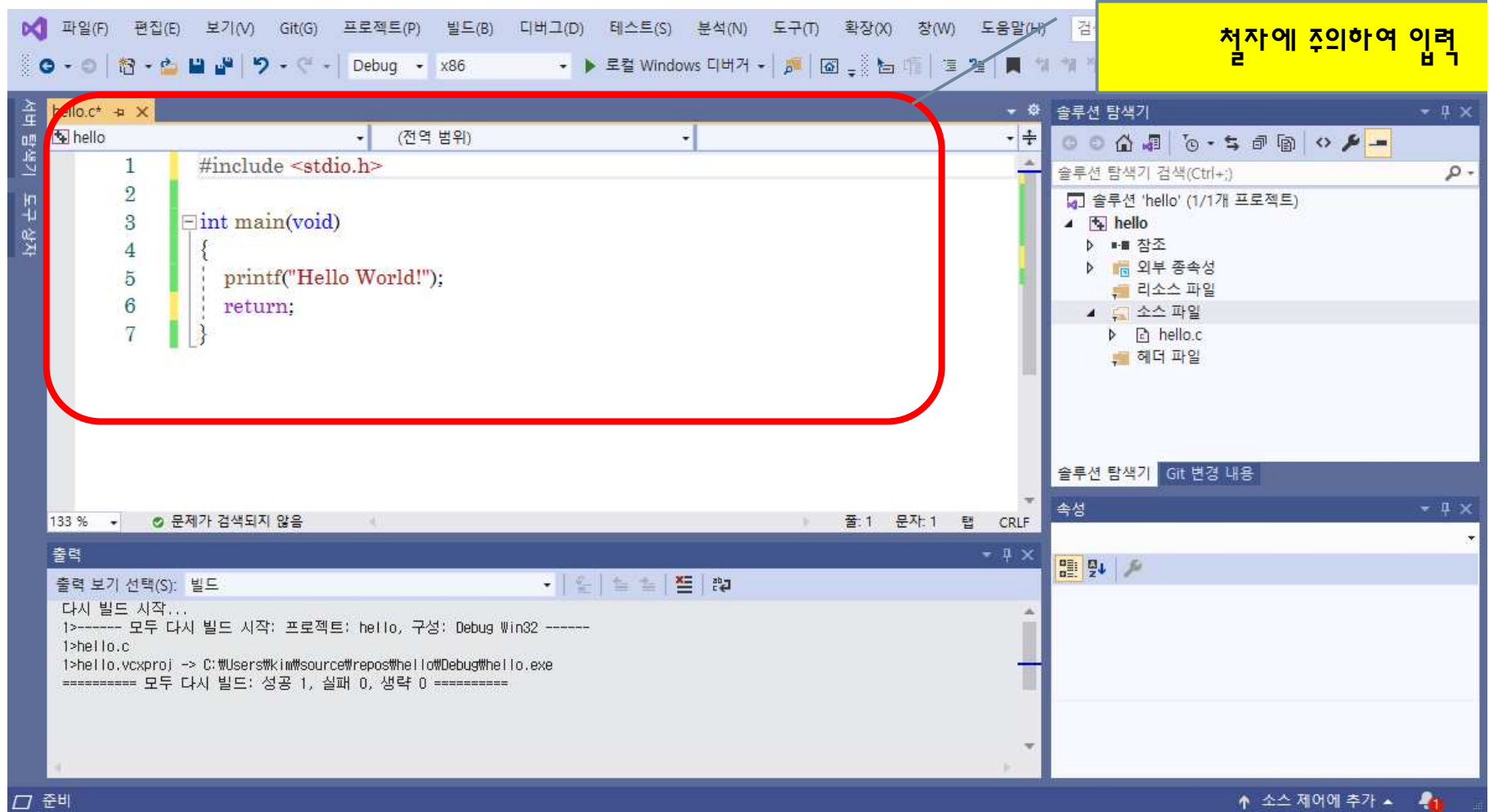
(2) 원하는 디렉토리 선택
D드라이브 -> 컴소1학년 -> A김국현 / B김국현 -> dev



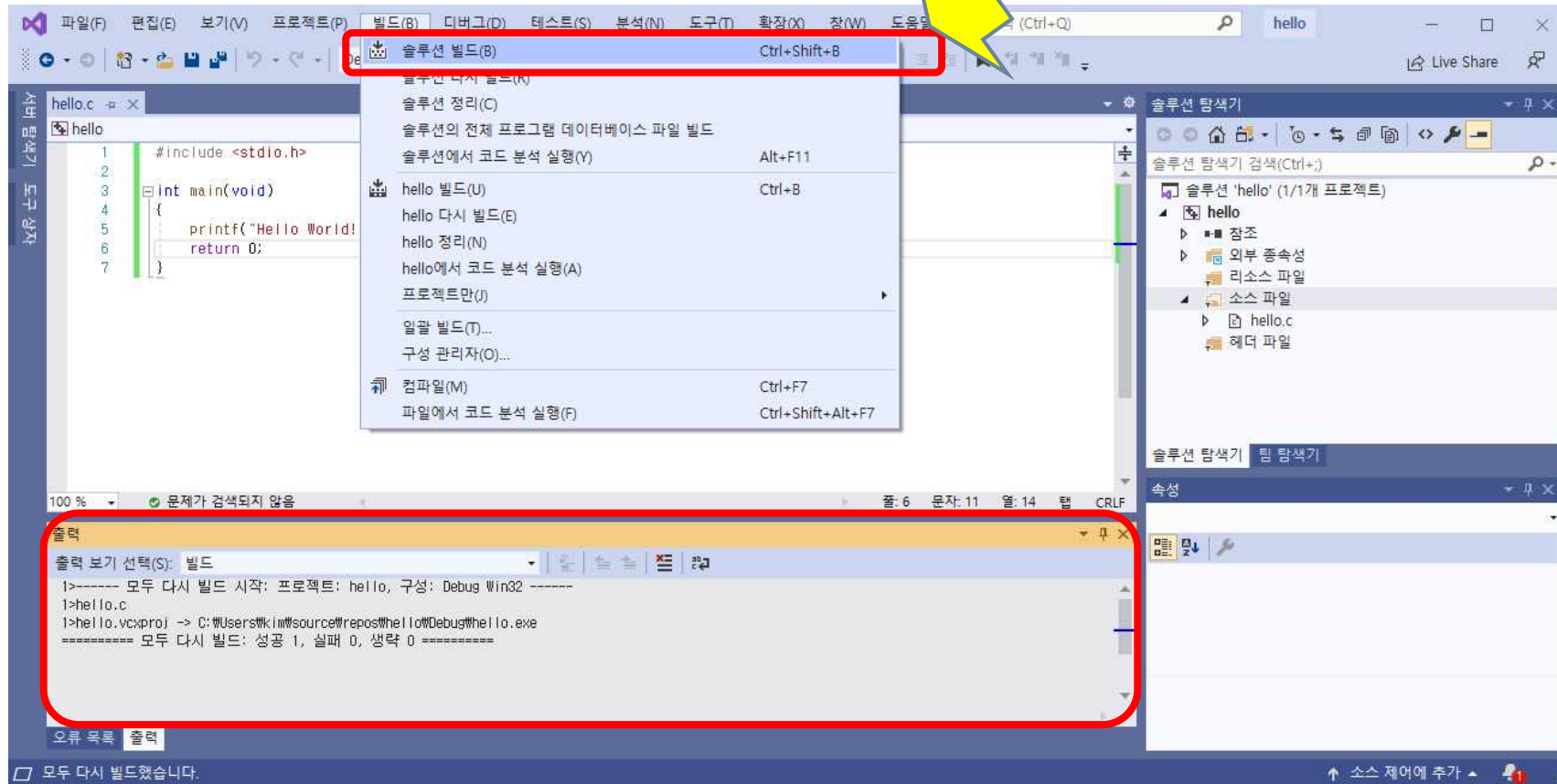
소스 파일 생성하기



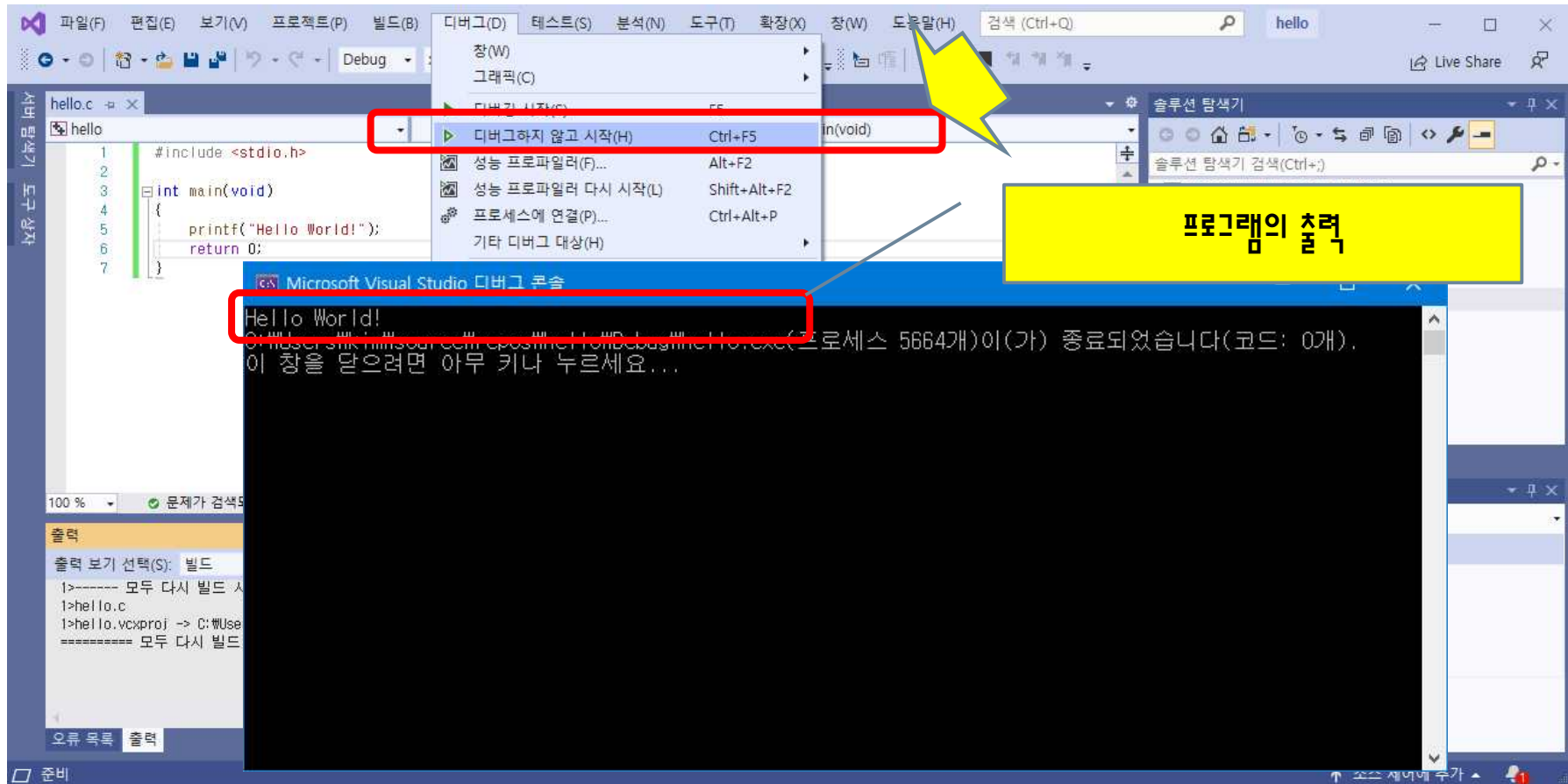
프로그램 입력



컴파일하기



프로그램 실행 하기



첫 번째 프로그램의 설명

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

```
int main(void)
```

```
{
```

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

```
    return 0;
```

```
}
```



Hello World!

작업을 적어주는 위치

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

```
int main(void)
```

```
{
```



```
    return 0;
```

```
}
```

여기다 원하는 작업을
수행하는 문장을 적어준다.

프로그램

간략한 소스 설명

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

헤더파일을 포함한다.

```
int main(void)
```

메인 함수 시작

```
{
```

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

화면에 "Hello World!"를 출력

```
    return 0;
```

외부로 0값을 반환

```
}
```

메인 함수 종료

프로그램

헤더 파일 포함

- `#include`는 소스 코드 안에 특정 파일을 현재의 위치에 포함

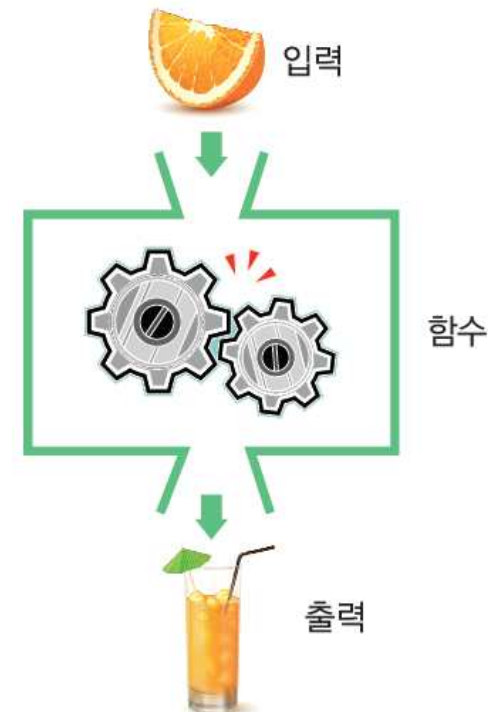
- 주의!: 전처리기 지시자 문장 끝에는 세미콜론(;)을 붙이면 안 된다.

`#include <stdio.h>`

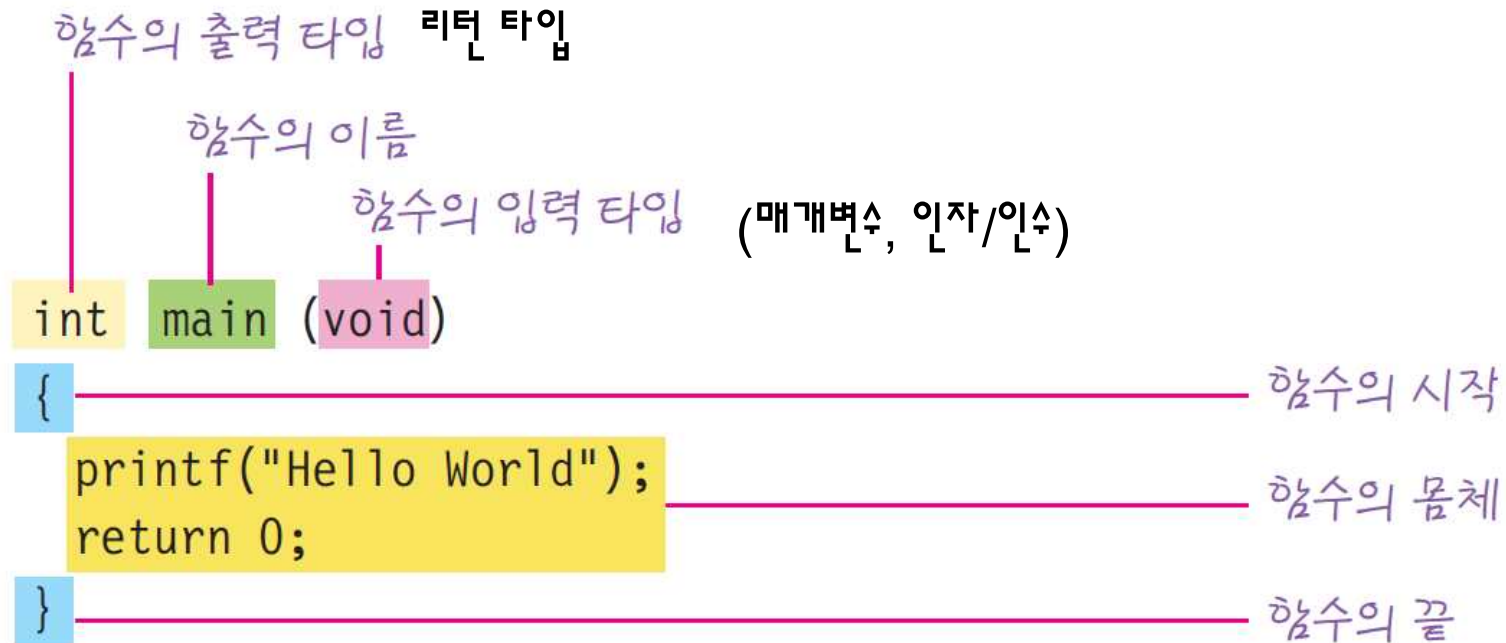
- 헤더 파일(header file): 컴파일러가 필요로 하는 정보를 가지고 있는 파일
- stdio.h: standard input output header file

함수

- 함수(function)
 - 프로그래밍의 기본 단위
 - 특정한 작업을 수행하기 위하여 작성된 독립적인 코드
- (참고) 수학적 함수 $y = x^2 + 1$
- 프로그램 = 함수의 집합
- main() 함수
 - 프로그램의 시작 함수

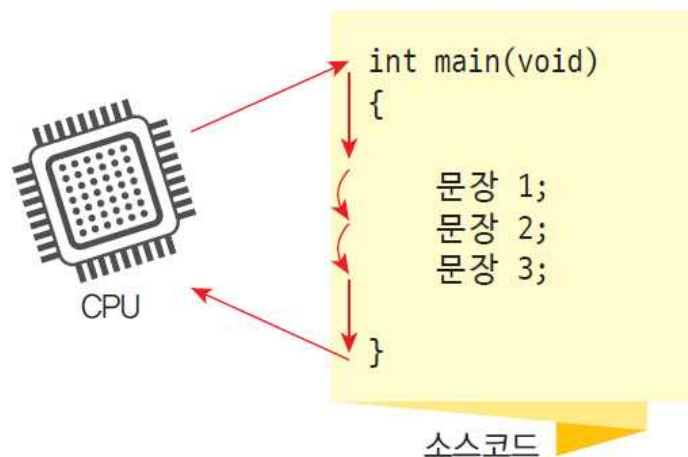


함수의 간단한 설명



문장(명령문)

- 함수는 여러 개의 문장으로 이루어진다.
- 문장들은 순차적으로 실행된다.
- 문장의 끝에는 반드시 ; 이 있어야 한다.



장간 점검

1. 문장의 끝에 추가하여야 하는 기호는?
2. C프로그램에 반드시 있어야 하는 함수는?
3. `printf()`가 하는 기능은 무엇인가?



응 프로그램 #1

- 다음과 같은 출력을 가지는 프로그램을 제작하여 보자.



첫 번째 버전

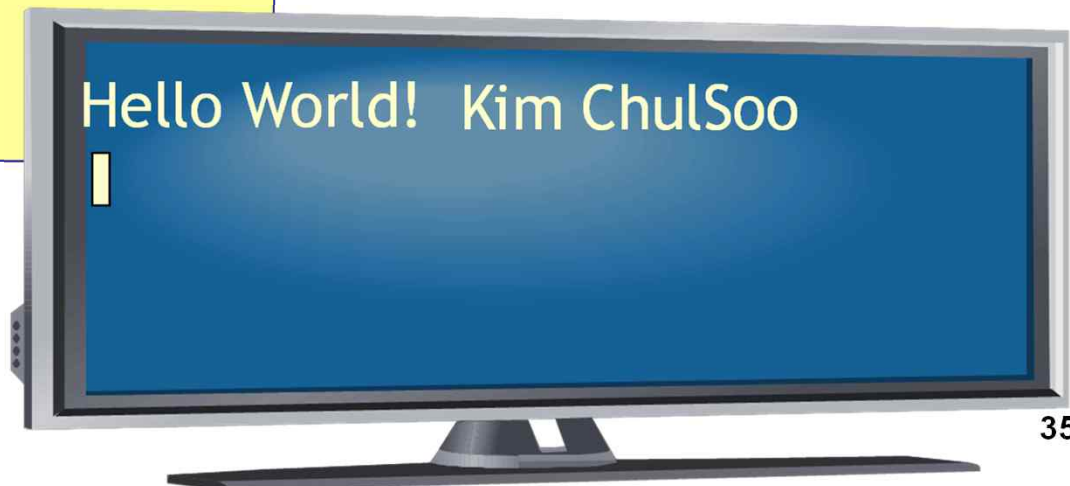
- 문장들은 순차적으로 실행된다는 사실 이용

```
#include <stdio.h>

int main(void)
{
    printf("Hello World!");
    printf("Kim ChulSoo");

    return 0;
}
```

2개의 문장은 순차적으로
실행된다



잘바꿈 문자 2개를 사용하면?

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



변경된 프로그램

- 줄바꿈 문자를 추가하면 우리가 원하던 결과가 된다.

```
#include <stdio.h>

int main(void)
{

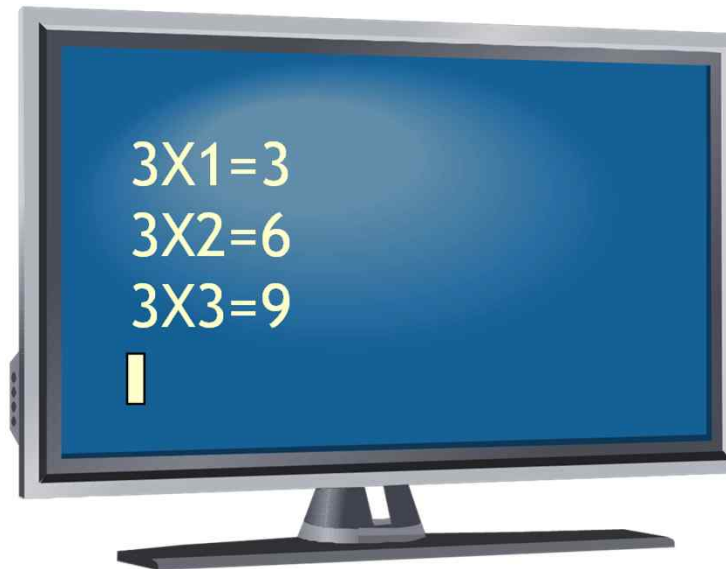
    printf("Hello World!\n");
    printf("Kim ChulSoo \n");

    return 0;
}
```



응 프로그램 #2

- 다음과 같은 출력하는 프로그램을 제작하여 보자.



- 역시 문장들은 순차적으로 수행된다는 점을 이용한다.

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

```
int main(void)
```

```
{
```

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

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

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

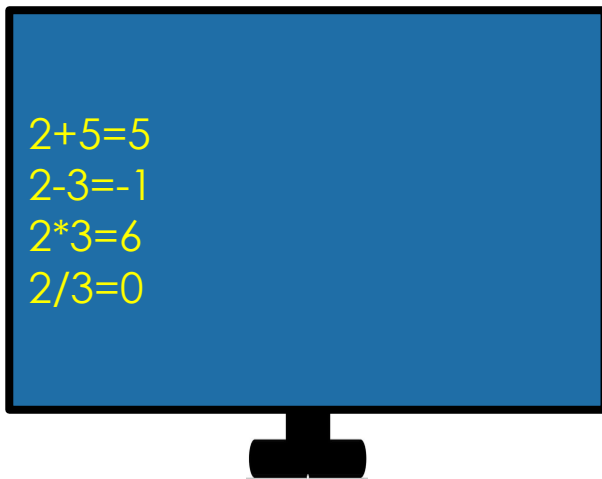
```
    return 0;
```

```
}
```

3개의 문장은
순차적으로 실행된다.

Lab: 간단한 계산을 해보자

- 덧셈과 뺄셈, 곱셈, 나눗셈 계산을 하는 프로그램을 작성해보자.



```
#include <stdio.h>

int main(void)
{
    printf("2+5=%d\n", 2 + 3);
    printf("2-3=%d\n", 2 - 3);
    printf("2*3=%d\n", 2 * 3);
    printf("2/3=%d\n", 2 / 3);
    return 0;
}
```


오류 수정 및 디버깅

- 컴파일이나 실행 시에 오류가 발생할 수 있다.
- 에러와 경고
 - 에러(error): 심각한 오류
 - 경고(warning): 경미한 오류

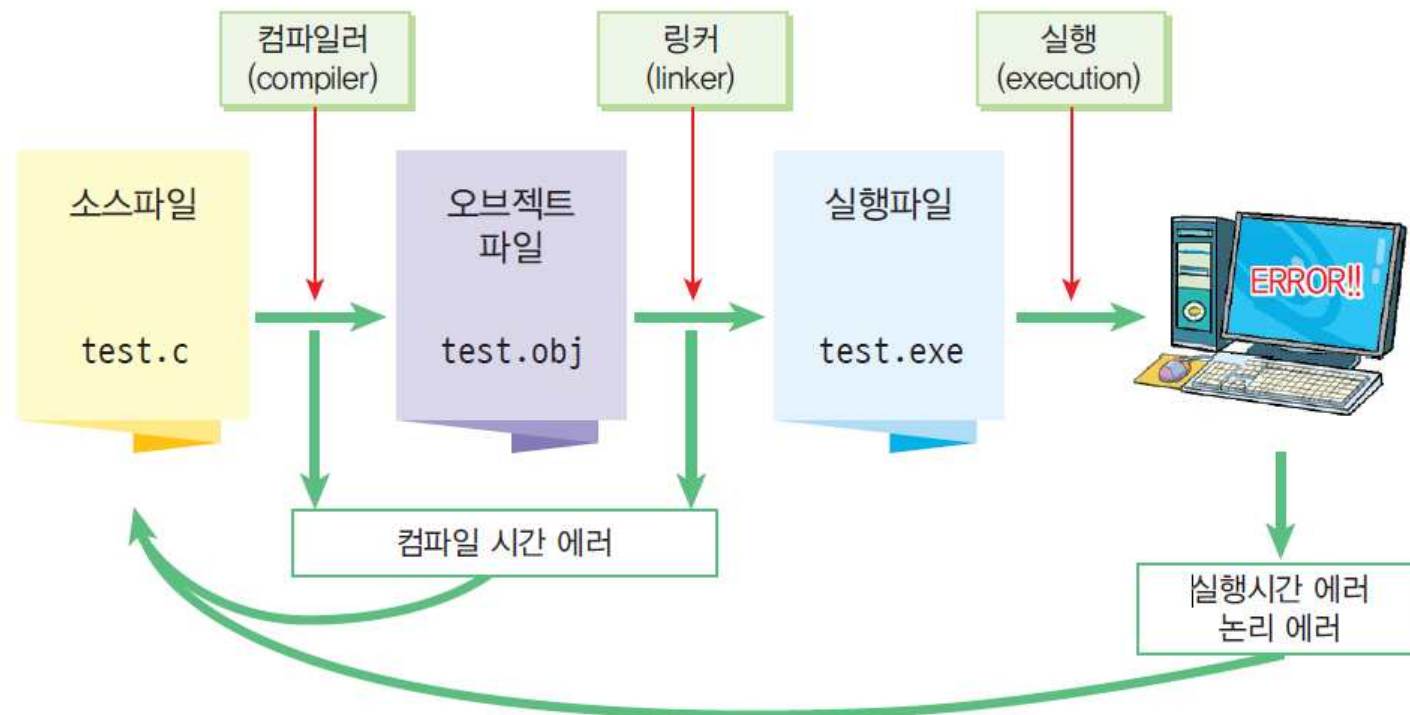


오류의 종류

- 오류의 종류
 - 컴파일 시간 오류: 대부분 문법적인 오류
 - Runtime(실행시간) 오류: 실행 중에 발생하는 오류
 - 잘못된 메모리 접근, 0으로 나누는 연산, ...
 - 논리 오류: 논리적으로 잘못되어서 결과가 의도했던 대로 나오지 않는 오류



오류 수정 과정



오류 #1

add - Microsoft Visual Studio

빠른 실행(Ctrl+Q)

파일(F) 편집(E) 보기(V) 프로젝트(P) 빌드(B) 디버그(D) 팀(M) 도구(T) 테스트(S) 분석(N) 창(W) 도움말(H) 로그인

error1.c* x86 로컬 Windows 디버거

add

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     printf("Hello World!");
6     return 0;
7 }
```

;이 생략되었다.

retrun 앞에 ;를 빠뜨렸다는 의미이다.

오류 목록

전체 솔루션 2 오류 0 경고 0 메시지 빌드 + IntelliSense 검색 오류 목록

코드	설명	프로젝트	파일	줄	비표시 오류(Suppr...)
C2143	'가 필요합니다. 구문 오류: ';'이(가) 'return' 앞에 없습니다.	add	error1.c	6	

오류가 발견된 소스 파일

오류가 발견된 줄번호

오류 목록 출력

준비 줄: 7 열: 2 문자: 2 INS 게시

오류 #2

Visual Studio interface showing a C program in '소스.c' with the following code:

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     printf(Hello World!);
6     return 0;
7 }
8
```

A red box highlights the line `printf(Hello World!);`. A callout bubble points to it with the text: "문자열을 표시할 때, “.”을 빠뜨렸다." (When displaying a string, I forgot the period ".").

The bottom panel shows the '오류 목록' (Error List) with the following errors:

코드	설명	프로젝트	파일	줄
E0020	식별자 "Hello"이(가) 정의되어 있지 않습니다.	Project14	소스.c	5
E0018	'!'가 필요합니다.	Project14	소스.c	5
C6031	반환 값이 무시되었습니다. 'scanf'.	Project14	소스.c	62
C6031	반환 값이 무시되었습니다. 'scanf'.	Project14	소스.c	65
C2065	'Hello': 선언되지 않은 식별자입니다.	Project14	소스.c	5

오류 #3

The screenshot shows the Microsoft Visual Studio IDE with a C program named 'error3.c'. The code is as follows:

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     print("Hello World!");
6     return 0;
7 }
```

A callout bubble points to the `print` function, stating: "printf이어야 한다." (It should be printf).

The '오류 목록' (Error List) pane at the bottom shows the following errors:

- C4013 'print'이(가) 정의되지 않았습니다. extern은 int형을 반환하는 add 것으로 간주합니다.
- LNK2015 print 외부 기호(참조 위치: _main 함수)에서 확인하지 못했음 add error3.obj 1
- LNK1120 1개의 확인할 수 없는 외부 참조입니다. add add.exe 1

A callout bubble points to the LNK2015 error, stating: "함수를 찾지 못했음." (Could not find the function).

The '솔루션 탐색기' (Solution Explorer) on the right shows the project structure for 'add' (1개 프로젝트):

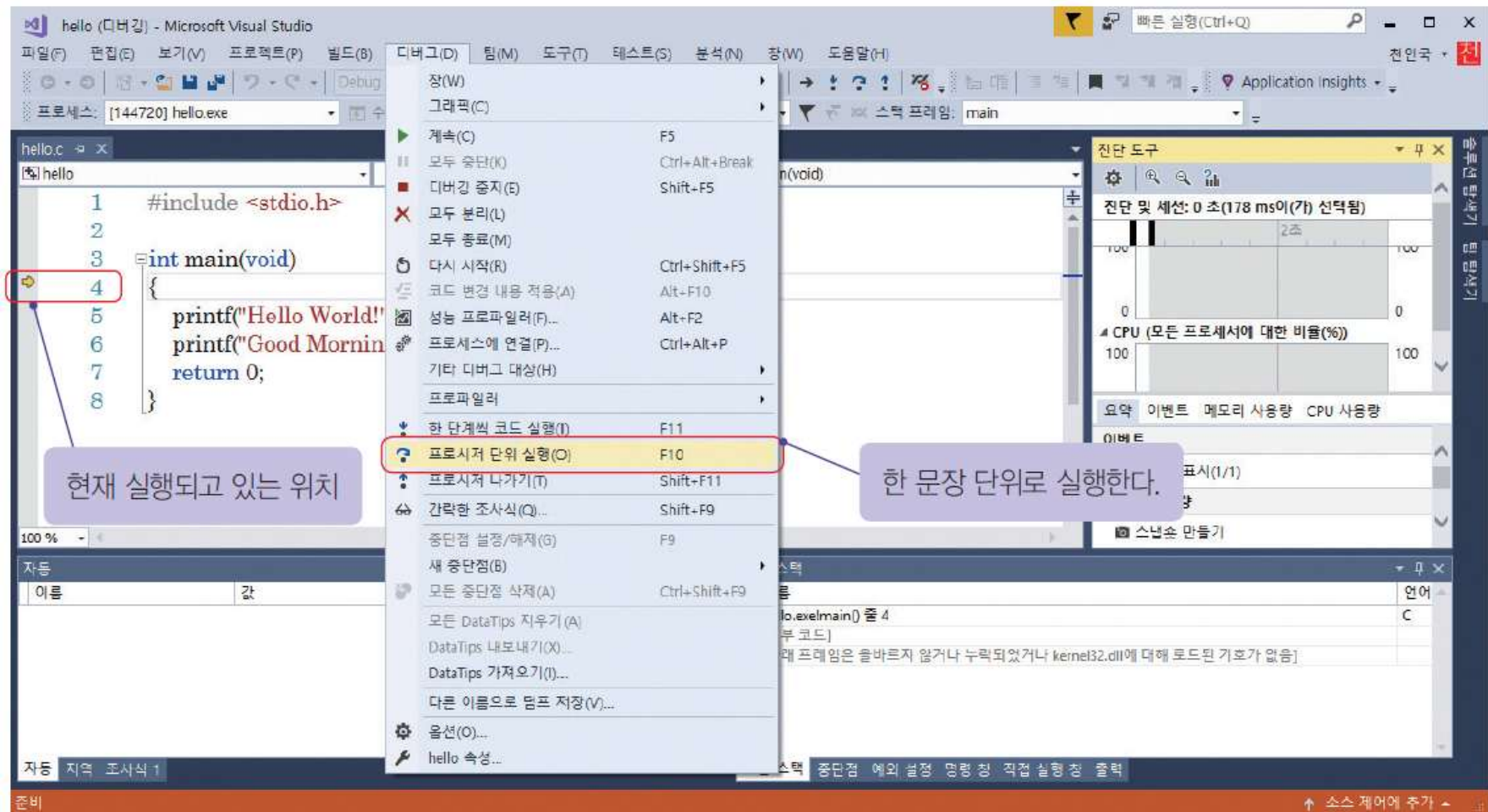
- add
 - 참조
 - 외부 종속성
 - 리소스 파일
 - 소스 파일
 - error3.c
 - 헤더 파일

The '속성' (Properties) pane on the right shows the properties for the 'main' VCCodeFunction:

속성	값
(Name)	main
File	d:\sources\Wch
FullName	main

The status bar at the bottom indicates '빌드 실패' (Build Failed).

디버거(debugger)



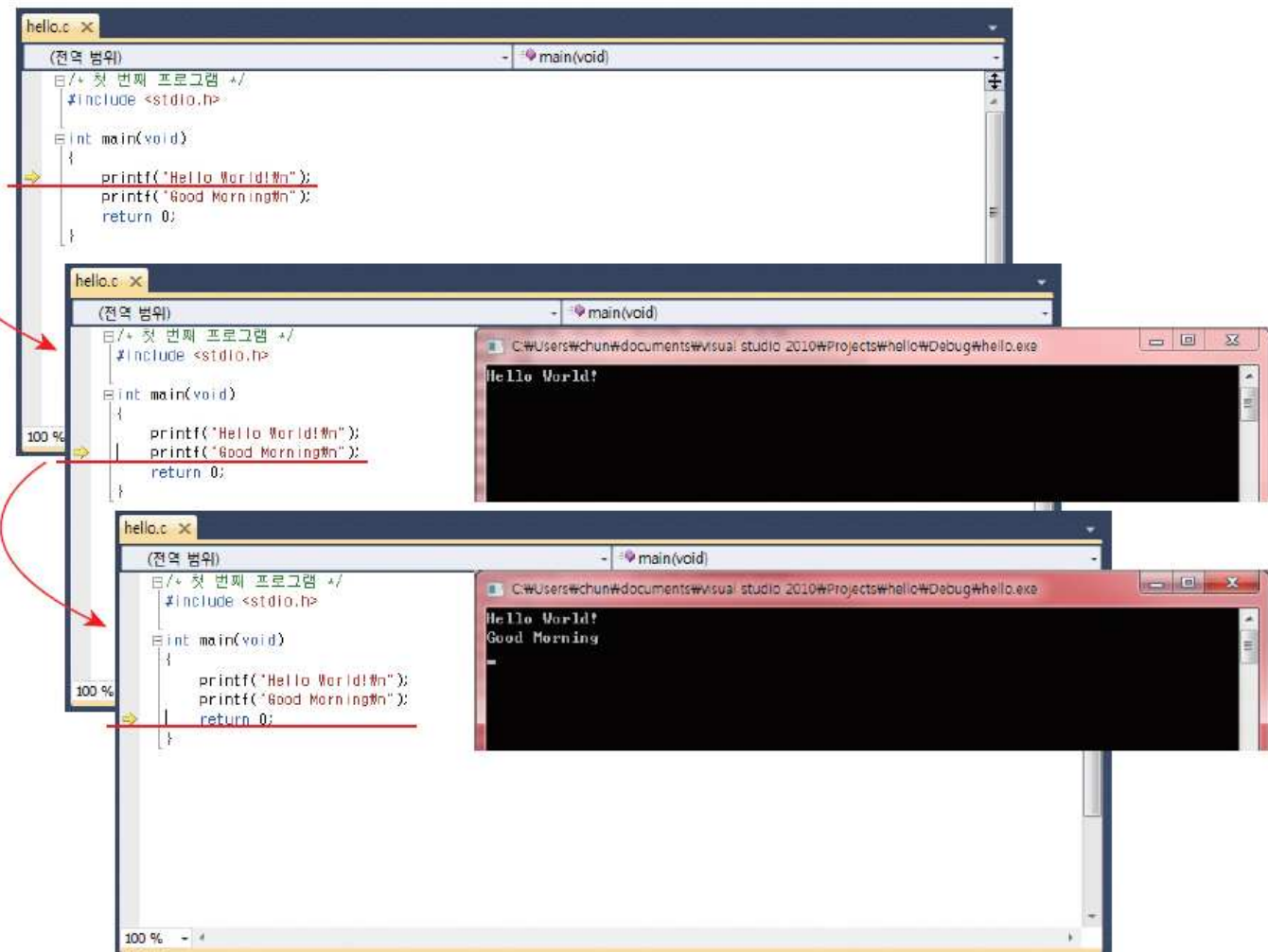
디버거의 명령어 정의

- 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

디버거의 실행 과정

F10을 누를 때마다
한 문장씩 실행된다.

F10을 누를 때마다
한 문장씩 실행된다.



Mini Project

bug.c

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

```
2
```

```
3  int Main(void)
```

```
4  (
```

```
5      printf(안녕하세요? \n);
```

```
6      printf(이번 코드에는 많은 오류가 있다네요 \n);
```

```
7      print(제가 다 고쳐보겠습니다.\n);
```

```
8      return 0;
```

```
9  )
```

main

(가 아니라 {이어야 한다.

문장의 끝에는 ;가 있어야 한다.

print가 아니고 printf이어야 한다.

문자열에는 따옴표를 붙인다.

Mini Project

- 오류가 수정된 프로그램

```
#include <stdio.h>

int main(void)
{
    printf("안녕하세요 ? \n");
    printf("이번 코드에는 많은 오류가 있네요\n");
    printf("제가 다 고쳐보겠습니다.\n");
    return 0;
}
```

Q & A



- In conclusion,
 - 컴파일 단계 ***
 - 개발환경 구축