1장. C 언어와 프로그래밍 개요

박 종 혁 교수
서울과학기술대학교 컴퓨터공학과
UCS Lab
Tel: 970-6702
Email: jhpark1@seoultech.ac.kr

목차

- ❖ C 소개
  - 프로그래밍의 개념
  - C 언어의 특징
- ❖ C 프로그램 개발
  - 일반적인 C 프로그램 개발 과정
  - Visual Studio를 이용한 C 프로그램 개발 과정

소프트웨어의 중요성

프로그램
응용 프로그램

앱

컴퓨터 하드웨어를
작동할 수 있게 만든다.

소프트웨어

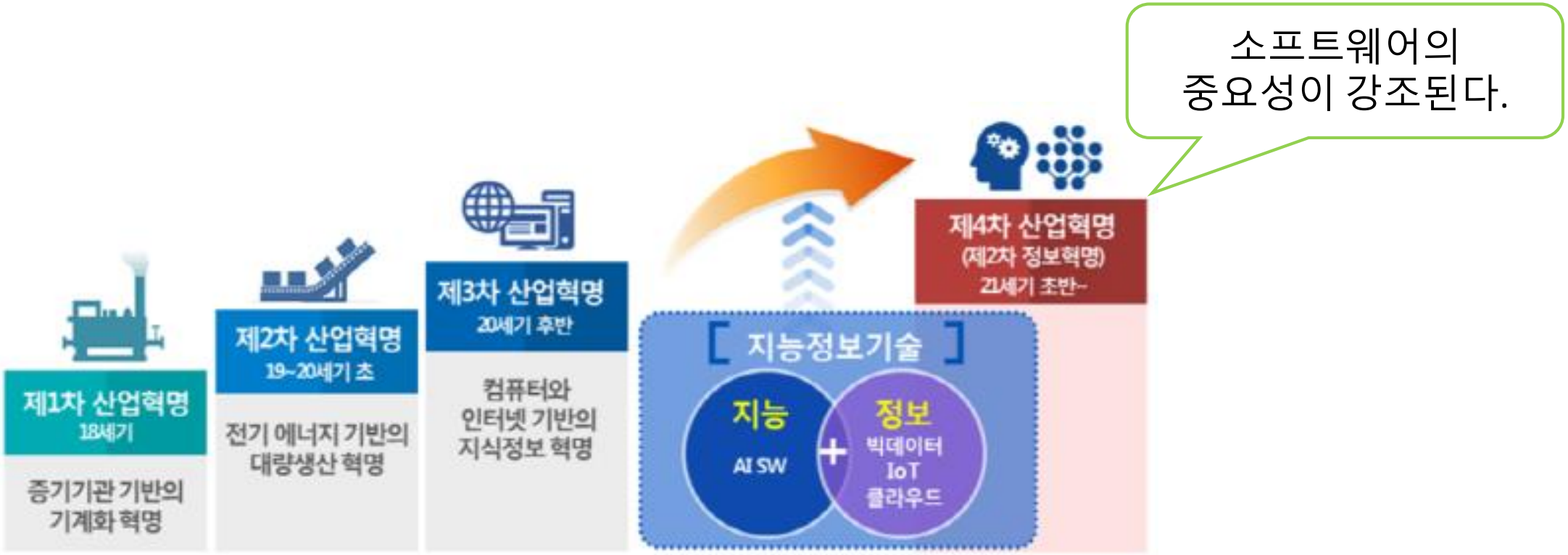
컴퓨터
하드웨어

PC, 노트북

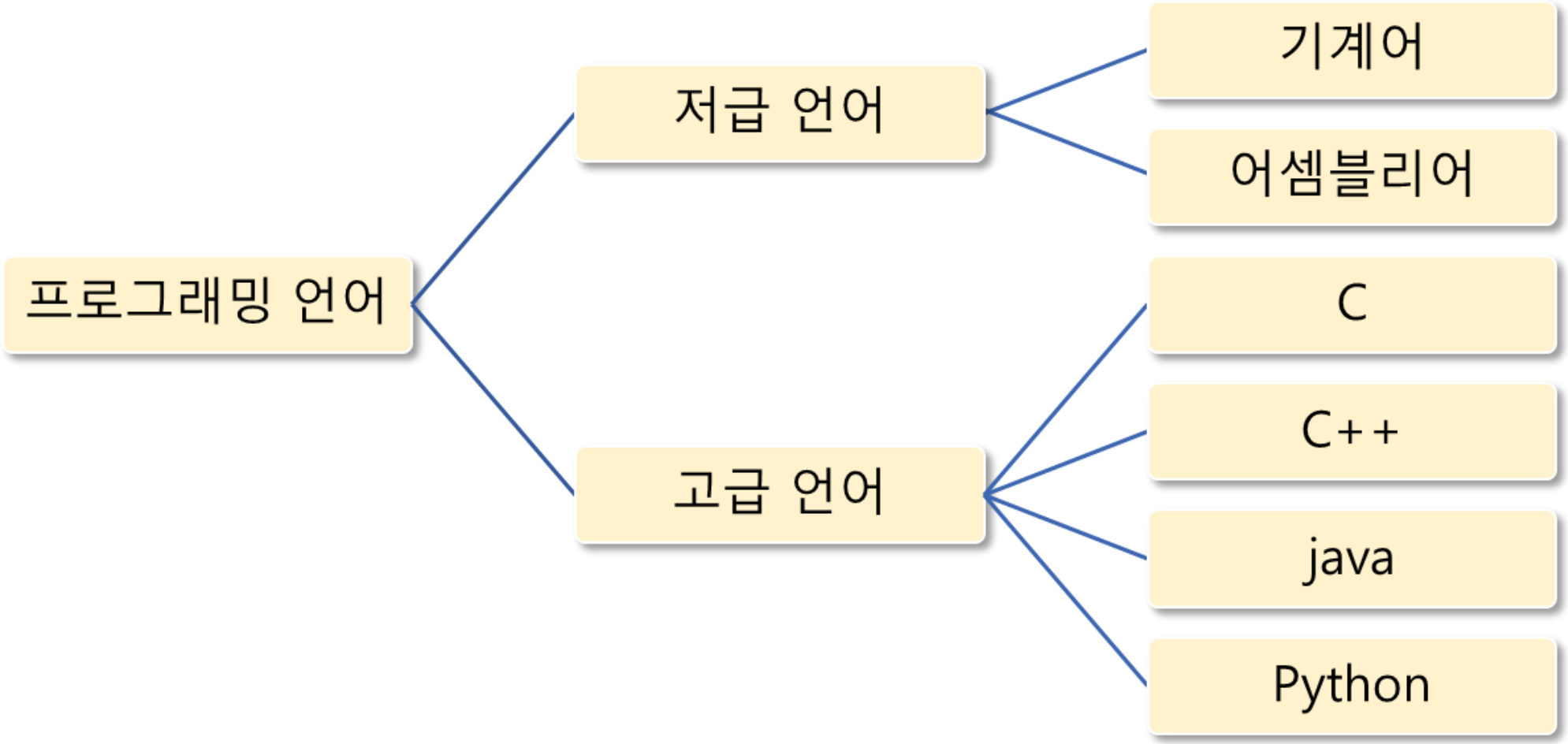
스마트폰

지능정보기술

- 인공지능, 사물 인터넷, 클라우드 컴퓨팅, 빅데이터, 모바일



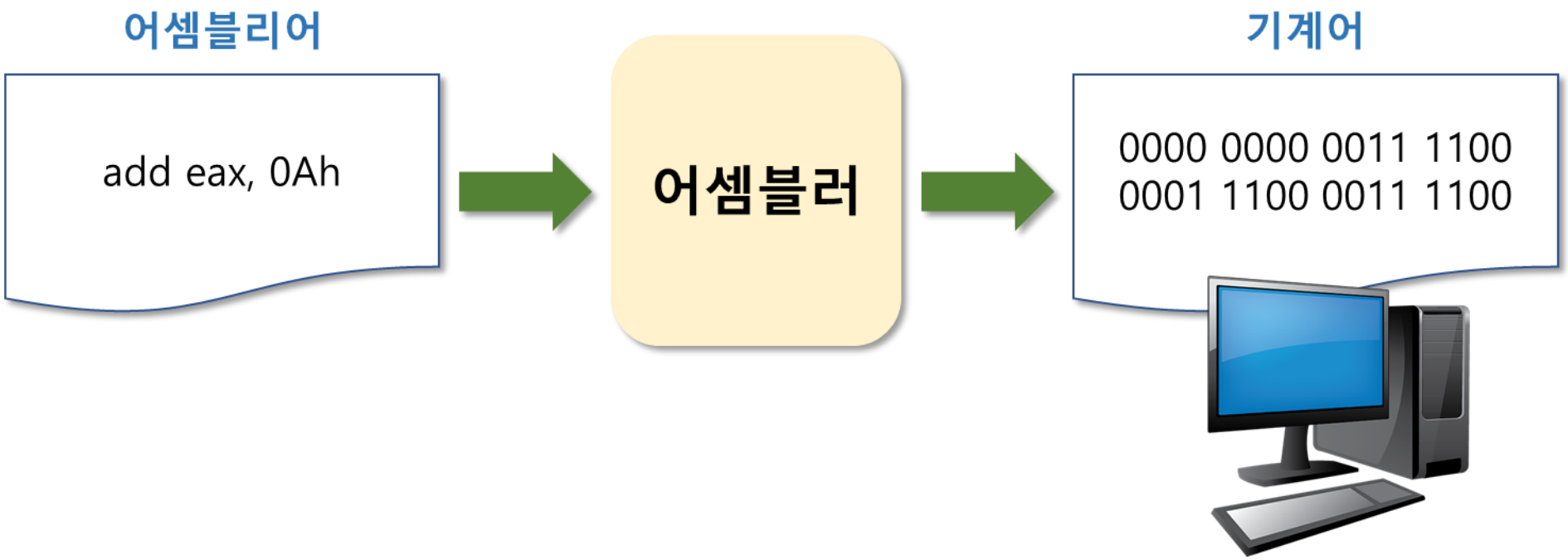
프로그래밍 언어



저급 언어

- **기계어** : 2진으로 된 CPU 명령어
- **어셈블리어** : 기계어와 1:1로 대응되는 니모닉 기호

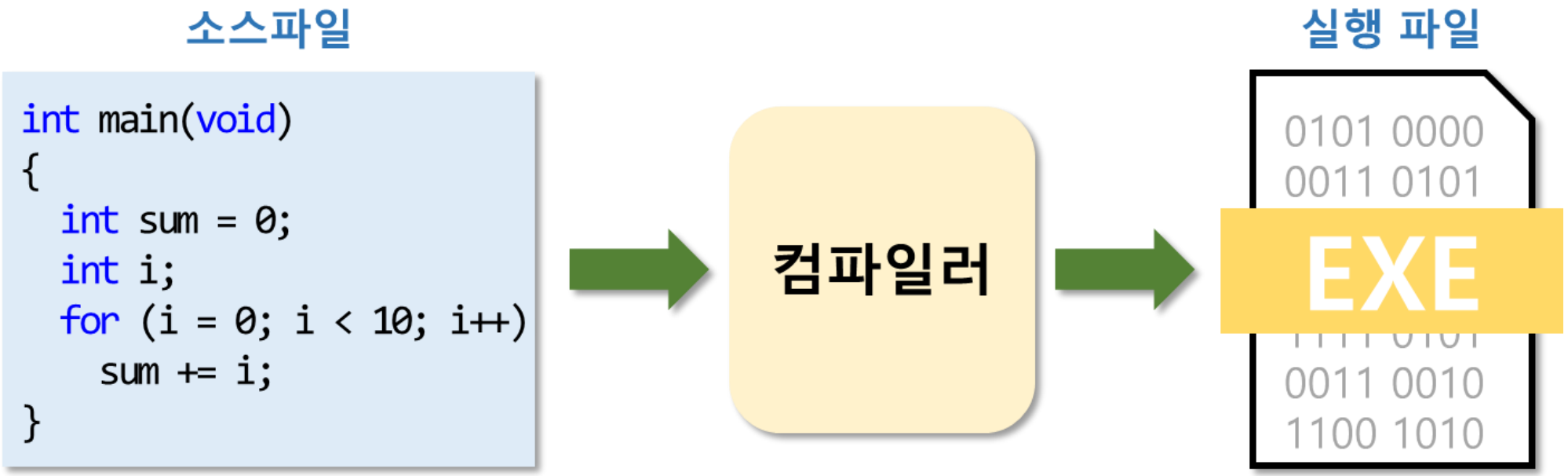
기계종속적



고급 언어

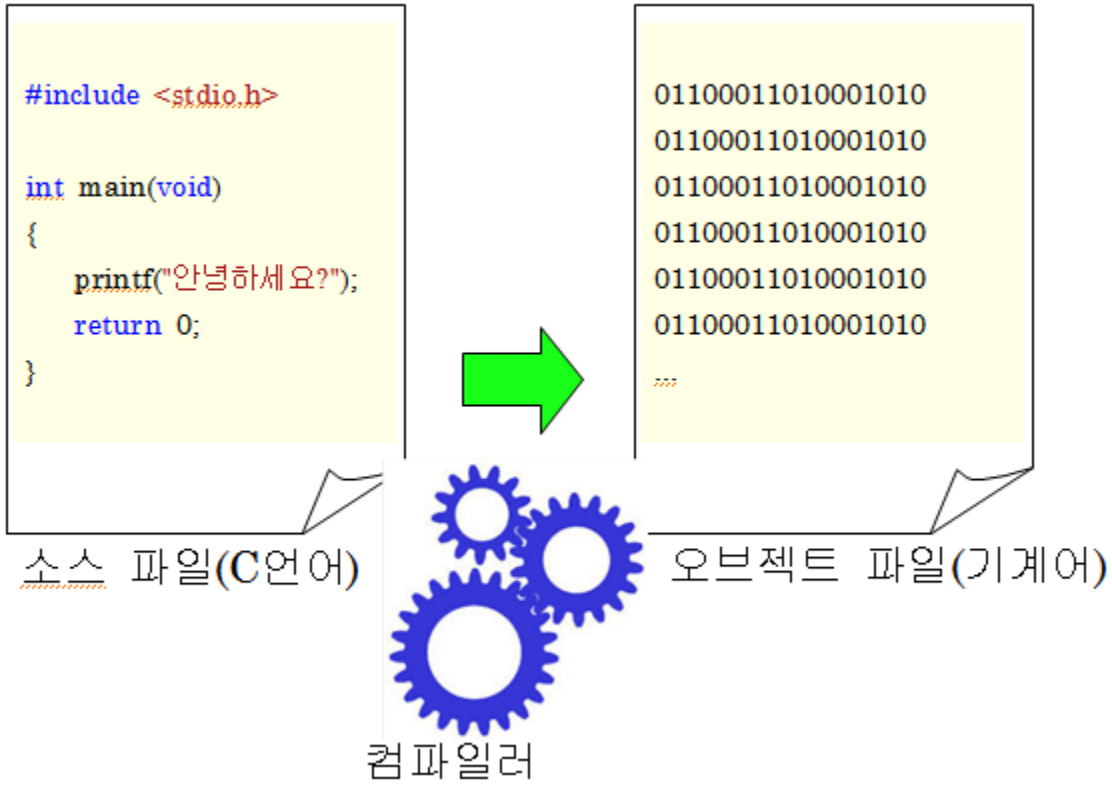
- CPU의 종류나 하드웨어의 특성에 얽매이지 않는 프로그램을 작성
- C나 C++, Java, Python

기계독립적



컴파일

- 컴파일: 소스 파일을 기계어 파일로 만드는 과정



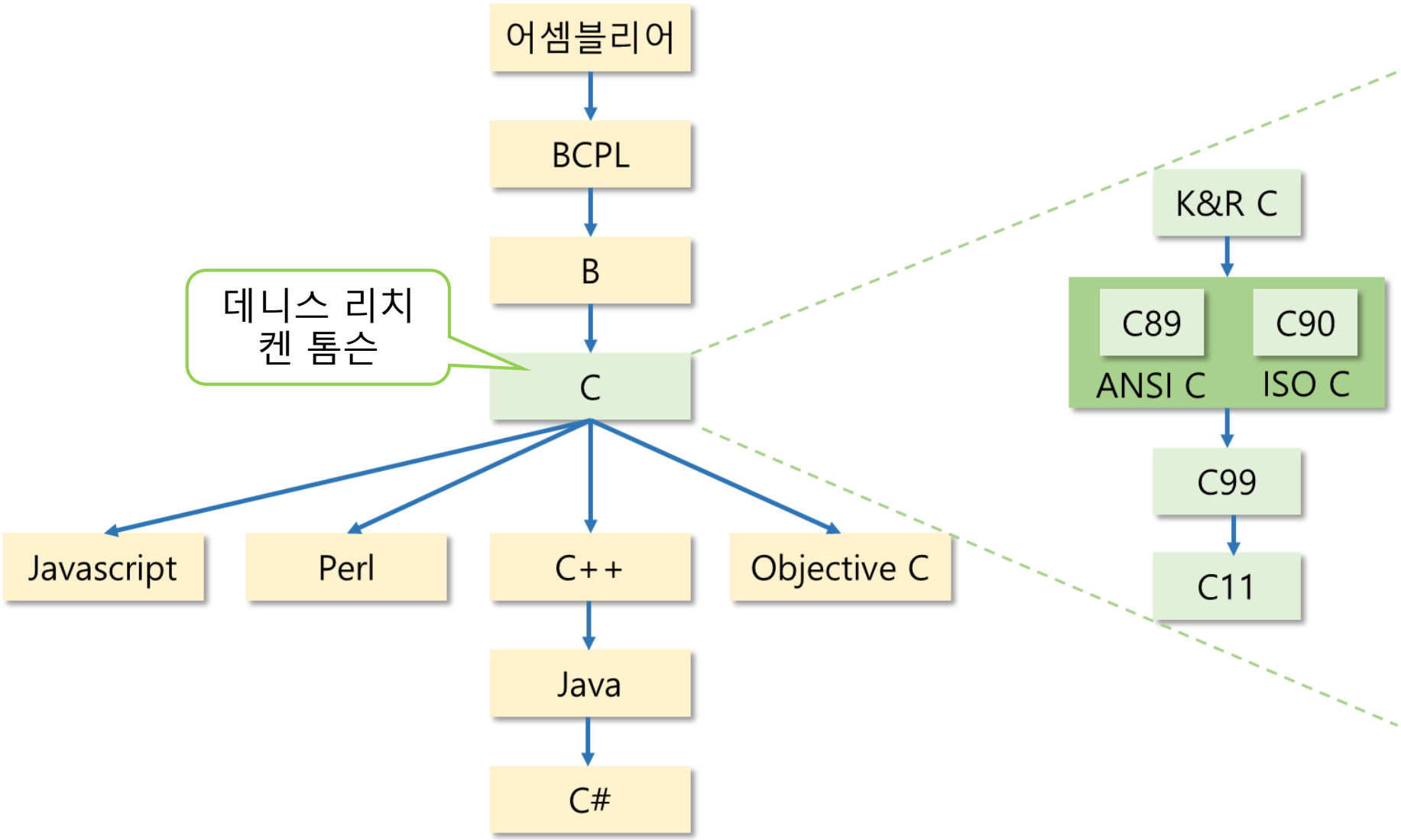
C의 개요

❖ C-Language란?
- 원하는 결과를 얻어내기 위한 Program 작성시 필요한 일종의 언어
- Unix 운영 체제하에서 시스템 프로그래밍을 하기위해 개발된 언어
- 구조적인 언어, 강력한 기능, 빠른 속도

C언어의 종류

- ANSI C (American National Standard Institute) : 표준화 작업,"Standard C"
- Visual C : Microsoft 사에서 개발
- Borland C : Microsoft 사에서 개발

C 언어의 역사


어셈블리어
BCPL
B
데니스 리치
켄 톰슨
C
Javascript
Perl
C++
Objective C
Java
C#
K&R C
C89
C90
ANSI C
ISO C
C99
C11

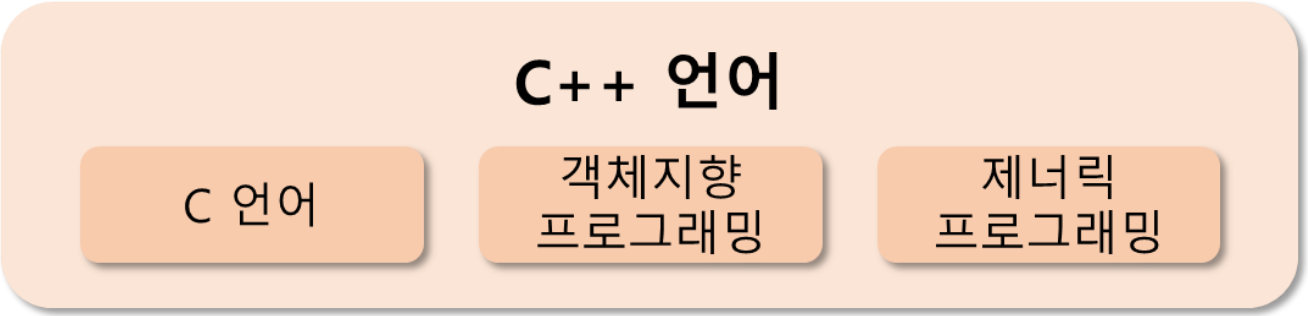
C 언어의 역사

❖ C 언어의 역사

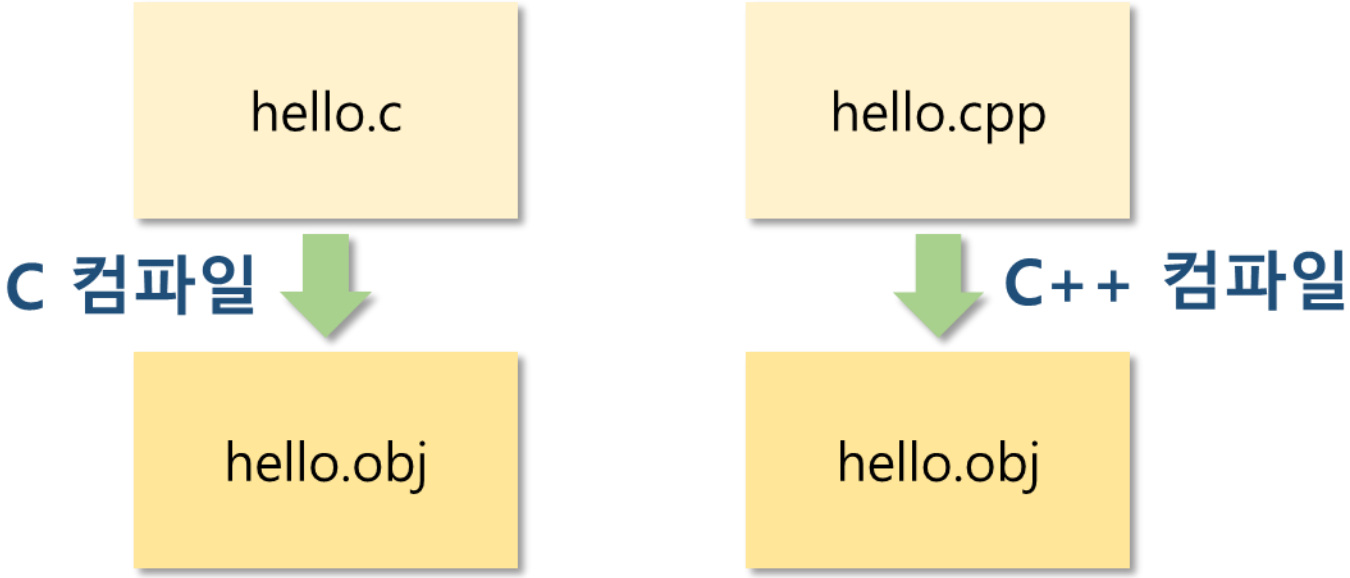
- ALGOL60 (ALGOrithmic Language): 1960년 국제 위원회에서 발표
- CPL (Combined Programing Language) :1963년 영국 캠브리지 런던 대학
- BCLP (Basic CLP) : 영국 캠브리지 런던대학
- B : 미국 AT&T Bell 연구소의 Ken Tompson, 1970년 발표
- C : 미국 AT&T Bell 연구소의 Dennis Ritche, 1972
- C++ : 표준 C에서 OOP(Object Oriented Programing, 객체 지향 프로그램)의 개념 추가

C++ 언어

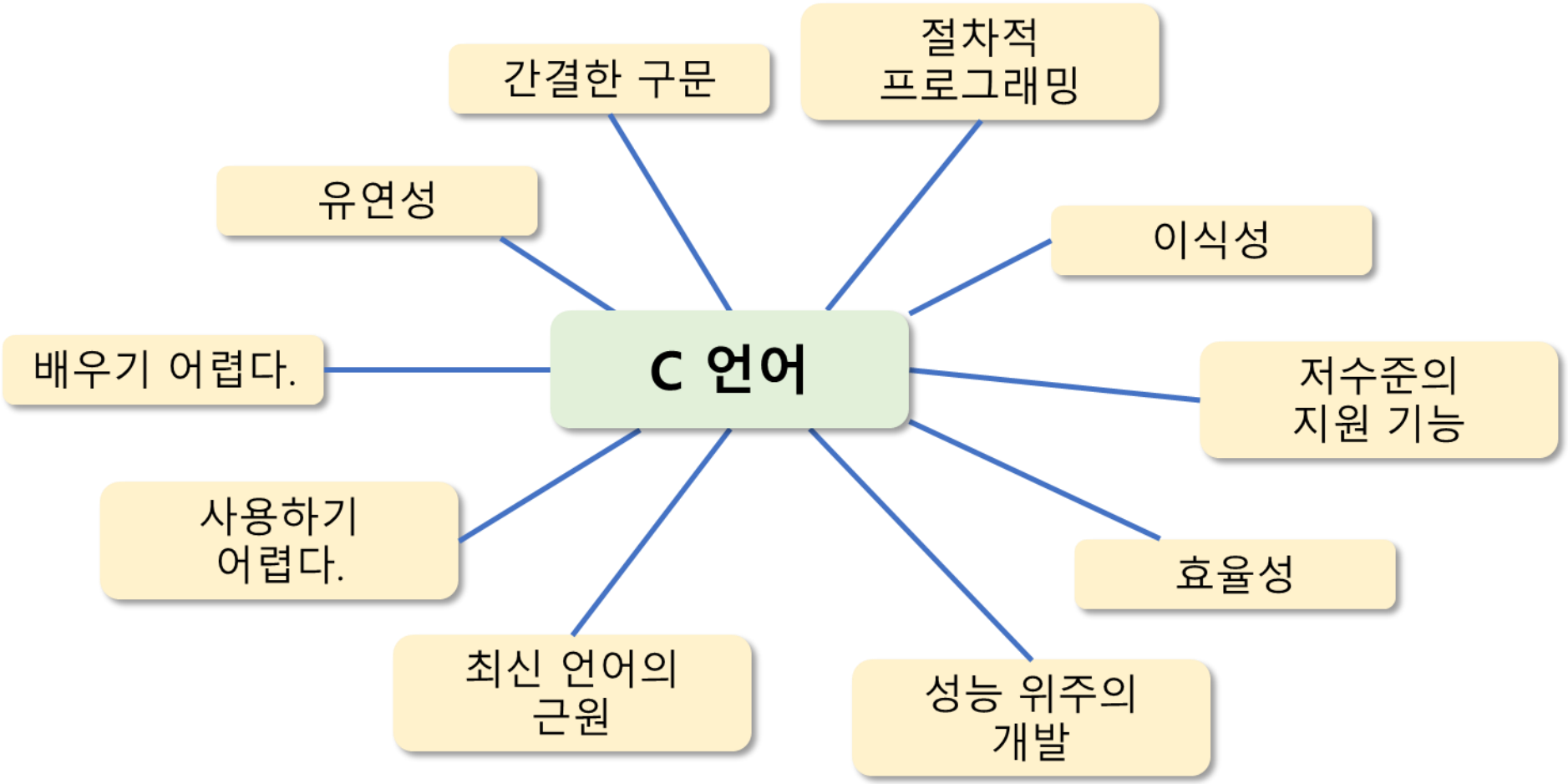
- C/C++ 컴파일러



- 소스 파일 확장자에 따라 C 또는 C++ 컴파일 수행



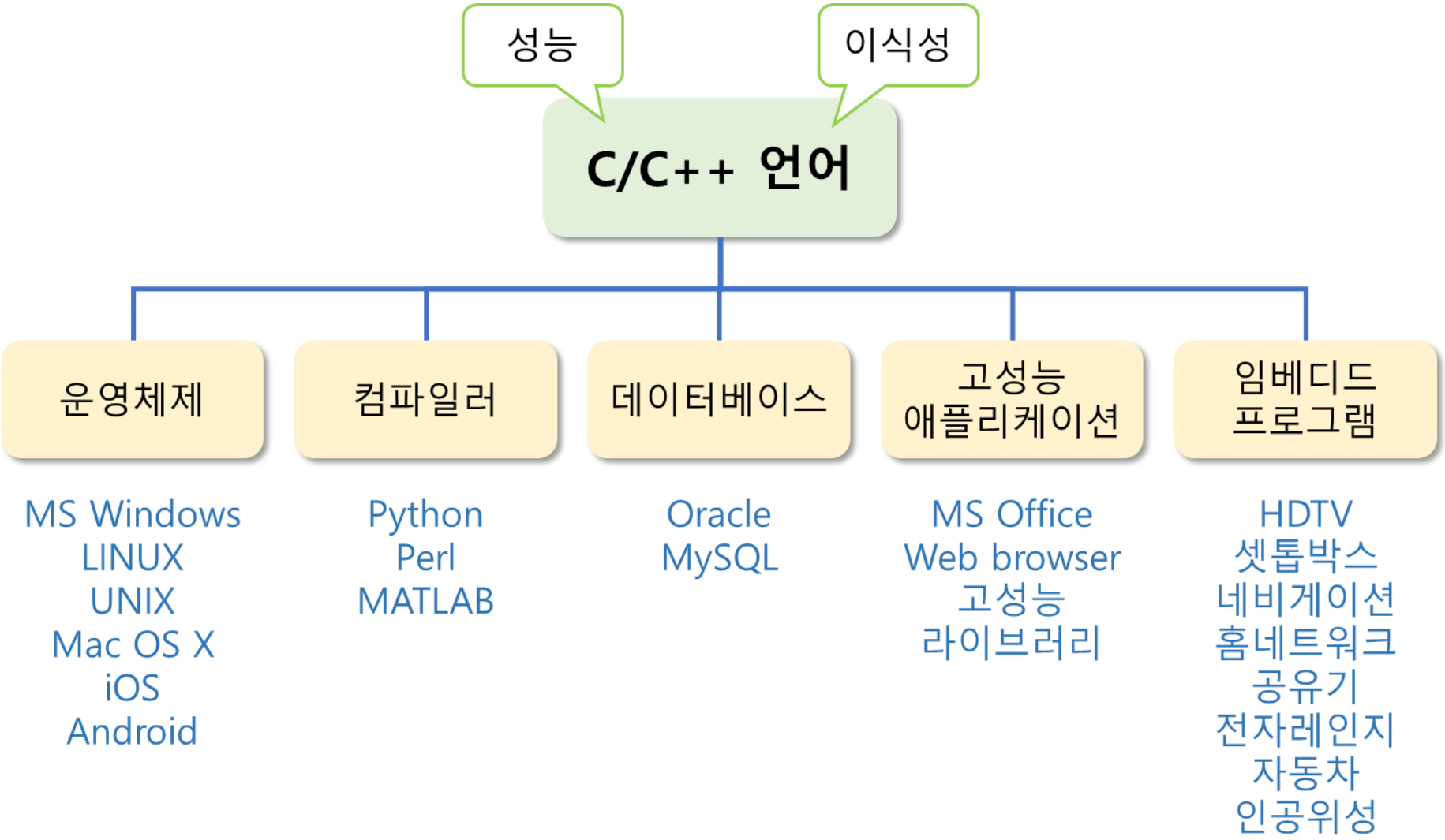
C 언어의 특징


C 언어
간결한 구문
절차적 프로그래밍
유연성
이식성
배우기 어렵다.
저수준의 지원 기능
사용하기 어렵다.
효율성
최신 언어의 근원
성능 위주의 개발

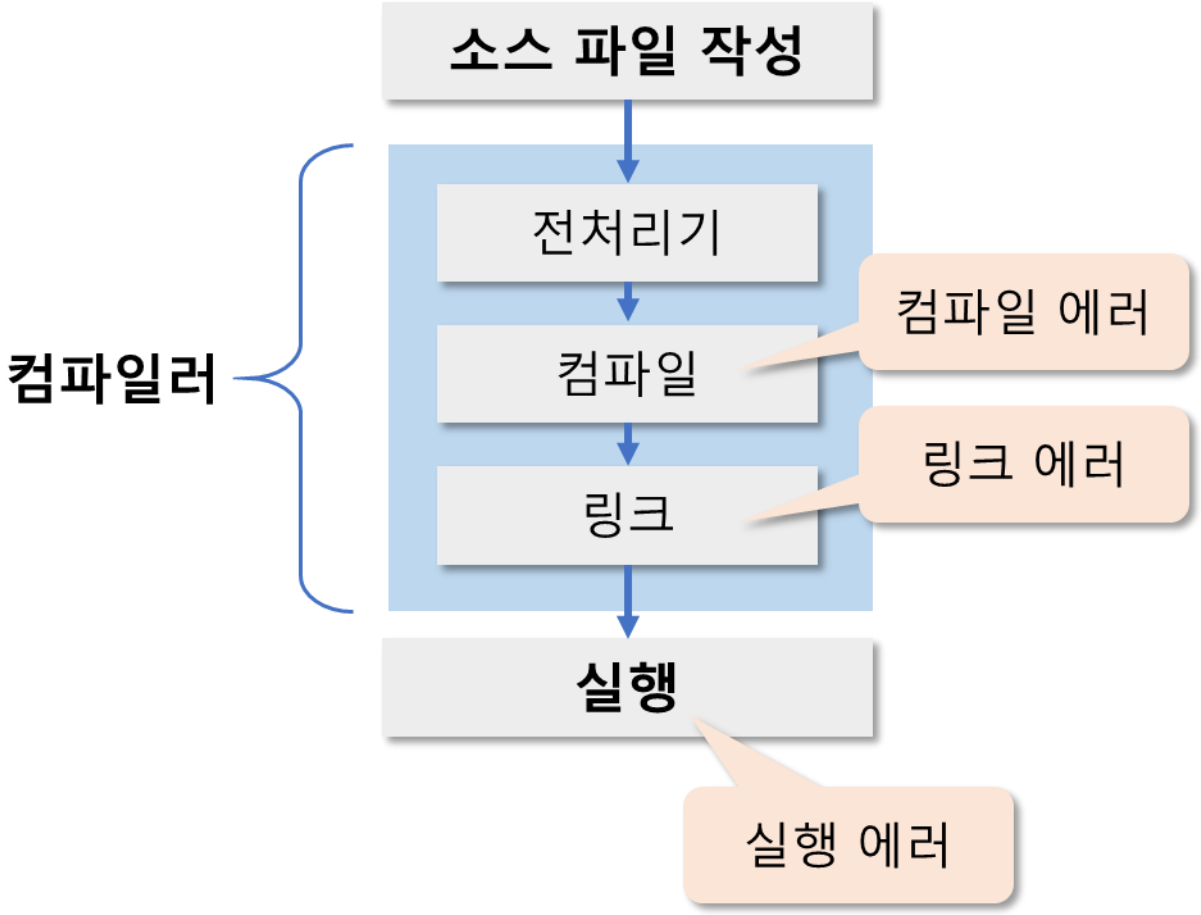
C 언어를 배워야 하는 이유

- ❖프로그래머들 사이에서 일종의 공용어 역할
- ❖대부분의 최신 언어(C++, Java, C#, Javascript, Perl 등)는 C 언어로부터 파생
  - C 언어를 알면 다른 언어를 배우는 것이 쉬워진다
- ❖프로그래밍의 기본 개념을 이해하는 데도 유용
  - 메모리 구조나 함수의 개념, 사용자 정의형, 제어 구조 이해
  - 프로그램의 내부 동작 원리 이해

C 언어의 활용 분야


성능
이식성
C/C++ 언어
운영체제
컴파일러
데이터베이스
고성능 애플리케이션
임베디드 프로그램
MS Windows
LINUX
UNIX
Mac OS X
iOS
Android
Python
Perl
MATLAB
Oracle
MySQL
MS Office
Web browser
고성능
라이브러리
HDTV
셋톱박스
네비게이션
홈네트워크
공유기
전자레인지
자동차
인공위성

일반적인 C 프로그램 개발 과정 (1/4)

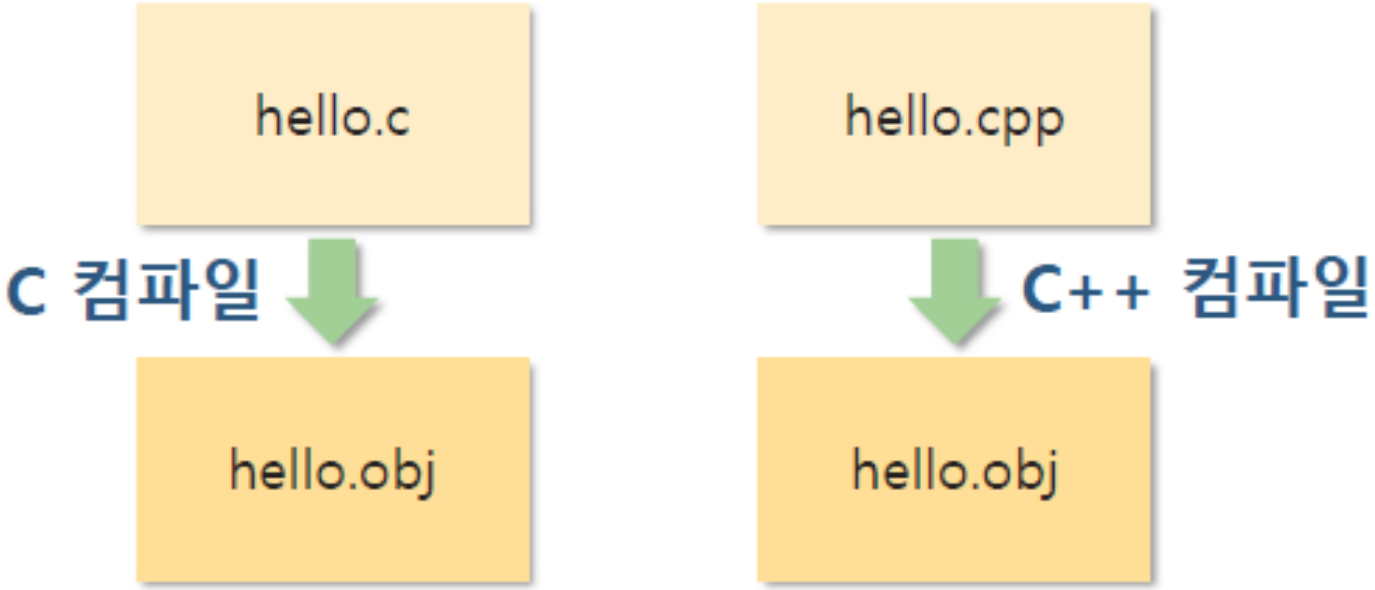


일반적인 C 프로그램 개발 과정 (2/4)

❖소스 파일 작성 → 컴파일 → 실행

❖소스 파일 작성

- 간단한 텍스트 편집기를 이용하거나,
- 통합 개발 환경(IDE)이 제공하는 소스 코드 편집기를 사용해서 작성한다
- C/C++ 컴파일러에서 C 컴파일을 하려면 .c 확장자를 가진 파일로 저장



일반적인 C 프로그램 개발 과정 (3/4)

❖컴파일

- 전처리기 → 구문 분석 → 코드 생성 → 링크
- 전처리기는 소스 파일을 컴파일하기 위해 준비한다
- 구문 분석에서는 소스 코드가 C 문법에 맞게 작성되었는지 검사한다
  - 잘못된 부분에 대하여 컴파일 에러 발생
- 컴파일 에러가 없으면 각각의 소스 파일마다 별개의 오브젝트 코드가 생성된다 → 오브젝트 파일
- 링커는 여러 개의 오브젝트 파일과 라이브러리를 링크해서 실행 파일을 생성한다
  - 링크하면서 문제가 있으면 링크 에러 발생
- 컴파일/링크 에러가 발생하면 소스 파일을 수정하고 컴파일하는 과정을 반복한다

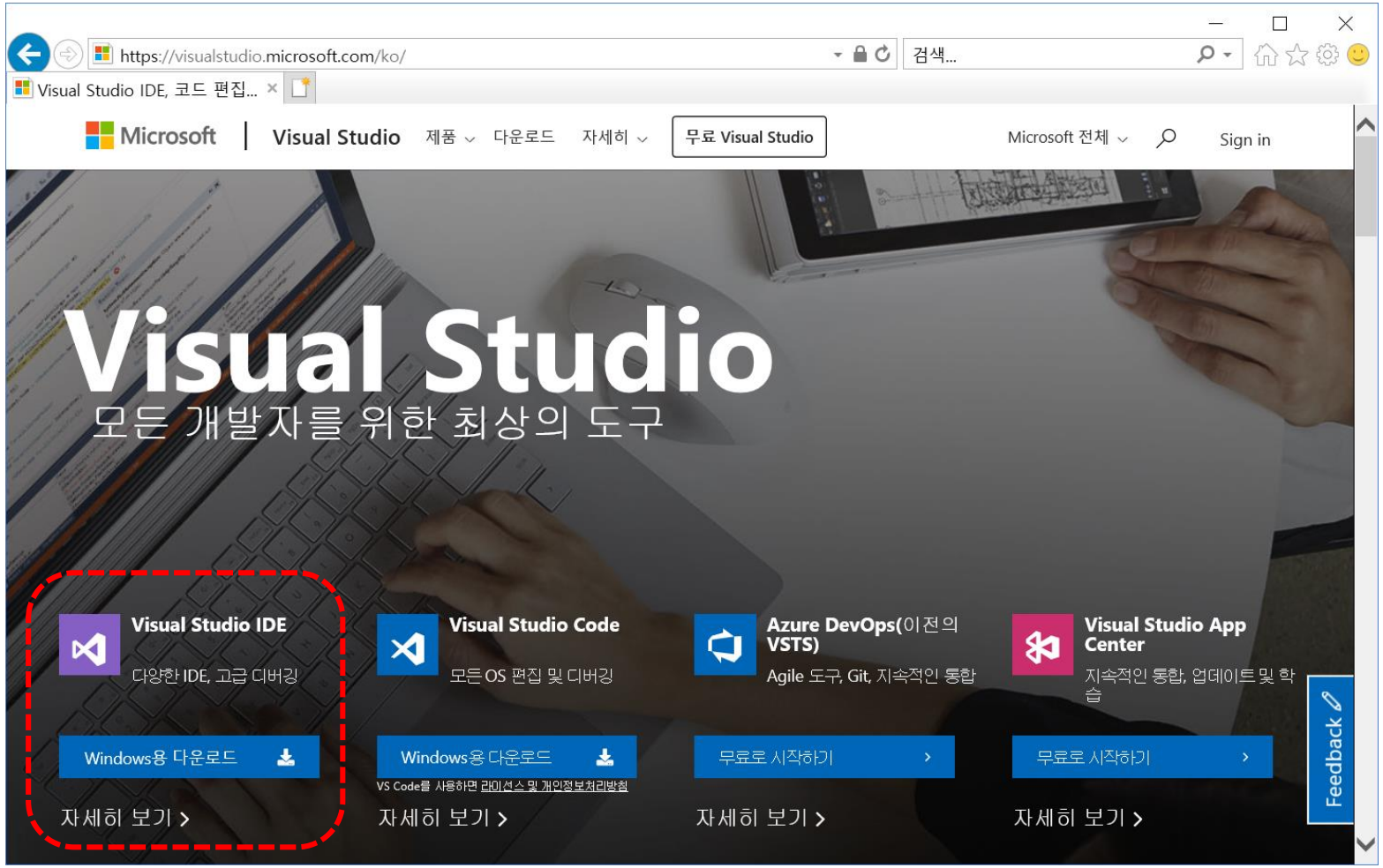
일반적인 C 프로그램 개발 과정 (4/4)

❖실행

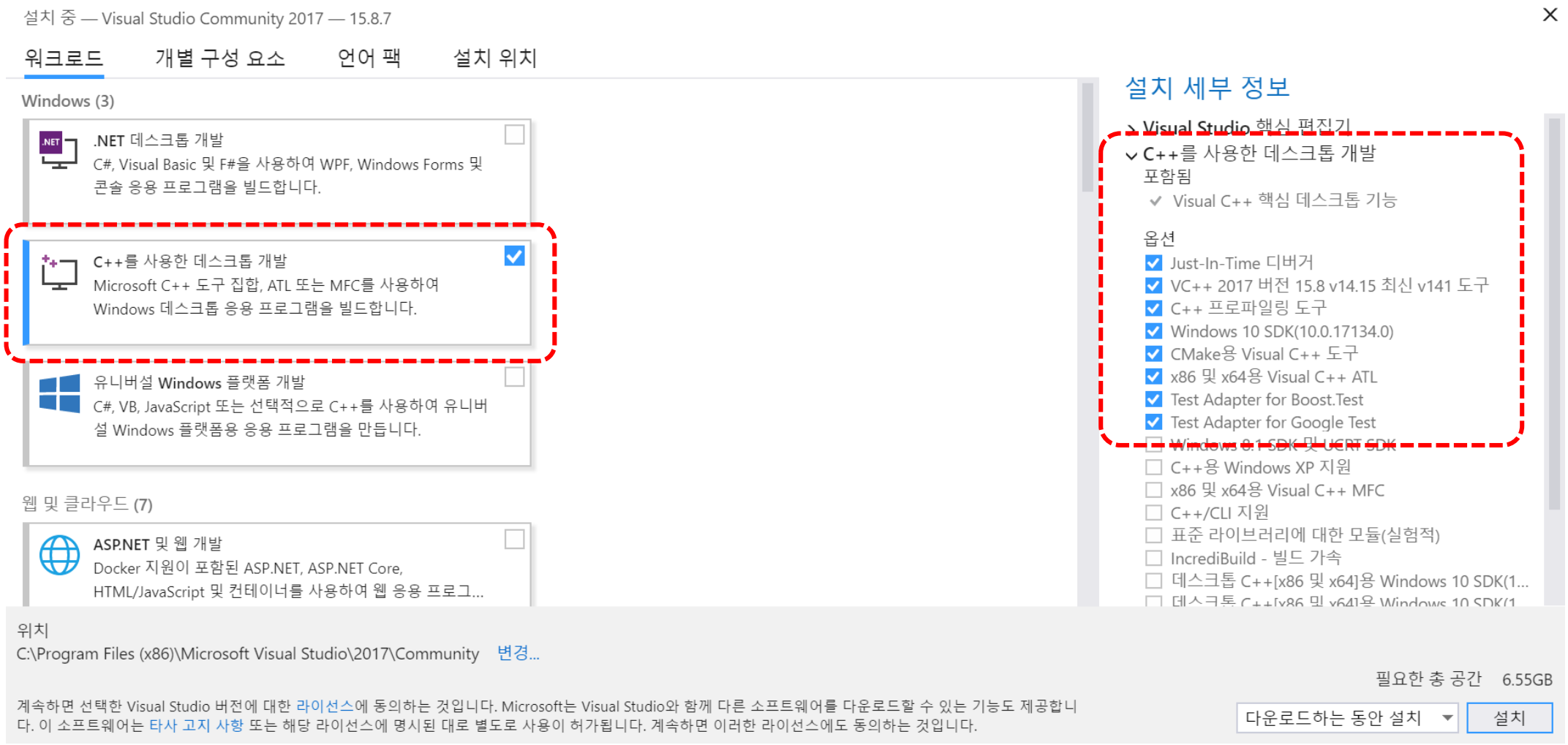
- 링크 결과 생성된 실행 파일을 실행해서 올바른 결과를 얻을 수 있는지 확인한다
  - 프로그램이 잘못된 실행 결과를 생성하거나 실행 중에 프로그램이 죽는 경우에 실행 에러 발생
  - 실행 에러는 프로그램의 논리가 잘못되어 발생하는 에러
- 디버깅 : 프로그램의 실행 흐름이 올바르게 진행되는지, 프로그램 내에서 사용된 수식의 값이 맞는지 등을 살펴봄으로써 실행 에러를 찾아서 고치는 과정

Visual Studio 다운로드

- https://visualstudio.microsoft.com/



Visual Studio 설치

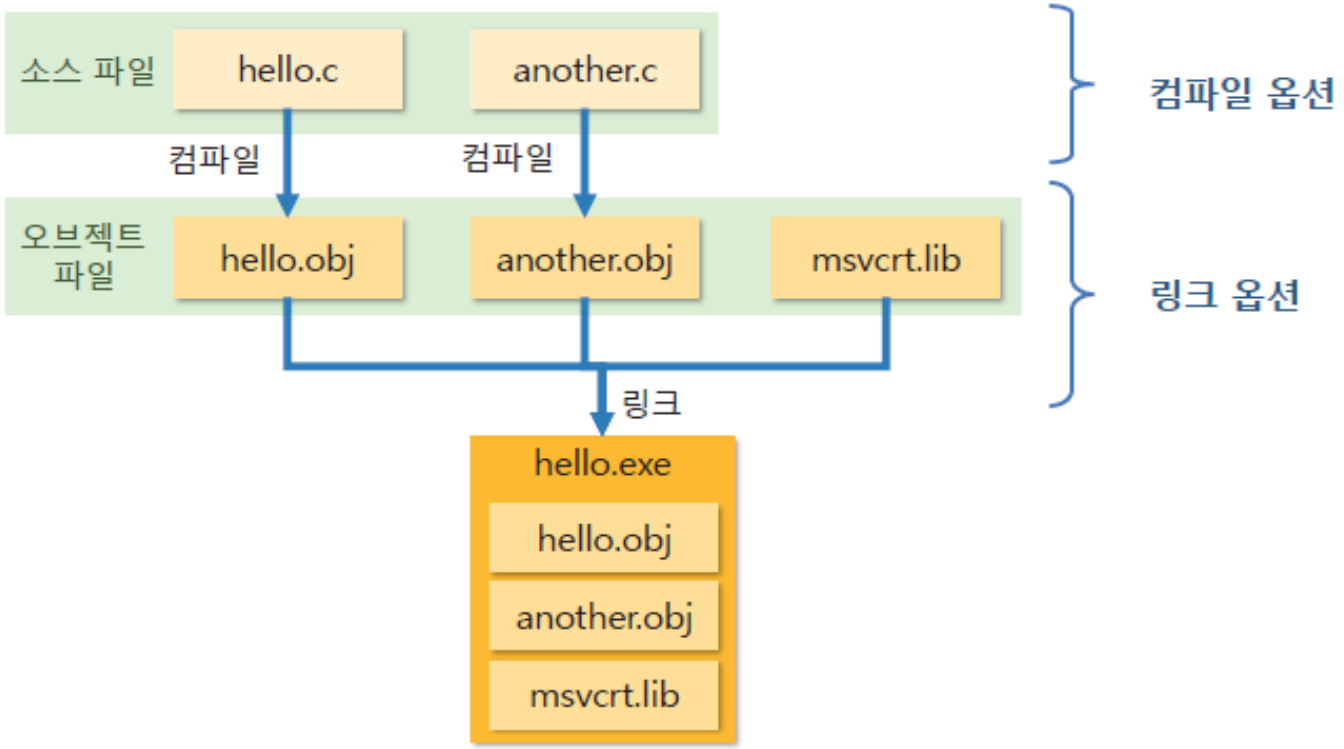


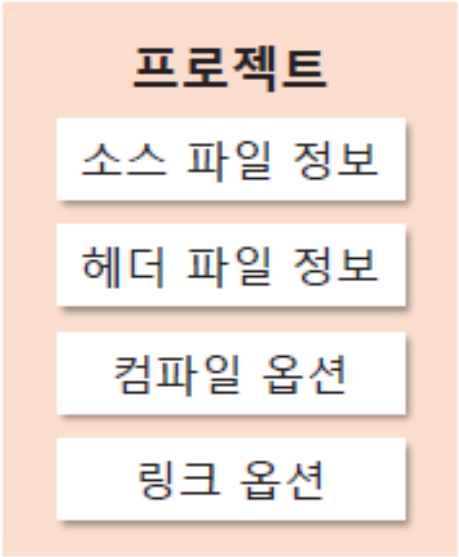
Visual Studio를 사용하려면 Microsoft 계정이 필요하다.

프로젝트와 솔루션 (1/2)

❖프로젝트 (.vcxproj)

- 소스 파일 및 헤더 파일에 대한 정보
- 소스 파일을 컴파일할 때 사용되는 컴파일 옵션
- 오브젝트 파일이나 라이브러리를 링크할 때 사용되는 링크 옵션

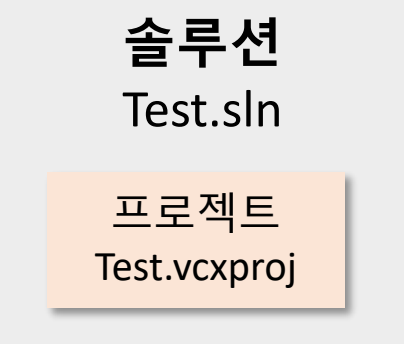




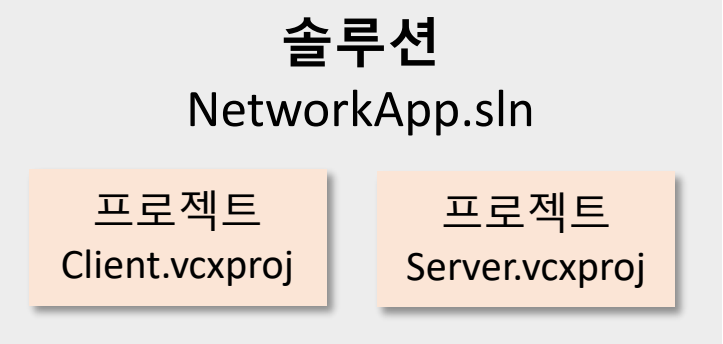
프로젝트와 솔루션 (2/2)

❖솔루션 (.sln)

- 서로 관련된 프로젝트들을 함께 관리하기 위해 솔루션을 사용한다
- 모든 프로젝트가 반드시 솔루션에 포함되어야 한다

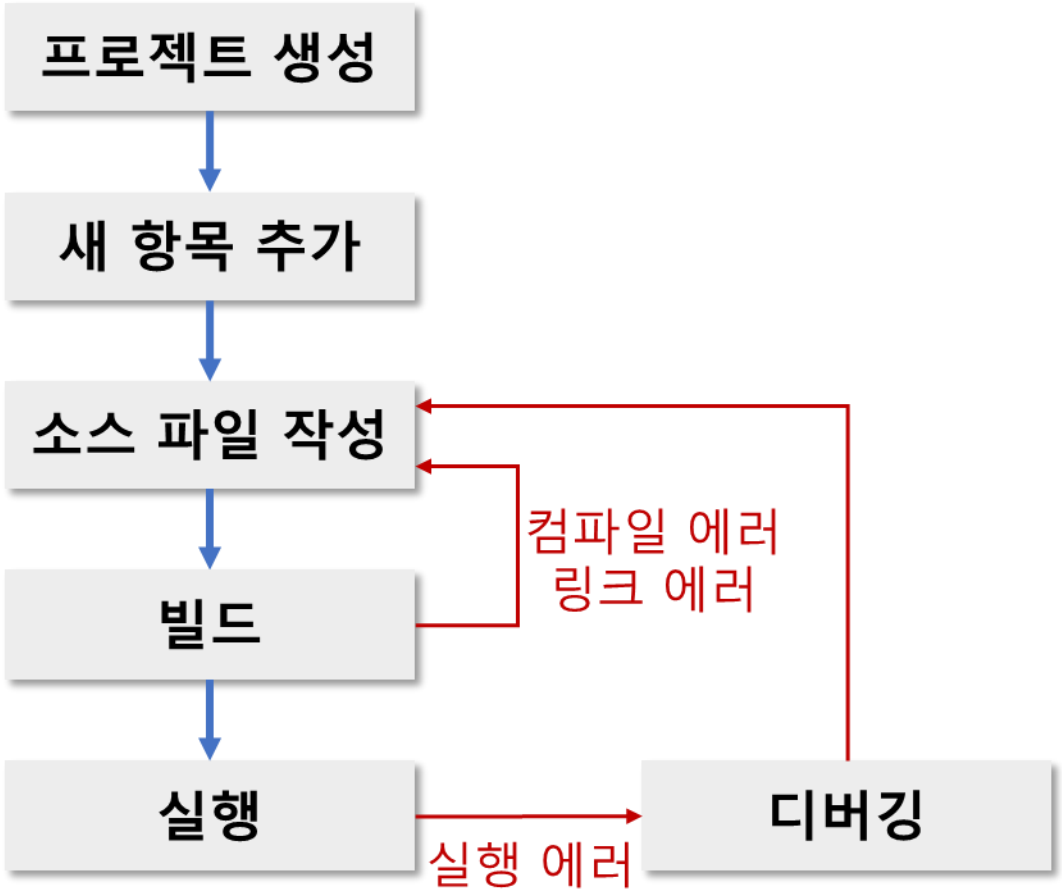


솔루션 내에
프로젝트가
하나인 경우



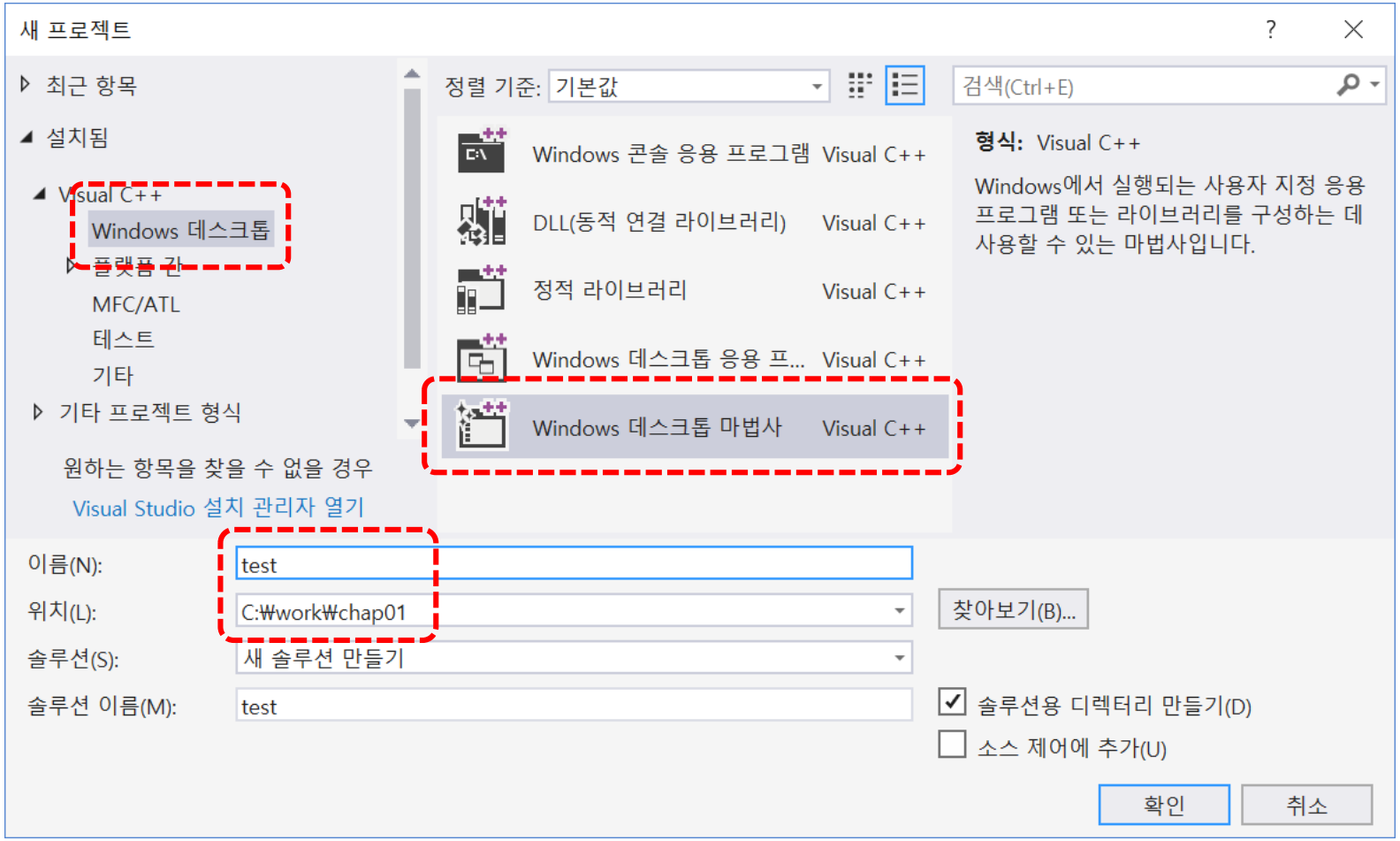
솔루션 내에
프로젝트가
여러 개인 경우

Visual Studio를 이용한 C 프로그램 개발 과정

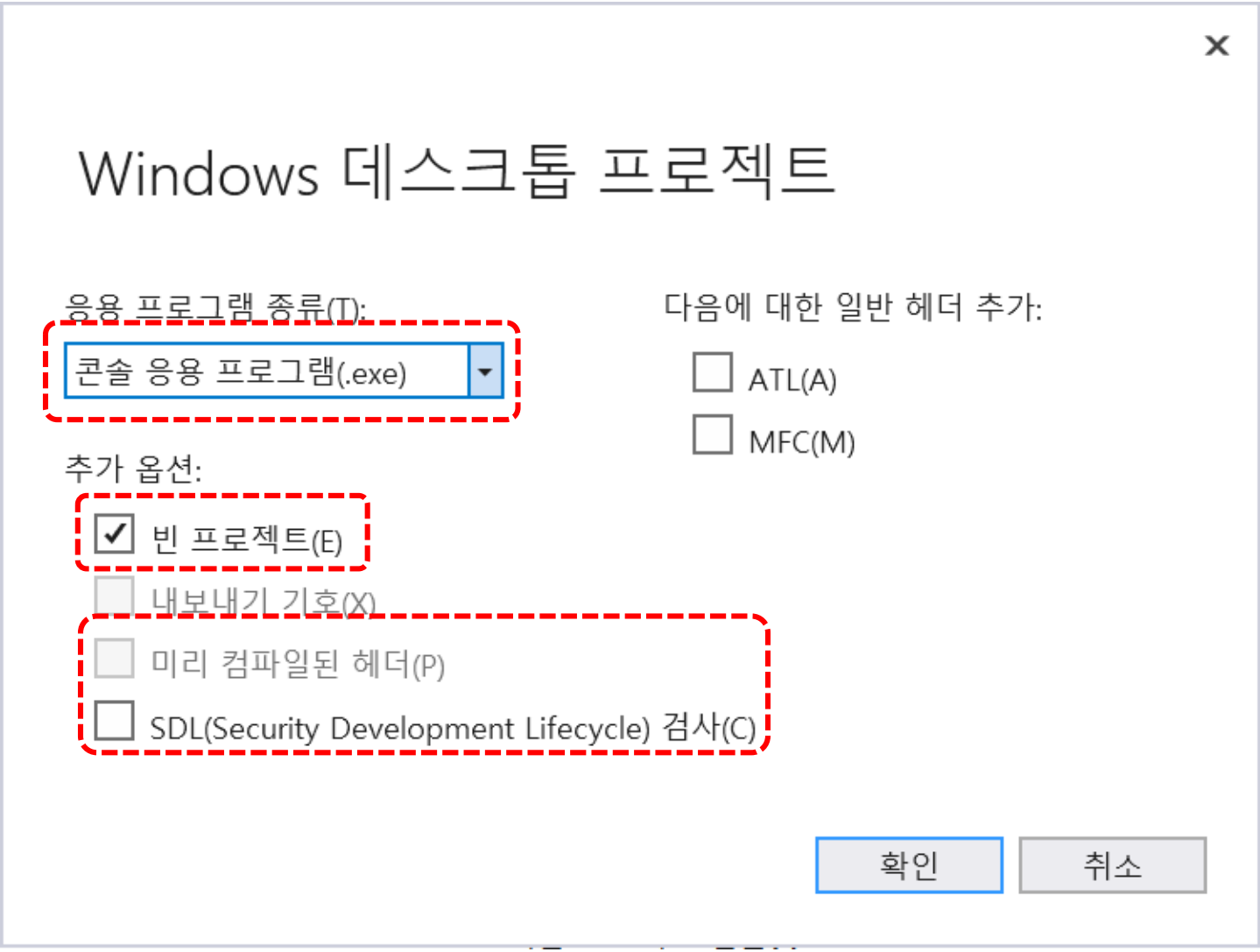


프로젝트 생성

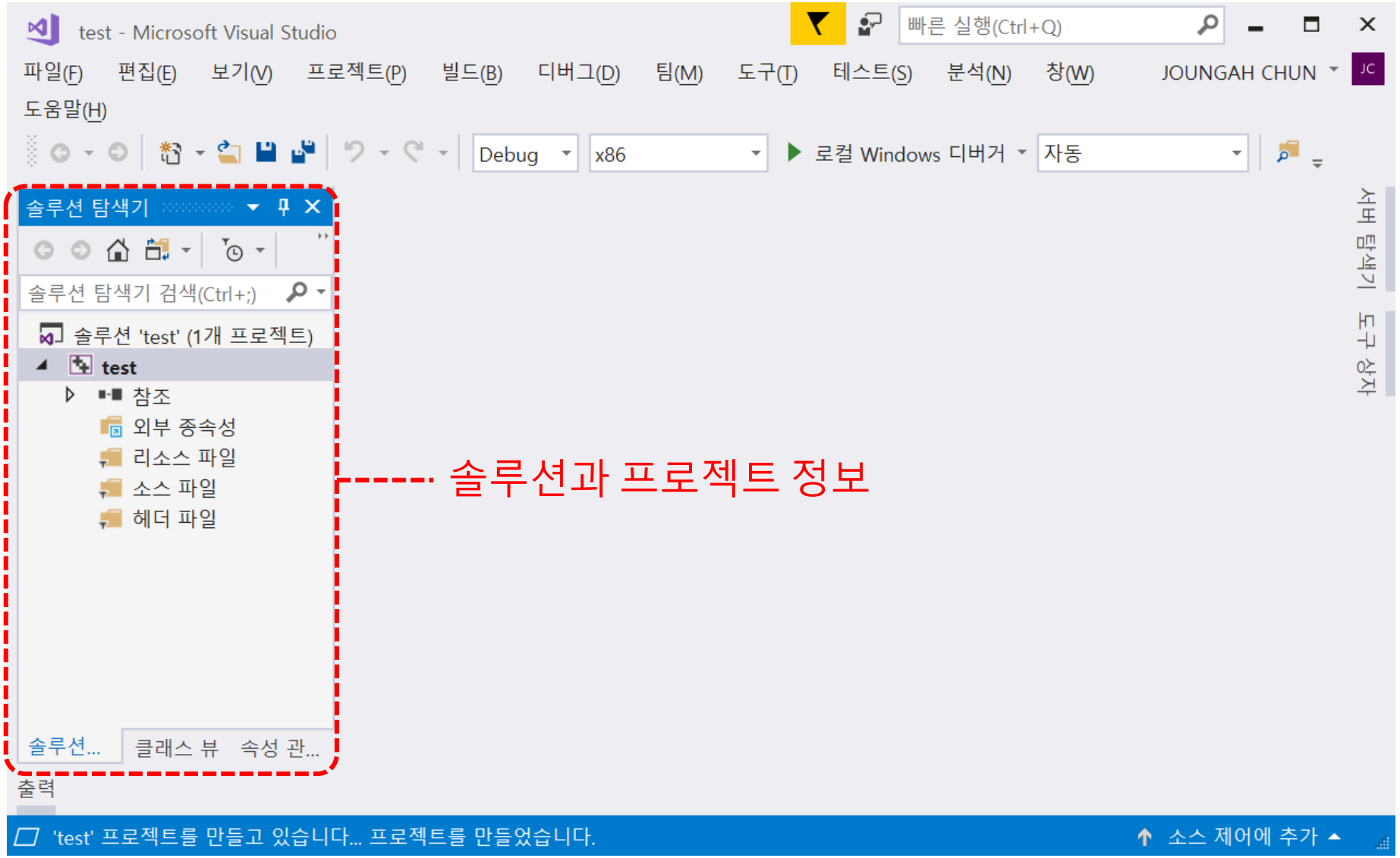
- [파일] → [새로 만들기] → [프로젝트]



Windows 데스크톱 프로젝트



빈 프로젝트 생성

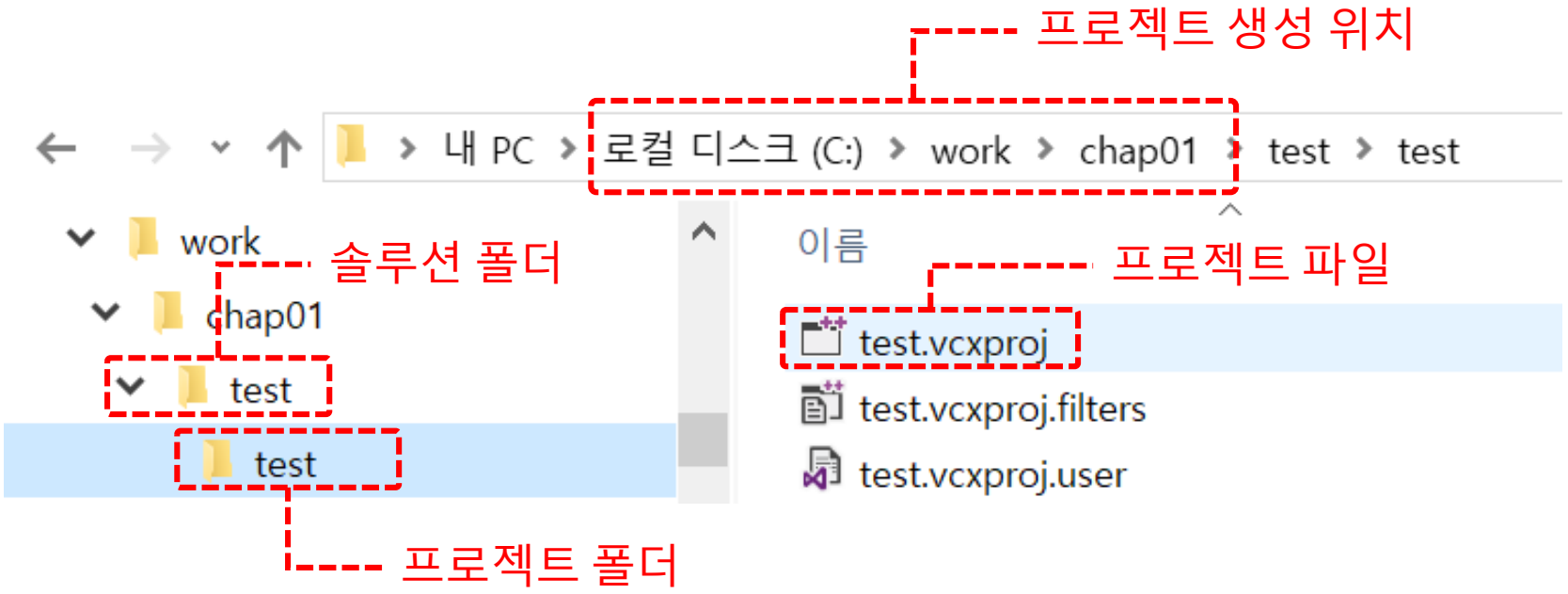


솔루션과 프로젝트 정보

솔루션 폴더와 프로젝트 폴더

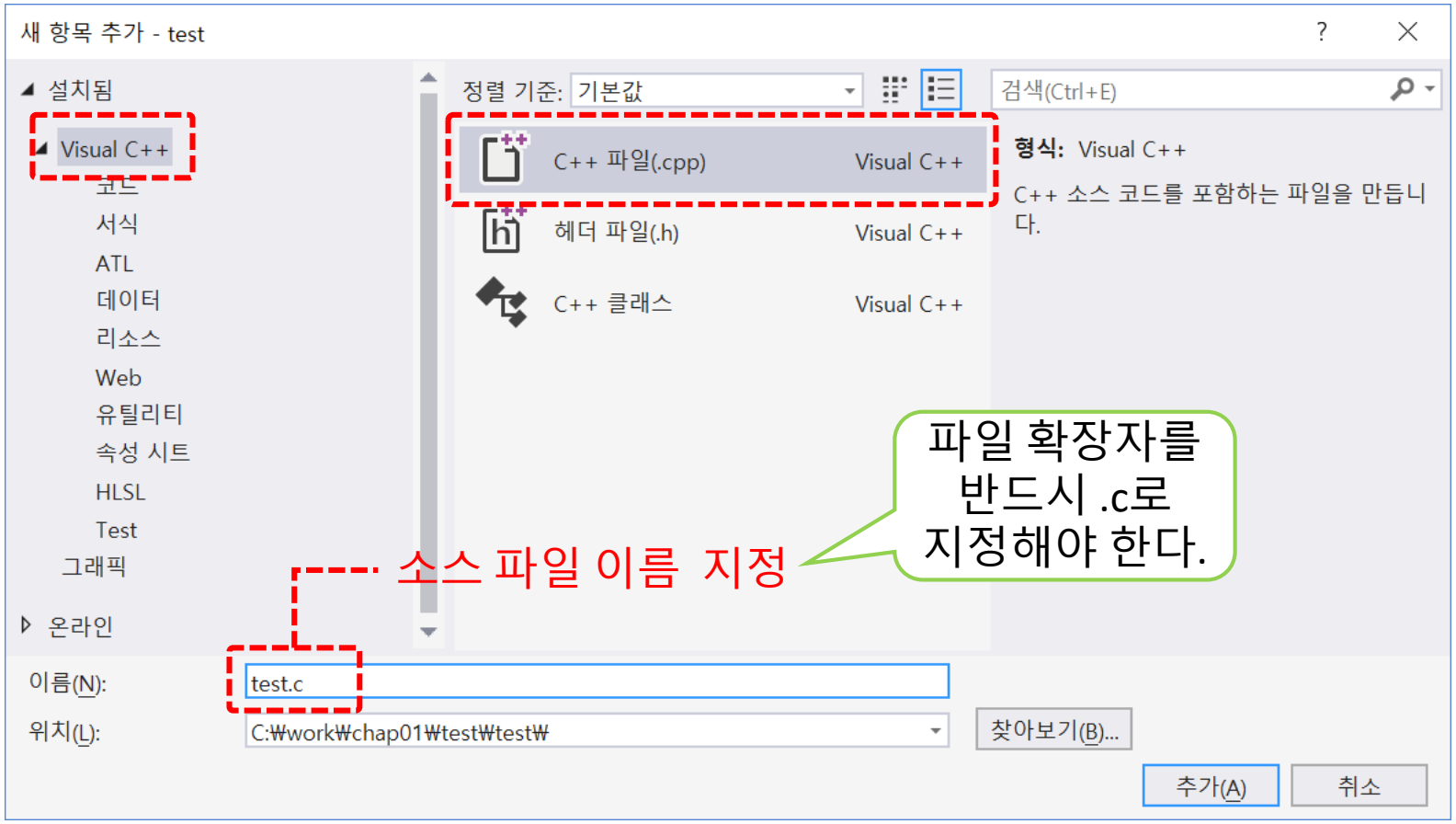
❖프로젝트를 생성하면, 디폴트로 프로젝트 이름과 같은 이름의 솔루션이 생성된다

- 솔루션 폴더와 솔루션 폴더 안에 프로젝트 폴더 생성
- 솔루션 폴더에는 솔루션 파일인 test.sln이, 프로젝트 폴더에는 프로젝트 파일인 test.vcxproj가 생성된다



새 항목 추가

- 솔루션 탐색기에서 test 프로젝트를 클릭한 다음 [프로젝트] → [새 항목 추가]



소스 파일 작성

test - Microsoft Visual Studio

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

Debug x86 로컬 Windows 디버거 자동

솔루션 탐색기

솔루션 탐색기 검색(Ctrl+;)

솔루션 'test' (1개 프로젝트)

test

참조

외부 종속성

리소스 파일

소스 파일

test.c

헤더 파일

test.c*

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

소스 파일 작성

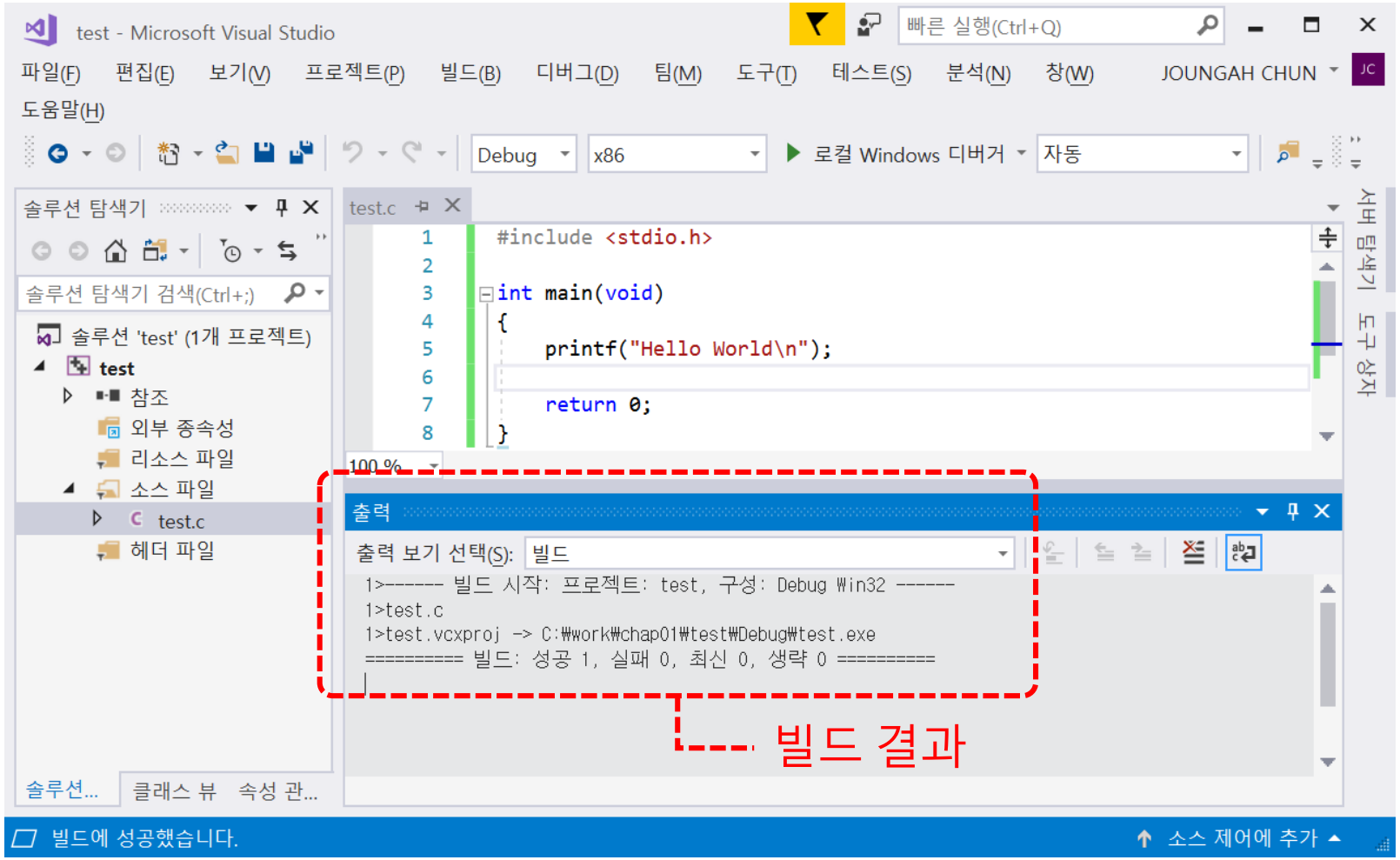
100 %

출력

준... 줄: 6 열: 9 문자: 9 INS 소스 제어에 추가

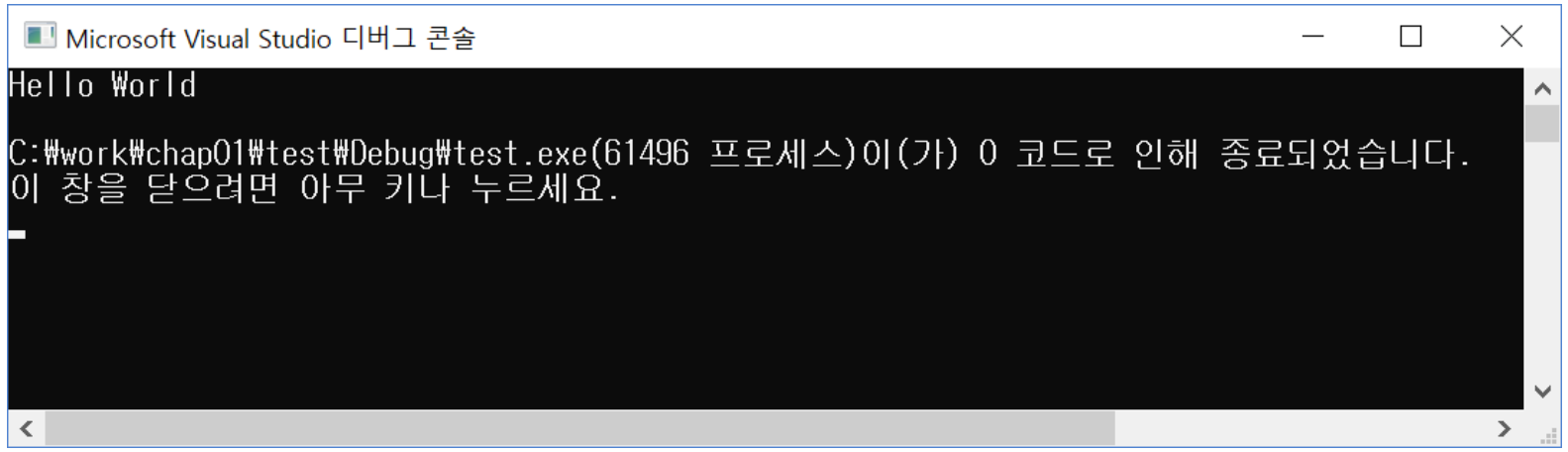
빌드

- [빌드] → [솔루션 빌드]


test - 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)
JOUNGAH CHUN
Debug
x86
로컬 Windows 디버거
자동
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'test' (1개 프로젝트)
test
참조
외부 종속성
리소스 파일
소스 파일
test.c
헤더 파일
test.c
#include <stdio.h>
int main(void)
{
printf("Hello World\n");
return 0;
}
100 %
출력
출력 보기 선택(S): 빌드
1>------ 빌드 시작: 프로젝트: test, 구성: Debug Win32 ------
1>test.c
1>test.vcxproj -> C:₩work₩chap01₩test₩Debug₩test.exe
========== 빌드: 성공 1, 실패 0, 최신 0, 생략 0 ==========
빌드 결과
솔루션... 클래스 뷰 속성 관...
서버 탐색기 도구 상자
빌드에 성공했습니다.
소스 제어에 추가

실행

- 컴파일 에러와 링크 에러가 없으면 프로젝트 빌드 결과 실행 파일이 생성된다
- [디버그] → [디버그하지 않고 시작]



- 직접 명령 프롬프트에 실행 파일의 완전 경로명을 입력한다
  c:\work\chap01\test\Debug\test.exe

Visual Studio 기능

기능	메뉴 항목	단축키
새 프로젝트 생성	[파일] → [새로 만들기] → [프로젝트]	Ctrl+Shift+N
새 항목 추가	솔루션 탐색기에서 프로젝트 이름 선택 후 [프로젝트] → [새 항목 추가]	Ctrl+Shift+A
기존 항목 추가	솔루션 탐색기에서 프로젝트 이름 선택 후 [프로젝트] → [기존 항목 추가]	Shift+Alt+A
빌드	[빌드] → [솔루션 빌드]	F7
실행	[디버그] → [디버그하지 않고 시작]	Ctrl+F5
솔루션/프로젝트 열기	[파일] → [열기] → [프로젝트/솔루션]	Ctrl+Shift+O
소스 파일 전체 자동 들여쓰기/띄어쓰기	소스 파일 편집기 클릭 후 [편집] → [고급] → [문서 서식]	Ctrl+K, Ctrl+D
블록 자동 들여쓰기/띄어쓰기	블록 선택 후 [편집] → [고급] → [선택 영역 서식]	Ctrl+K, Ctrl+F
디버깅 시작	[디버그] → [디버깅 시작]	F5
한 단계씩 코드 실행	[디버그] → [한 단계씩 코드 시작]	F11
프로시저 단위 실행	[디버그] → [프로시저 단위 실행]	F10
중단점 설정/해제	[디버그] → [중단점 설정/해제]	F9

질의 및 응답

참고문헌

- 천정아, 『Core C Programming』, 연두에디션(2019)
- C가 보이는 그림책, ANK Co., Ltd. , 성안당 (2018)
- Greg Perry, Dean Miller 『어서와 C언어는 처음이지』, 천인국 옮김, 인피니티북스(2015)
- KELLEY (역 : 김명호 외), 『A Book on C』, 홍릉과학출판사 (2003)
- 윤성우, 『 열혈 C 프로그래밍』, 오렌지미디어
- 천인국, 『 쉽게 풀어쓴 C언어 Express』, 생능출판사
- 서현우, 『 뇌를 자극하는 C 프로그래밍』, 한빛미디어
- 강성수, 『 쾌도난마 C프로그래밍』, 북스홀릭
- 고응남, 『 C프로그래밍 기초와 응용실습』, 정익사