

1주차 강의 자료

# 객체 지향 프로그래밍 I

담당 : 유영란

# 과목 개요

- ▶ 본 강좌는 “**객체 지향 프로그래밍 I**” 과정입니다.
- ▶ 객체 지향 프로그래밍 언어 중 **C++**로 학습합니다.
- ▶ **C++**언어는 **C**언어의 문법에 객체지향의 개념을 추가한 언어로 완벽한 객체지향언어는 아닙니다.
- ▶ **3주** 동안은 **C**언어의 문법과 **C++**의 입출력 객체 및 개념을 이용한 프로그래밍 연습을 하며 학습과제를 작성합니다.

개정판



눈과 직관만으로도 누구나 쉽게 이해할 수 있는 명품 C++ 강좌

명품

# C++ Programming

황기태 지음

- C++ 언어와 프로그래밍 기술을 쉽게 풀어내는 설명과 그림으로 구성되어 있어 C++ 언어를 쉽게 이해할 수 있다.
- 강조 단어, 소스 하이라이팅, 활동선, 그림, 삽화, 다양한 색 등이 어우러져 책을 읽는 재미와 함께 눈과 직관만으로도 쉽게 이해할 수 있다.
- 틈틈이 셀프 테스트용 Check Time 문제를 두고, 연습문제를 난이도와 함께 이론과 실습 문제들로 구분하여 단계별로 충실히 복습할 수 있다.
- 각 장별로 도전해 볼 수 있는 Open Challenge 문제와 활동 예제 및 실전 연습 거리가 풍부하다.
- 책의 구성구석에 이해를 돕기 위한 설명과 팁이 마련되어 있어 재미를 더해 준다.



#### 홈페이지 제공 자료

영어 단어 파일 word.txt  
예제 및 그림의 소스 코드  
Open Challenge 정답, 설명용 배치 파일  
연습문제 이론 정답(출수면)  
연습문제 실습 정답(답부 문제판)  
C++ 기초 학습을 위한 PDF와 소스 코드

예제 등 소스는 비주얼 스튜디오 2017 프로젝트로 제공



## 교재 소개

- 도서명 : 명품 C++ Programming
- 저자 : 황기태
- 출판사 : 생능 출판사
- 페이지 : 696p
- 출판일 : 2019.

# 강의 계획서

주	해당 장	주 제	비 고
1	01장	오리엔테이션(과정 및 과목 소개) 및 객체지향언어의 개요 및 개발환경	과제 제시
2	01장	C++프로그래밍의 작성과 실행, C언어와의 차이점 및 기초문법 활용예제 작성	
3	02장	C++프로그래밍의 기본 문법 – 제어구조와 배열, 함수와 문자열	
4	03장	클래스와 객체	
5	04장	객체 포인터와 객체 배열, 객체의 동적 생성	
6	05장	함수의 참조, 복사 생성자	
7	06장	함수 중복과 static 멤버	
8	중간고사		
9	07장	프렌드 함수와 연산자 중복	
10	08장	상속	
11	09장	가상 함수와 추상 클래스	
12	10장	템플릿과 표준 템플릿 라이브러리(STL)	수시시험
13	11장	C++ 입출력 시스템	과제 제출
14	13장	예외처리	
15	기말고사		

과제: 클래스와 객체 개념을 활용한 프로그래밍 작성

# 학습 목표

- ▶ C++ 프로그래밍의 기본구조를 활용한 프로그램을 테스트할 수 있다.
- ▶ 객체지향 언어의 특징을 이용하여 프로그램을 테스트할 수 있다.
  - ▶ 클래스의 정의 및 객체의 생성하는 방법을 익힌다.
  - ▶ 캡슐화된 객체지향 프로그래밍을 할 수 있다.
  - ▶ 배열과 포인터와의 관계를 학습하고,
  - ▶ 참조를 활용한 프로그램을 작성할 수 있다.
  - ▶ 연산자 중복과 함수의 중복을 이해하여 다형성을 표현한 프로그램을 구현할 수 있다.
  - ▶ 상속의 개념을 이해하여 클래스간의 상속 관계를 갖도록 정의하고, 부모클래스와 자식 클래스의 생성자 메커니즘을 이용한 프로그램을 작성할 수 있다.
  - ▶ 가상 함수의 필요성과 개념에 대해 학습하고, 다형성을 제공하는 가상함수 프로그램을 작성할 수 있다.
  - ▶ C++ 입출력 시스템을 이용하여 프로그램과 입출력 장치를 연결하는 프로그램을 실행할 수 있다.
  - ▶ 파일 입출력에 필요한 스트림 클래스를 살펴보고, 이진 파일과 텍스트파일을 입출력하는 프로그램을 구현할 수 있다.

# 학습 및 평가 방법

## ▶ 수업 방식

- ▶ 이론 + 실습 + 질의응답 + 퀴즈(학습활동, 학습과제)

## ▶ 평가 방식

- ▶ 정기평가 (60%) : 중간고사(30%) + 기말고사(30%)
- ▶ 수시시험(5%)
- ▶ 과제물(10%) : 평가항목(주제 적용, 내용 구성, 제출기한)
- ▶ 출결(20%) : 총 수업시간의 80% 이상 출석
- ▶ 기타(5%) : 수업태도 + 질의응답

## ▶ 최종 평가(상대 평가)

- ▶ 정기평가 + 수시시험 + 과제물 + 출결 + 기타



1장.

C++ 시작

# 목차

1. 소프트웨어의 중요성을 이해한다.
2. C++ 언어의 역사를 이해한다.
3. C++ 언어의 특징을 이해한다.
4. C++ 프로그램의 개발 과정을 설명할 수 있다.
5. C++ 표준 라이브러리에 대해 설명할 수 있다.
6. Visual Studio를 이용하여 C++ 프로그램을 개발하는 과정을 이해한다.



***“Everybody in this country should learn  
how to program a computer.  
because it teaches you how to think.”***

***- Steve Jobs -***

“이 나라에 살고 있는 모든 사람들은 컴퓨터 프로그래밍을 배워야 한다. 왜냐하면 생각하는 방법을 가르쳐 주기 때문이다.”

- 스티브 잡스 -

# 소프트웨어의 중요성

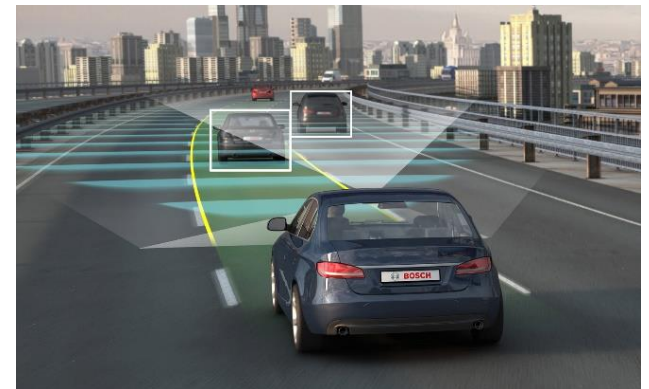
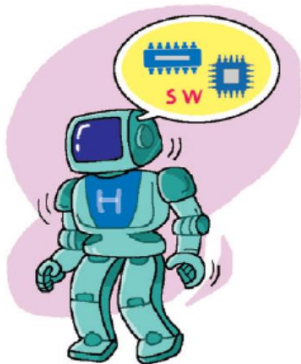
## ■ 소프트웨어 기업의 세상

Software is eating the world.

eBay, Facebook, Groupon, Skype, Twitter, Android, Netflix, Google, Apple, Samsung

## ■ 4차 산업의 핵심에 소프트웨어가 있다

- 자율주행 자동차, AI(구글의 알파고, IBM 왓슨), IoT



# 프로그래밍과 프로그래밍 언어

## ■ 프로그래밍 언어

### ■ 기계어(machine language)

- 0, 1의 이진수로 구성된 언어
- 컴퓨터의 CPU는 본질적으로 기계어만 처리 가능

### ■ 어셈블리어

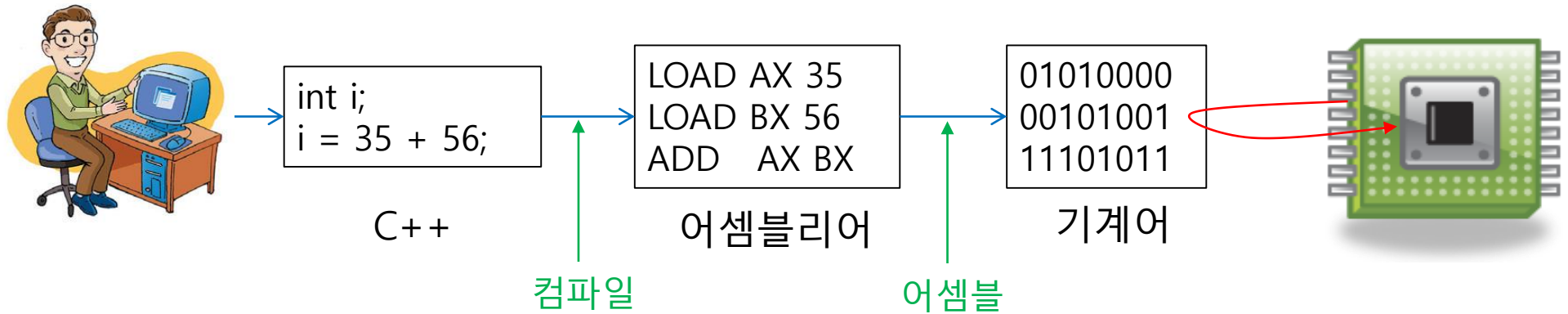
- 기계어의 명령을 ADD, SUB, MOVE 등과 같은 상징적인 니모닉 기호(mnemonic symbol)로 일대일 대응시킨 언어
- 어셈블러 : 어셈블리어 프로그램을 기계어 코드로 변환

### ■ 고급언어

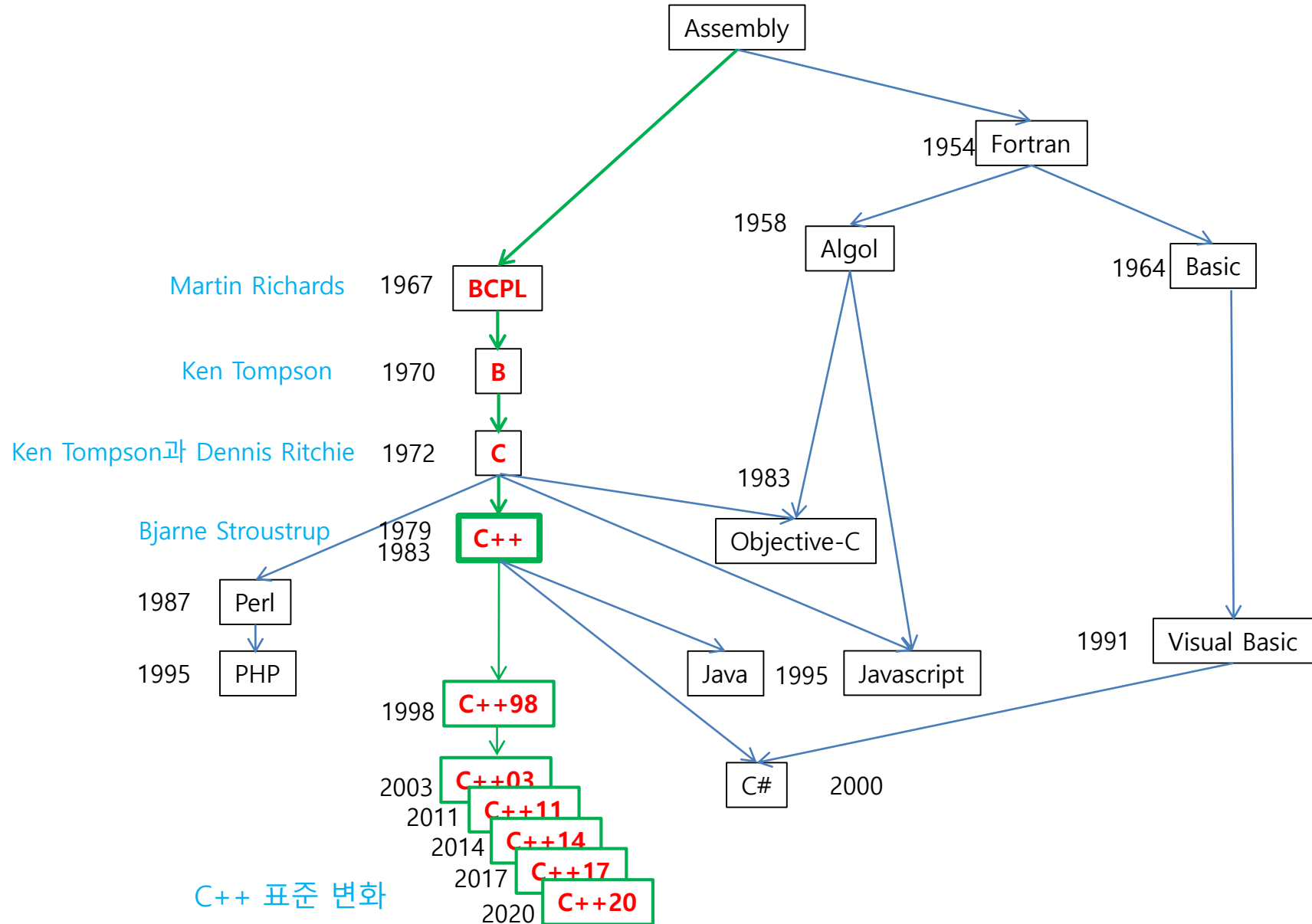
- 사람이 이해하기 쉽고 복잡한 작업, 자료 구조, 알고리즘을 표현하기 위해 고안된 언어
- Pascal, Basic, C/C++, Java, C#
- 컴파일러 : 고급 언어로 작성된 프로그램을 기계어 코드로 변환

# 사람과 컴퓨터, 기계어와 고급 언어

35 + 56 = ?



# 프로그래밍 언어의 진화와 C++의 기원



# C++ 버전

연도	C++ 표준	비공식적인 이름
1998	ISO/IEC 14882:1998	C++98
2003	ISO/IEC 14882:2003	C++03
2011	ISO/IEC 14882:2011	C++11
2014	ISO/IEC 14882:2014	C++14
2017	아직 정해지지 않음	C++17
2020	아직 정해지지 않음	C++20

# 표준 C++ 프로그램의 중요성

## ■ C++ 언어의 표준

- 1998년 미국 표준원(ANSI, American National Standards Institute)
  - C++ 언어에 대한 표준 설정
- ISO/IEC 14882 문서에 작성됨. 유료 문서
- 표준의 진화
  - 1998년(C++98), 2003년(C++03), 2007년(C++TR1), 2011년(C++11)

## ■ 표준의 중요성

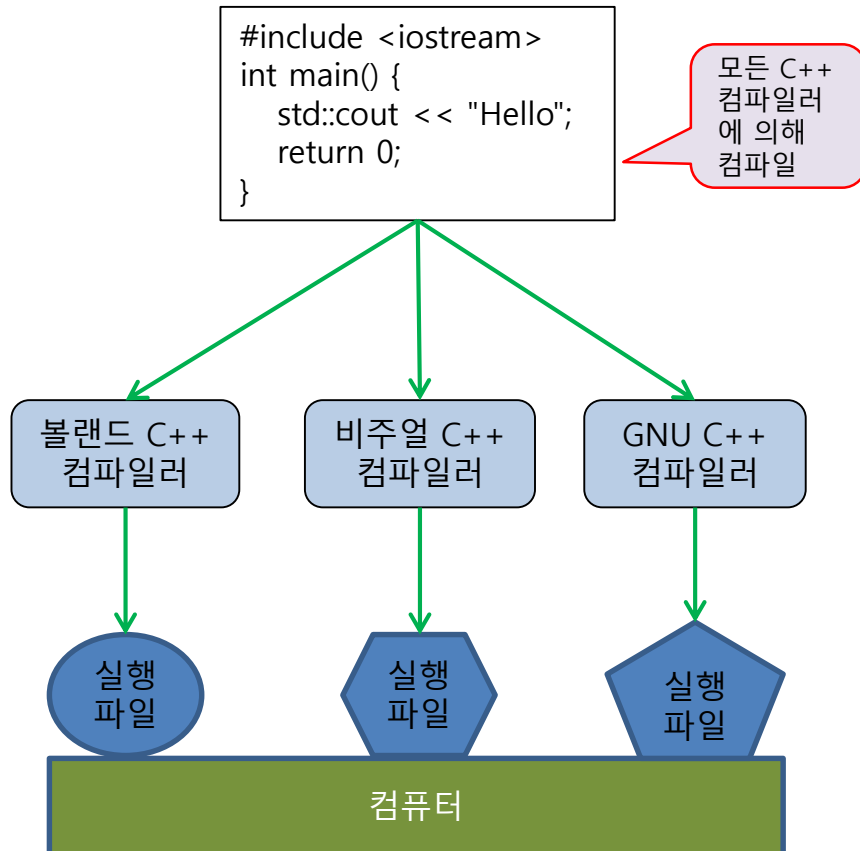
- 표준에 의해 작성된 C++ 프로그램
  - 모든 플랫폼. 모든 표준 C++ 컴파일러에 의해 컴파일
  - 동일한 실행 결과 보장
  - 운영체제와 컴파일러의 종류에 관계없는 높은 호환성

## ■ 비 표준 C++ 프로그램

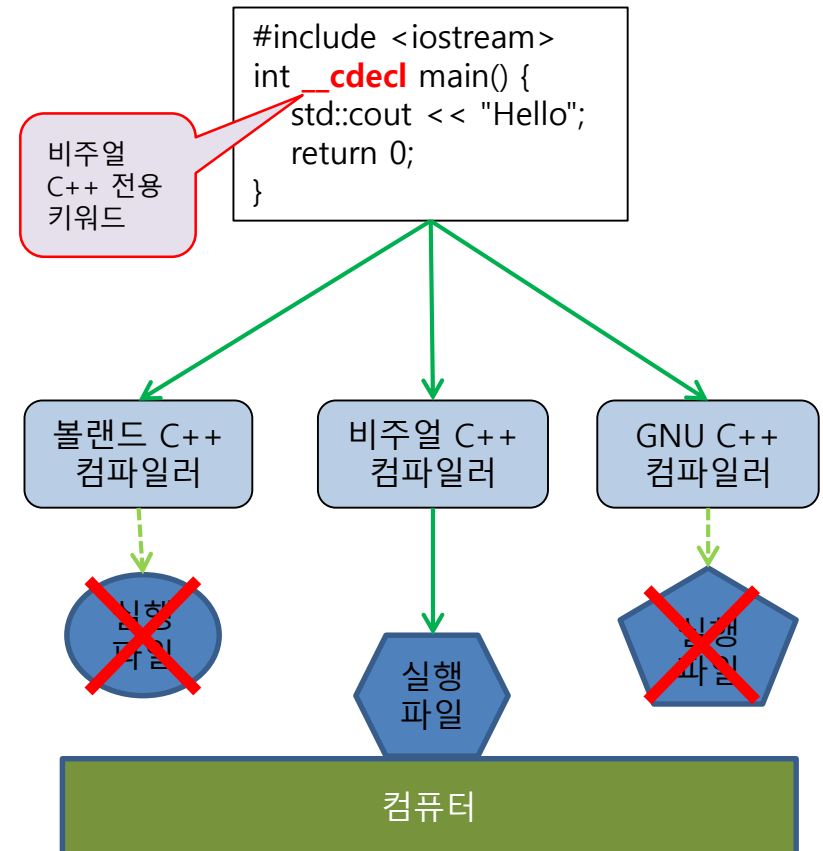
- Visual C++, Borland C++ 등 컴파일러 회사 고유의 비 표준 구문
  - 특정 C++ 컴파일러에서만 컴파일
- 호환성 결여

# 표준/비표준 C++ 프로그램의 비교

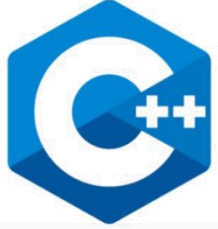
표준 C++ 규칙에 따라  
작성된 C++ 프로그램



표준 C++ 규칙에 따라  
작성되지 않는 비주얼 C++ 프로그램







## ■ C++ 언어

- C언어를 바탕으로 함
- 객체지향 프로그래밍(Object-Oriented Programming: OOP)을 지원하기 위해 만들어짐
- 대다수의 응용 프로그램을 만들 때 가장 많이 사용함
- 강력함과 편리함의 양쪽 장점을 모두 내포한 언어

## ■ C++의 특징

- C의 유연성에 객체지향의 편리성을 접목시킴
- 기존의 C언어로 개발된 모든 프로그램을 수정 없이 사용 가능함
- C언어에 익숙해지면 C++도 빠른 적응 가능함
- 대부분의 운영체제에서 C++를 지원함

# C++ 언어의 주요한 설계 목적

## ■ C 언어와의 호환성

- C 언어의 문법 체계 계승
  - 소스 레벨 호환성 - 기존에 작성된 C 프로그램을 그대로 가져다 사용
  - 링크 레벨 호환성 - C 목적 파일과 라이브러리를 C++ 프로그램에서 링크

## ■ 객체 지향 개념 도입

- 캡슐화, 상속, 다형성
- 소프트웨어의 재사용을 통해 생산성 향상
- 복잡하고 큰 규모의 소프트웨어의 작성, 관리, 유지보수 용이

## ■ 엄격한 타입 체크

- 실행 시간 오류의 가능성을 줄임
- 디버깅 편리

## ■ 실행 시간의 효율성 저하 최소화

- 실행 시간을 저하시키는 요소와 해결
  - 작은 크기의 멤버 함수 잦은 호출 가능성 -> 인라인 함수로 실행 시간 저하 해소

# C 언어에 추가한 기능

- 함수 중복(function overloading)
  - 매개 변수의 개수나 타입이 다른 동일한 이름의 함수들 선언
- 디폴트 매개 변수(default parameter)
  - 매개 변수에 디폴트 값이 전달되도록 함수 선언
- 참조와 참조 변수(reference)
  - 하나의 변수에 별명을 사용하는 참조 변수 도입
- 참조에 의한 호출(call-by-reference)
  - 함수 호출 시 참조 전달
- new/delete 연산자
  - 동적 메모리 할당/해제를 위해 new와 delete 연산자 도입
- 연산자 재정의
  - 기존 C++ 연산자에 새로운 연산 정의
- 제네릭 함수와 클래스
  - 데이터 타입에 의존하지 않고 일반화시킨 함수나 클래스 작성 가능

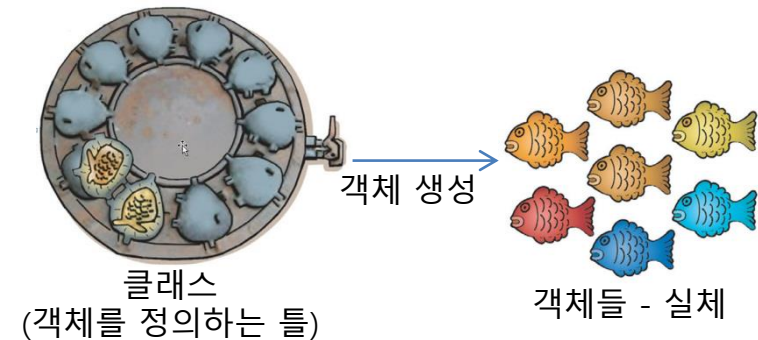
# C++ 객체 지향 특성 - 캡슐화

## ■ 캡슐화(Encapsulation)

- 데이터를 캡슐로 싸서 외부의 접근으로부터 보호
- C++에서 클래스(class 키워드)로 캡슐 표현

## ■ 클래스와 객체

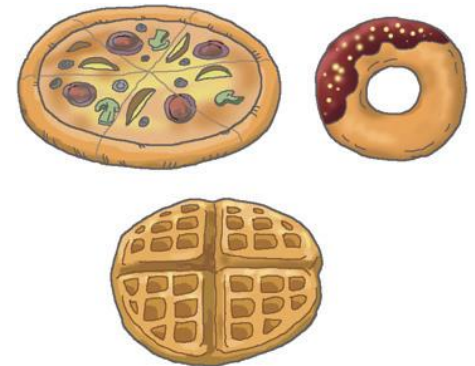
- 클래스 - 객체를 만드는 틀
- 객체 - 클래스라는 틀에서 생겨난 실체
- 객체(object), 실체(instance)는 같은 뜻



```
class Circle {  
    private:  
        int radius; // 반지름 값  
    public:  
        Circle(int r) { radius = r; }  
        double getArea() { return 3.14*radius*radius; }  
};
```

멤버들

원을 추상화한 Circle 클래스



원 객체들(실체)

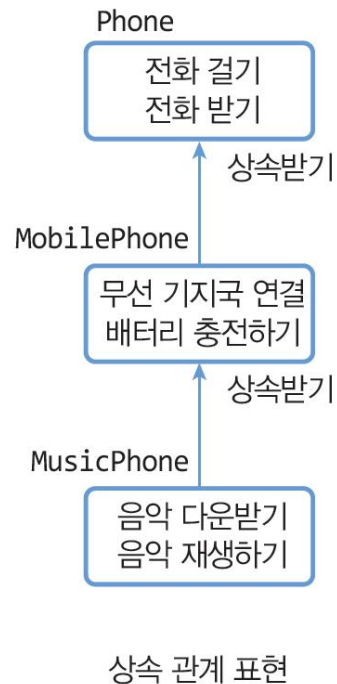
# C++ 객체 지향 특성 - 상속성

## ■ 객체 지향 상속(Inheritance)

- 자식이 부모의 유전자를 물려 받는 것과 유사

## ■ C++ 상속

- 객체가 자식 클래스의 멤버와 부모 클래스에 선언된 모양 그대로 멤버들을 가지고 탄생



```
class Phone {  
    void call();  
    void receive();  
};  
  
class MobilePhone : public Phone {  
    void connectWireless();  
    void recharge();  
};  
  
class MusicPhone : public MobilePhone {  
    void downloadMusic();  
    void play();  
};
```

C++로 상속 선언



전화기



휴대 전화기



음악 기능  
전화기

# C++ 객체 지향 특성 - 다형성

## 다형성(Polymorphism)

- 하나의 기능이 경우에 따라 다르게 보이거나 다르게 작동하는 현상
- 연산자 중복, 함수 중복, 함수 재정의(overriding)

```
2 + 3 --> 5  
"남자" + "여자" --> "남자여자"  
redColor 객체 + blueColor 객체 --> purpleColor 객체
```

+ 연산자 중복

```
void add(int a, int b) { ... }  
void add(int a, int b, int c) { ... }  
void add(int a, double d) { ... }
```

add 함수 중복



함수 재정의(오버라이딩)

# C++ 언어에서 객체 지향을 도입한 목적

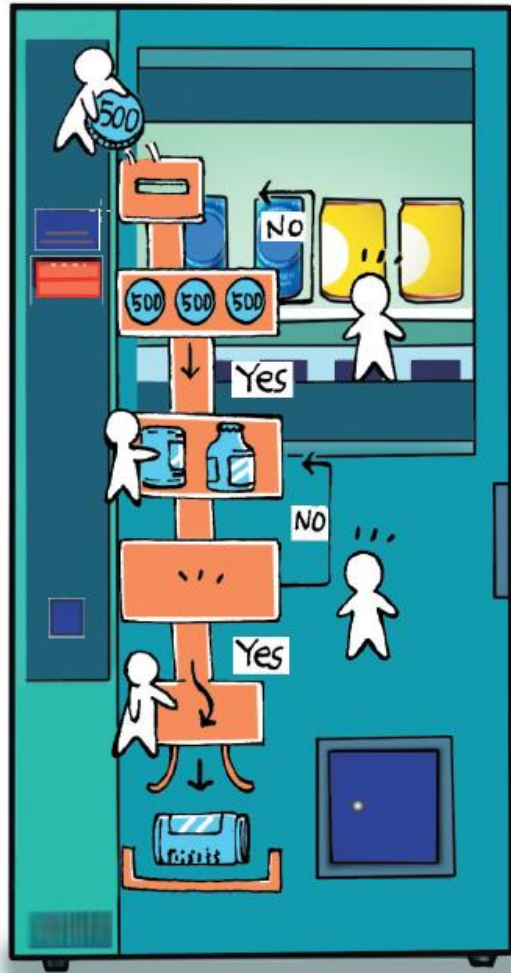
## ■ 소프트웨어 생산성 향상

- 소프트웨어의 생명 주기 단축 문제 해결 필요
- 기 작성된 코드의 재사용 필요
- C++ 클래스 상속 및 객체 재사용으로 해결

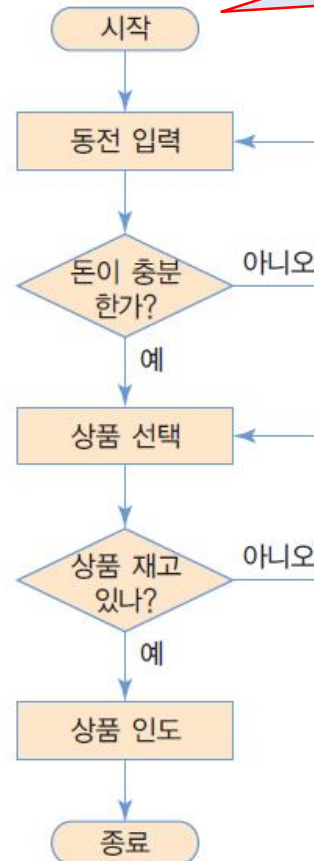
## ■ 실세계에 대한 쉬운 모델링

- 과거의 소프트웨어
  - 수학 계산이나 통계 처리에 편리한 절차 지향 언어가 적합
- 현대의 소프트웨어
  - 물체 혹은 객체의 상호 작용에 대한 묘사가 필요
  - 실세계는 객체로 구성된 세계
  - 객체를 중심으로 하는 객체 지향 언어 적합

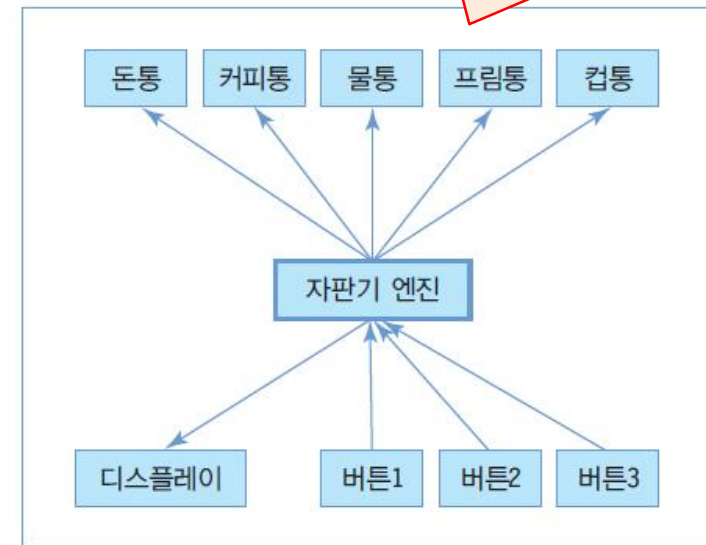
# 절차 지향 프로그래밍과 객체 지향 프로그래밍



- 실행하고자 하는 절차대로 일련의 명령어 나열.
- 흐름도를 설계하고 흐름도에 따라 프로그램 작성



- 객체들을 정의하고, 객체들의 상호 관계, 상호 작용으로 구현



(a) 절차 지향적 프로그래밍으로 구현할 때의 흐름도

(b) 객체 지향적 프로그래밍으로 구현할 때의 객체 관계도



# C++와 제네릭 프로그래밍

## ■ 제네릭 함수와 제네릭 클래스

- 제네릭 함수(generic function)
  - 동일한 프로그램 코드에 다양한 데이터 타입을 적용할 수 있게 일반화 시킨 함수
- 제네릭 클래스(generic class)
  - 동일한 프로그램 코드에 다양한 데이터 타입을 적용할 수 있게 일반화 시킨 클래스
- template 키워드로 선언
  - 템플릿 함수 혹은 템플릿 클래스라고도 부름
- Java, C# 등 다른 언어에도 동일한 개념 있음

## ■ 제네릭 프로그래밍(generic programming)

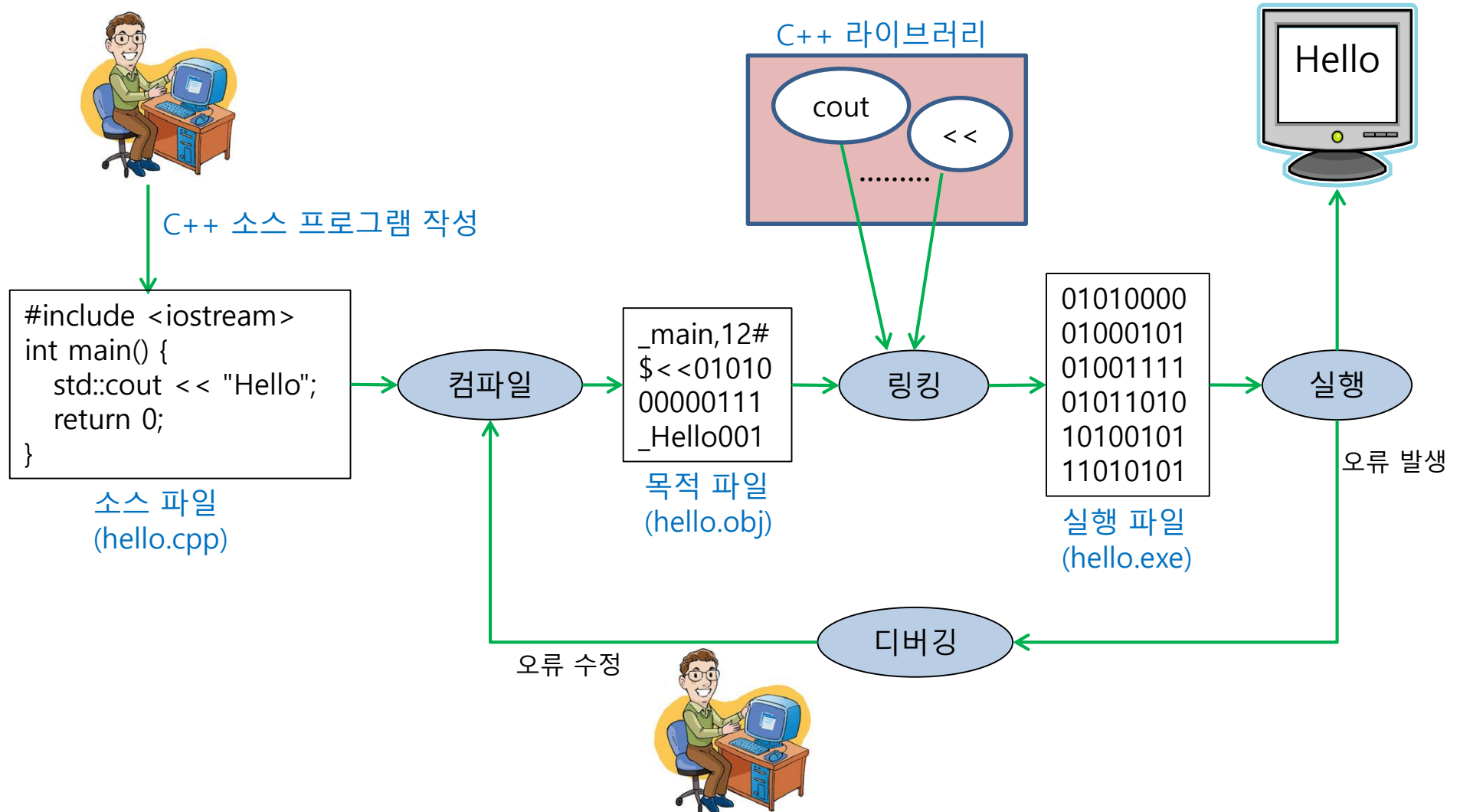
- 제네릭 함수와 제네릭 클래스를 활용하여 프로그램을 작성하는 새로운 프로그래밍 패러다임
- 점점 중요성이 높아지고 있음

# C++ 언어의 아킬레스

## ■ C++ 언어는 C 언어와의 호환성 추구

- 장점
  - 기존에 개발된 C 프로그램 코드 활용
- 단점
  - 캡슐화의 원칙이 무너짐
    - ✓ C++에서 전역 변수와 전역 함수를 사용할 수 밖에 없음
    - ✓ 부작용(side effect) 발생 염려

# C++ 프로그램 개발 과정



# C++ 프로그램 작성 및 컴파일

## ■ 편집

- C++ 소스 프로그램은 텍스트 파일
  - 아무 텍스트 편집기로 편집 가능
- C++ 소스 프로그램의 표준 확장자는 .cpp
- C++ 통합 개발 소프트웨어 이용 추천
  - C++ 소스 편집, 컴파일, 링킹, 실행, 디버깅 등 모든 단계 통합 지원
  - 대표적인 소프트웨어 - Visual Studio

## ■ 컴파일

- C++ 소스 프로그램을 기계어를 가진 목적 파일로 변환
  - cpp 파일을 obj 파일로 변환

_main	PROC				; COMDAT
<b>; 3 : int main() {</b>					
00000	55		push	ebp	
00001	8b ec		mov	ebp, esp	
00003	81 ec c0 00 00				
	00		sub	esp, 192	; 000000c0H
00009	53		push	ebx	
0000a	56		push	esi	
0000b	57		push	edi	
0000c	8d bd 40 ff ff		lea	edi, DWORD PTR [ebp-192]	
00012	b9 30 00 00 00		mov	ecx, 48	; 00000030H
00017	b8 cc cc cc cc		mov	eax, -858993460	; ccccccccH
0001c	f3 ab		rep stosd		
<b>; 4 : std::cout &lt;&lt; "Hello";</b>					
0001e	68 00 00 00 00		push	OFFSET ??_C@_05COLMCDPH@Hello?\$AA@	
00023	a1 00 00 00 00		mov	eax, DWORD PTR __imp_?cout@std@@@3V?\$basic_ostream@DU?\$	
00028	50		push	eax	
00029	e8 00 00 00 00		call	??\$?6U?\$char_traits@D@std@@@YAAAV?\$basic_ostream@DU?\$	
0002e	83 c4 08		add	esp, 8	
<b>; 5 : return 0;</b>					
00031	33 c0		xor	eax, eax	
<b>; 6 :}</b>					
00033	5f		pop	edi	
00034	5e		pop	esi	
00035	5b		pop	ebx	
00036	81 c4 c0 00 00				
	00		add	esp, 192	; 000000c0H
0003c	3b ec		cmp	ebp, esp	
0003e	e8 00 00 00 00		call	__RTC_CheckEsp	
00043	8b e5		mov	esp, ebp	
00045	5d		pop	ebp	
00046	c3		ret	0	
<b>_main</b>	<b>ENDP</b>				

int main() { 라인을 컴파일한 기계어 코드

어셈블리어 코드

```

1 #include <iostream>
2
3 int main() {
4     std::cout << "Hello";
5     return 0;
6 }

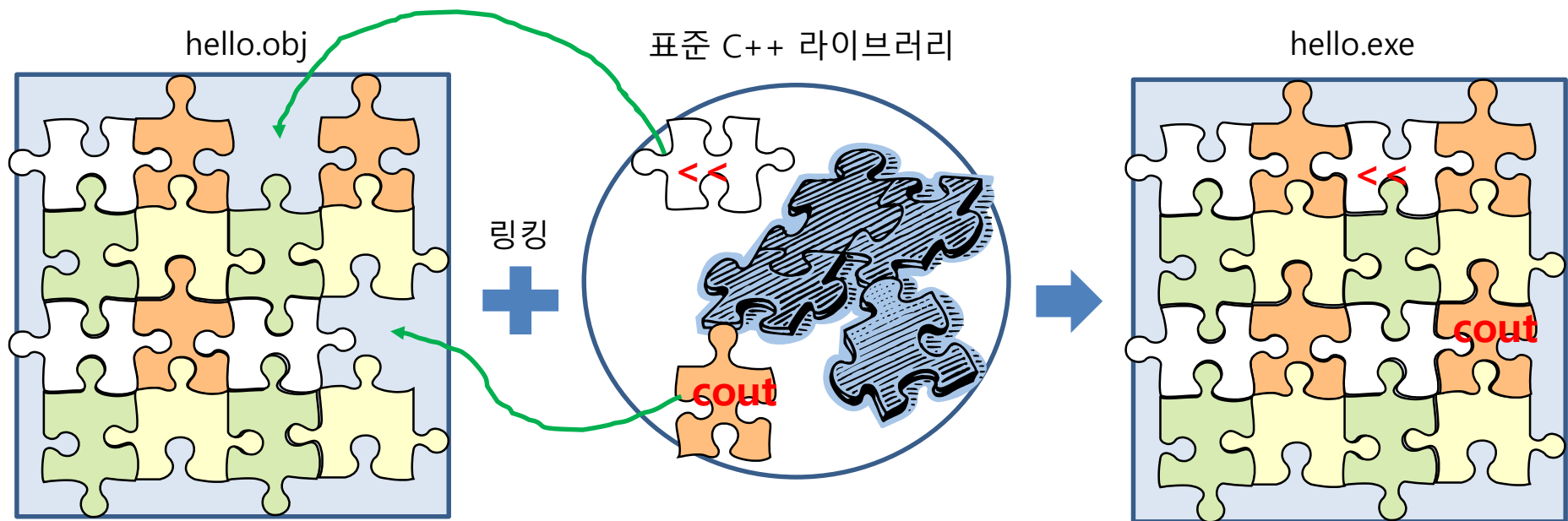
```

# 링킹

## 링킹

- 목적 파일끼리 합쳐 실행 파일을 만드는 과정
  - 목적 파일은 바로 실행할 수 없음
- 목적 파일과 C++ 표준 라이브러리의 함수 연결, 실행 파일을 만드는 과정

hello.obj + cout 객체 + << 연산자 함수 → hello.exe를 만듦



# 프로그램 실행과 디버깅

## ■ 실행 파일은 독립적으로 바로 실행 가능

## ■ 실행 중에 발생하는 오류

- 원하는 결과가 나오지 않거나 실행 중에 프로그램의 비정상 종료

## ■ 디버깅

- 실행 중에 발생한 오류를 찾는 과정
- 디버거
  - 디버깅을 도와주는 프로그램
  - 컴파일러를 만드는 회사에서 함께 공급
- 소스 레벨 디버깅
  - C++ 소스를 한 라인씩 실행하고 변수 값의 변화를 보면서 오류 발견
  - Visual Studio는 소스 레벨 디버깅 지원

# C++ 표준 라이브러리

## ■ C++ 표준 라이브러리는 3 개의 그룹으로 구분

- C 라이브러리
  - 기존 C 표준 라이브러리를 수용, C++에서 사용할 수 있게 한 함수들
  - 이름이 c로 시작하는 헤더 파일에 선언됨
- C++ 입출력 라이브러리
  - 콘솔 및 파일 입출력을 위한 라이브러리
- C++ STL 라이브러리
  - 제네릭 프로그래밍을 지원하기 위해 템플릿 라이브러리

algorithm	complex	exception	list	stack
bitset	csetjmp	fstream	locale	stdexcept
cassert	csignal	functional	map	strstream
cctype	cstdarg	omanip	memory	streambuf
cerrno	cstddef	ios	new	string
cfloat	cstdio	iosfwd	numeric	typeinfo
ciso646	cstdlib	iostream	ostream	utility
climits	cstring	istream	queue	valarray
clocale	ctime	iterator	set	vector
cmath	deque	limits	sstream	

C 라이브러리

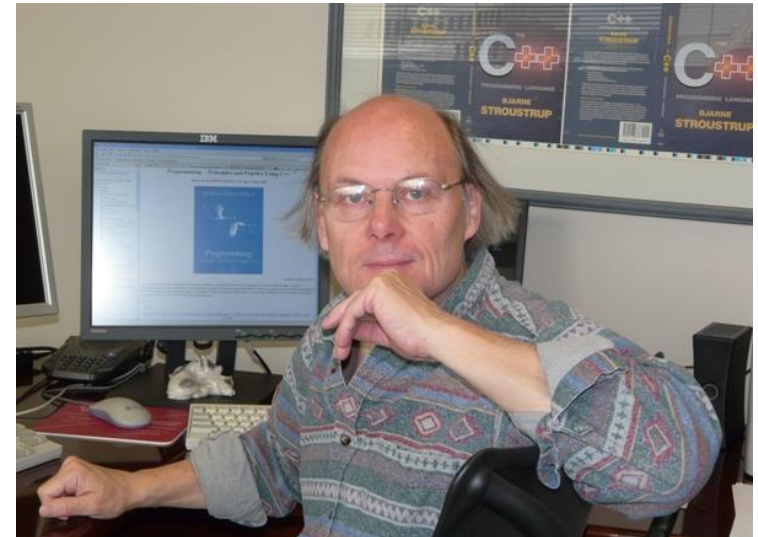
STL 라이브러리

C++ 입출력 라이브러리



# C++ 언어의 개발























- C++는 1980년대 초에 AT&T 벨 연구소의 비야네 스트롭스트롭(Bjarne Stroustrup)에 의하여 개발
- C++는 C언어를 유지, 확장
- C with Classes → C++
- C++는 C언어에 클래스 개념을 추가하고 이어서 가상 함수, 연산자 중복 정의, 다중 상속, 템플릿, 예외 처리 등의 기능이 차례로 추가



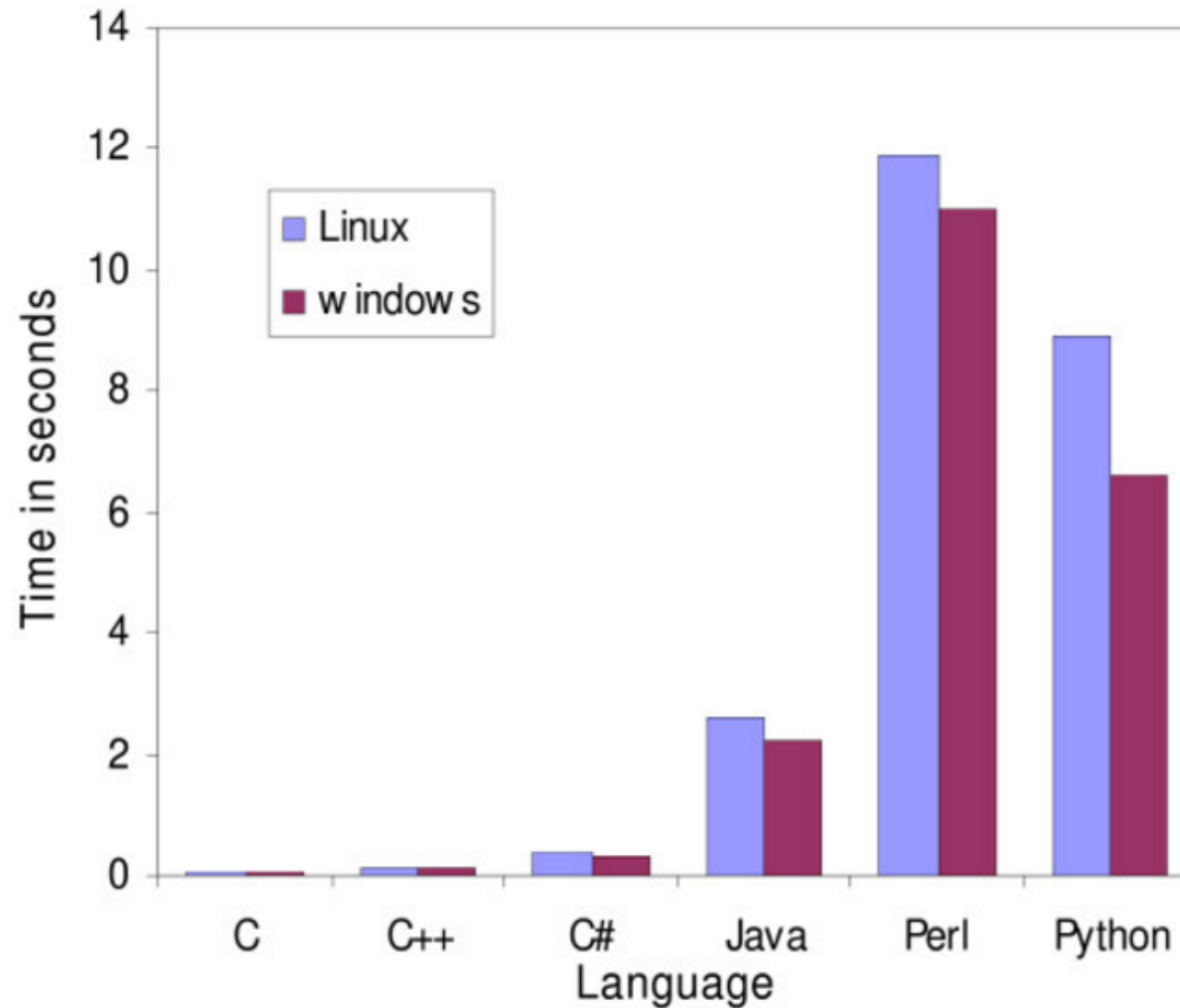
# C++의 설계 철학

- 엄격한 타입 검사, 범용 언어, 효율적, 이식성
- 여러 가지의 프로그래밍 스타일을 지원
  - 절차 지향 프로그래밍
  - 데이터 추상화
  - 객체 지향 프로그래밍
  - 일반화 프로그래밍
- 프로그래머가 자유롭게 선택할 수 있도록 설계
- 최대한 C와 호환
- 플랫폼에 의존적이거나 일반적이지 않은 특징은 제거

# C++는 어디에 사용되는가?

Language Rank	Types	Spectrum Ranking
1. C	  	100.0
2. Java	  	98.1
3. Python	 	98.0
4. C++	  	95.9
5. R		87.9
6. C#	  	86.7
7. PHP		82.8
8. JavaScript	 	82.2
9. Ruby	 	74.5
10. Go	 	71.9

# C++의 실행 속도



# C++ 프로그램 개발 단계

- ① 텍스트 에디터로 C++ 언어 프로그램을 작성하여 파일로 저장한다.
- ② 소스 파일을 컴파일한다.
- ③ 프로그램을 실행한다.

# 비주얼 스튜디오란?

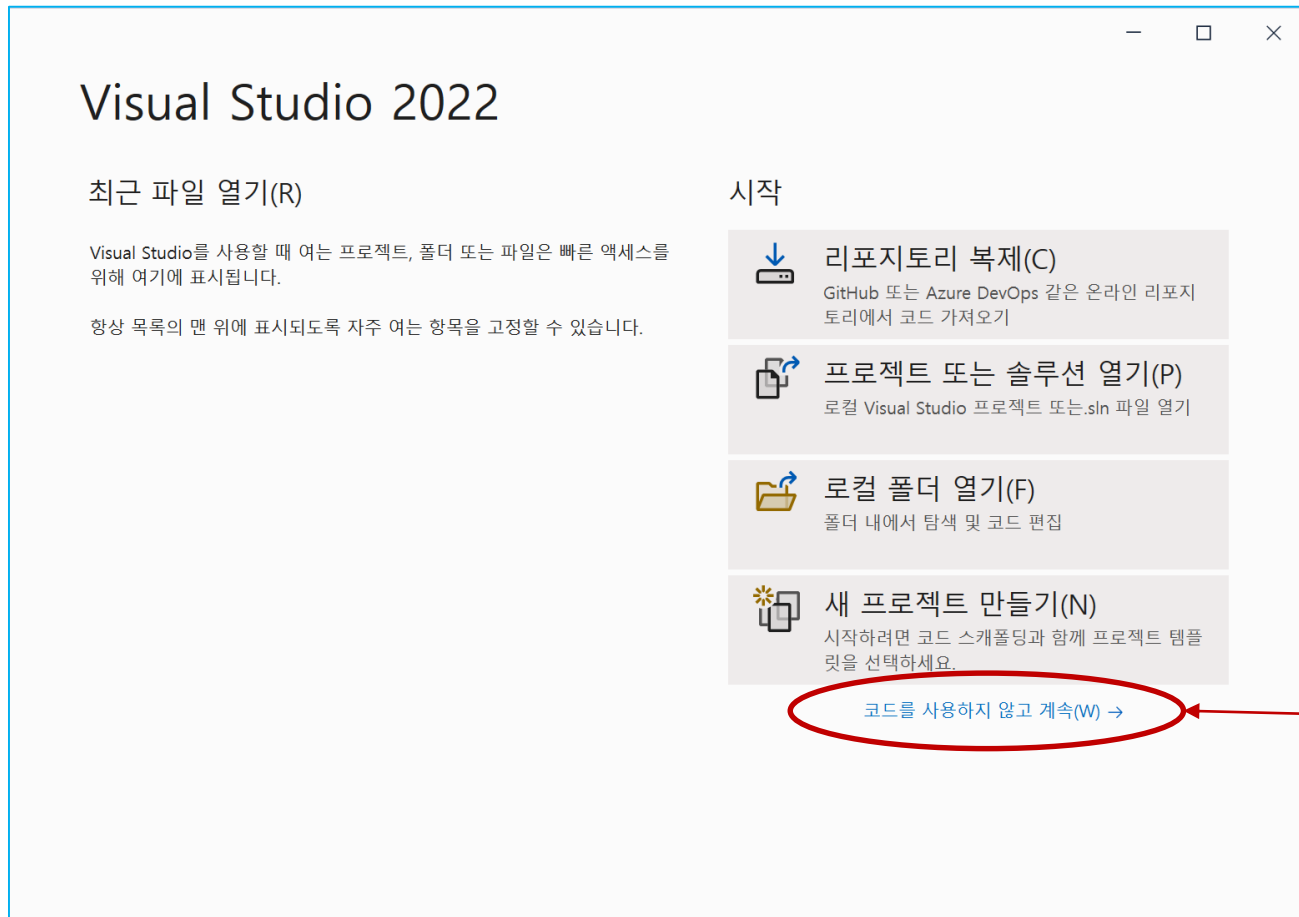
- 통합 개발 환경(IDE: integrated development environment)은 프로그램 개발에 필수적인 편집, 컴파일, 실행, 디버깅 기능을 하나로 통합한 도구



# 비주얼 스튜디오 버전

- 커뮤니티 버전(Visual Studio Community)
- 프로페셔널 버전(Visual Studio Professional)
- 엔터프라이즈 버전(Visual Studio Enterprise)

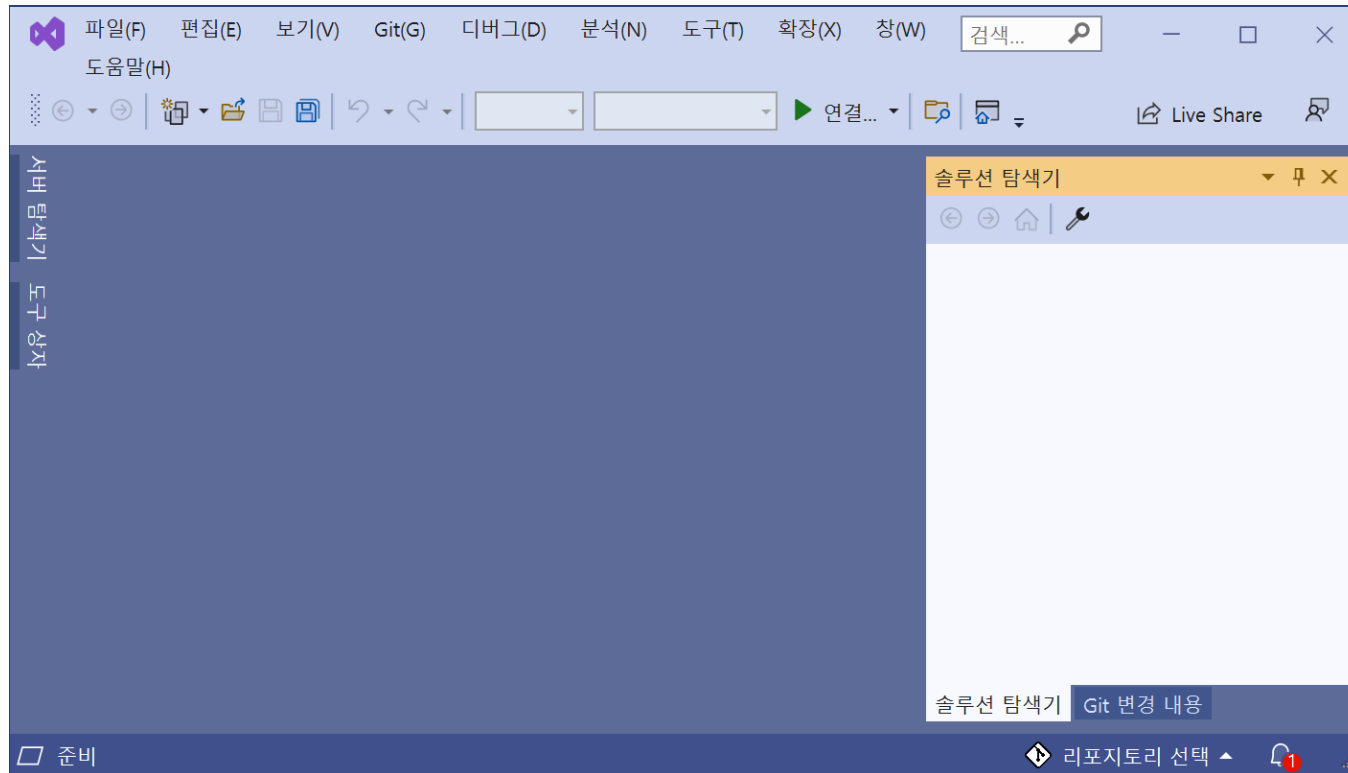
# Visual Studio 시작



클릭하면  
다음 슬라이드로



# Visual Studio 스크린



# 프로젝트 만들기



# 새 프로젝트 구성

□ ×

## 새 프로젝트 구성

빈 프로젝트 C++ Windows 콘솔

프로젝트 이름(J)

C:\WC++\Wchap1\Hello  
폴더가 생긴다.

Hello

위치(L)

C:\WC++

솔루션 위치

...

솔루션 이름(M) ⓘ

chap1

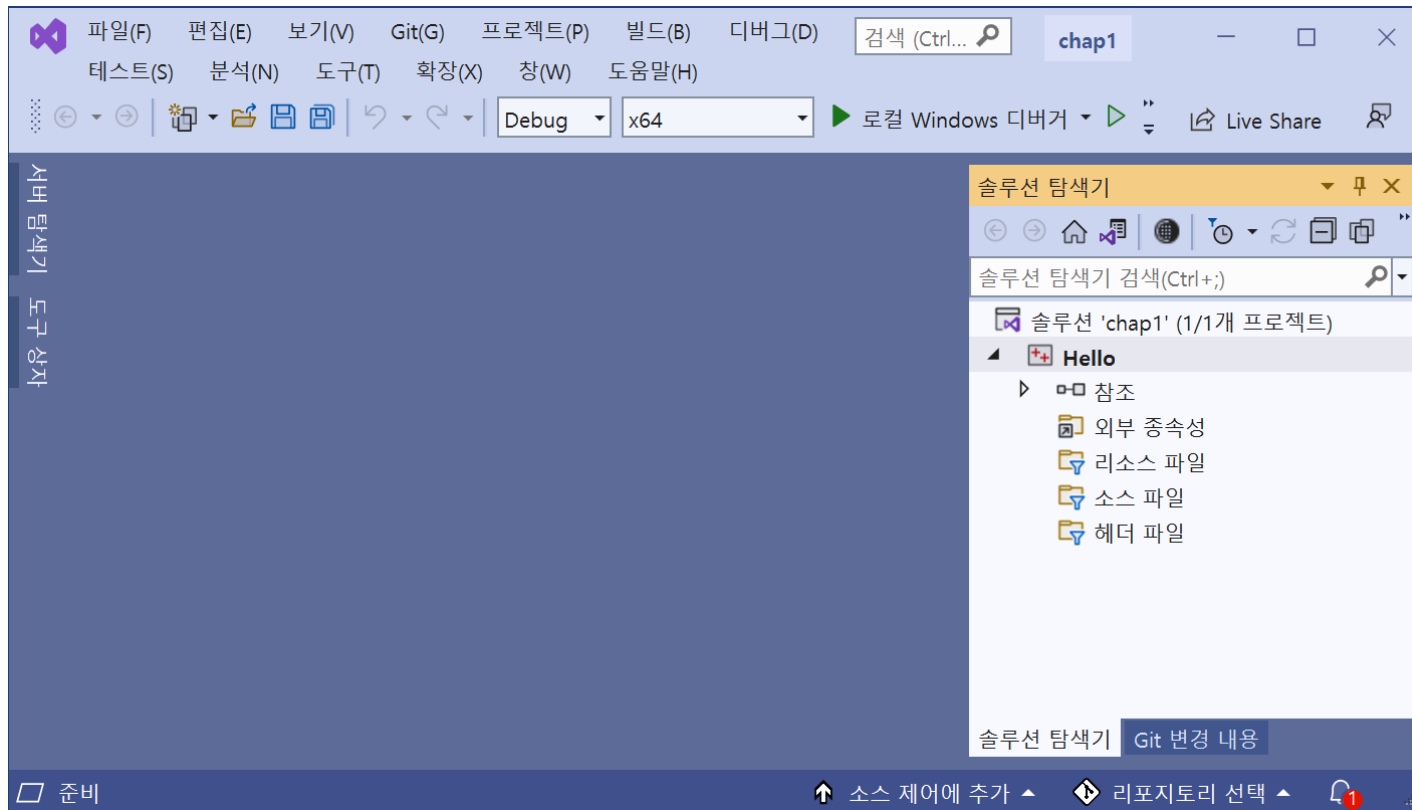
☐ 솔루션 및 프로젝트를 생성한다.

C:\WC++\Wchap1  
폴더를 생성한다.

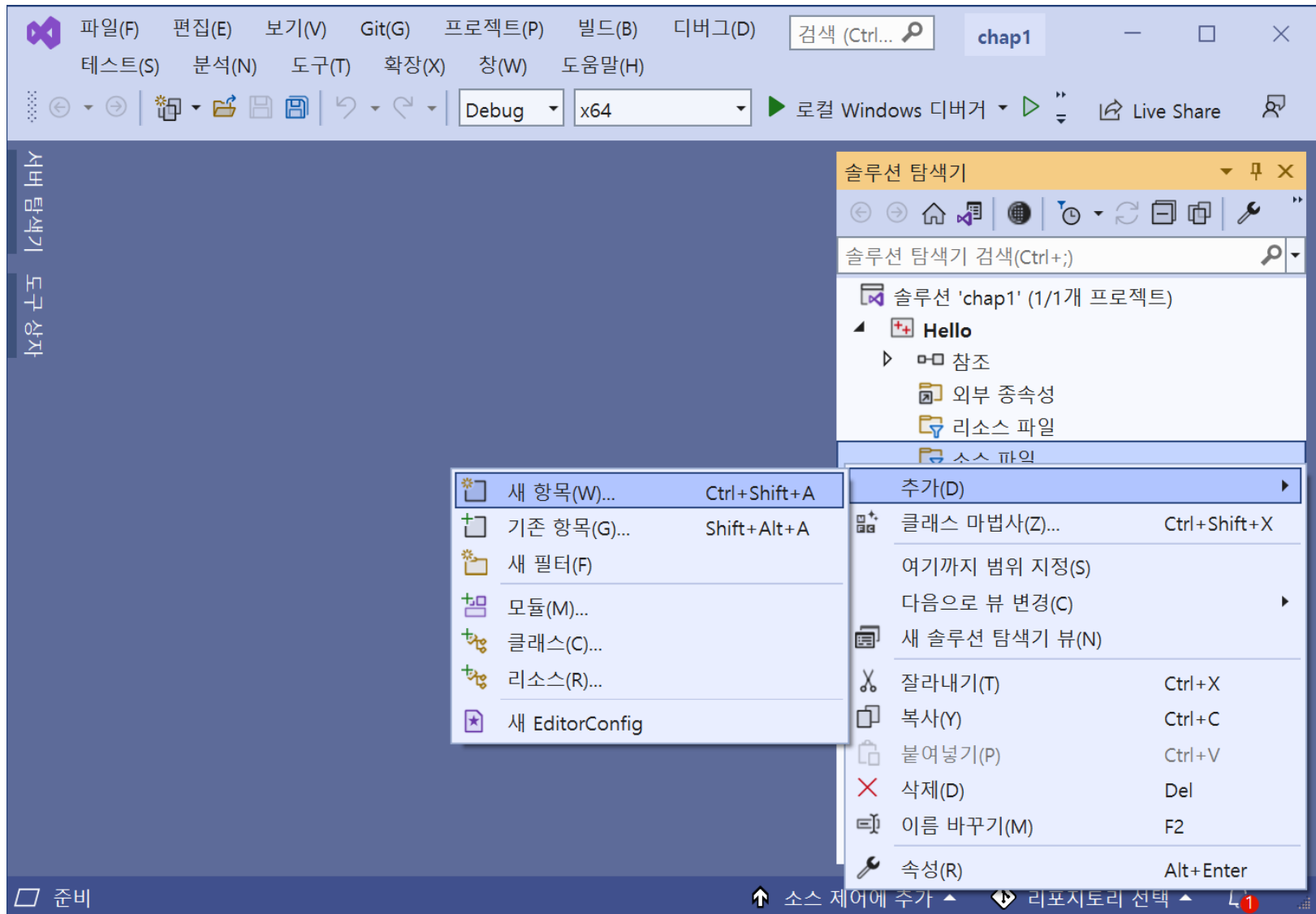
뒤로(B)

만들기(C)

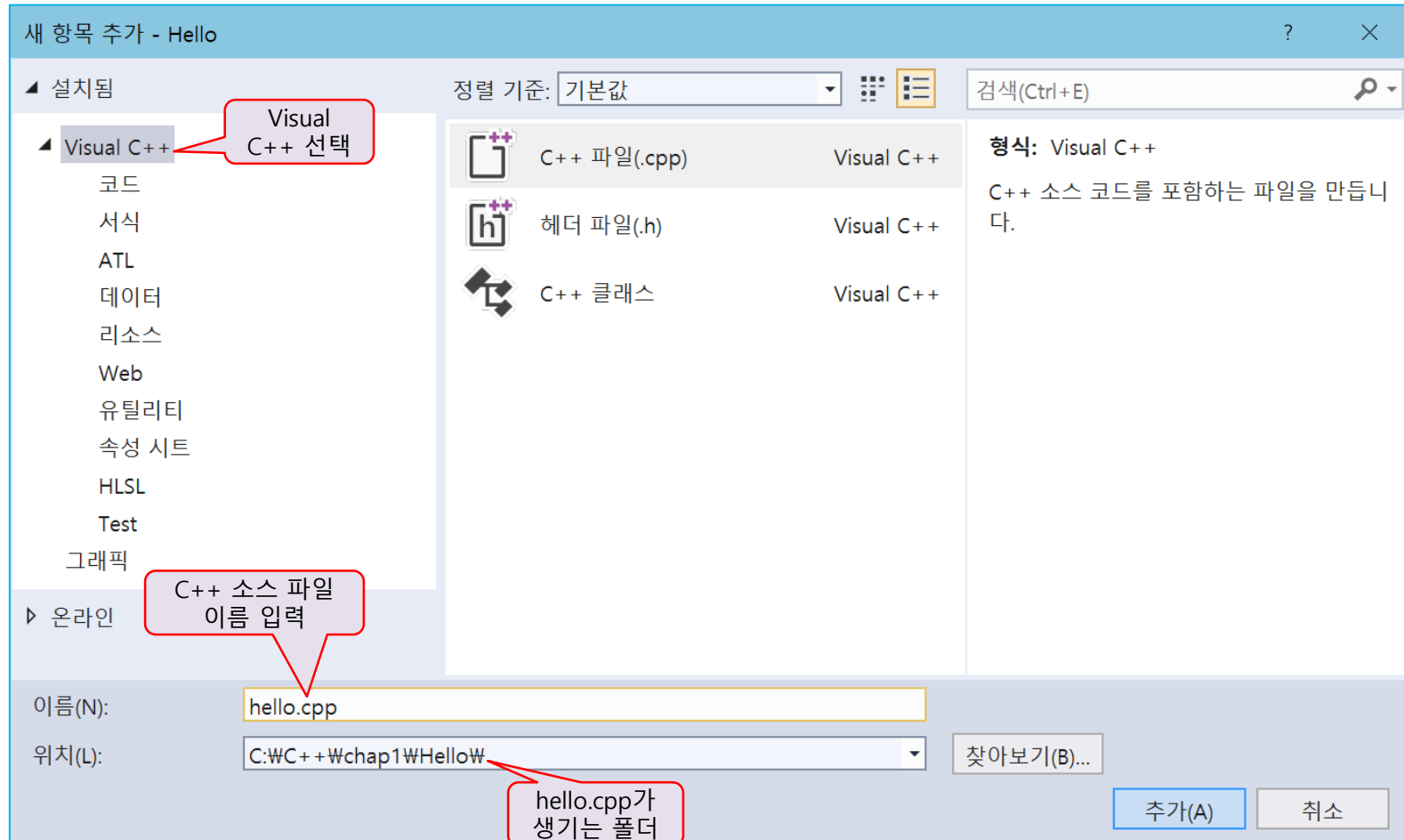
# Hello 프로젝트 생성 후



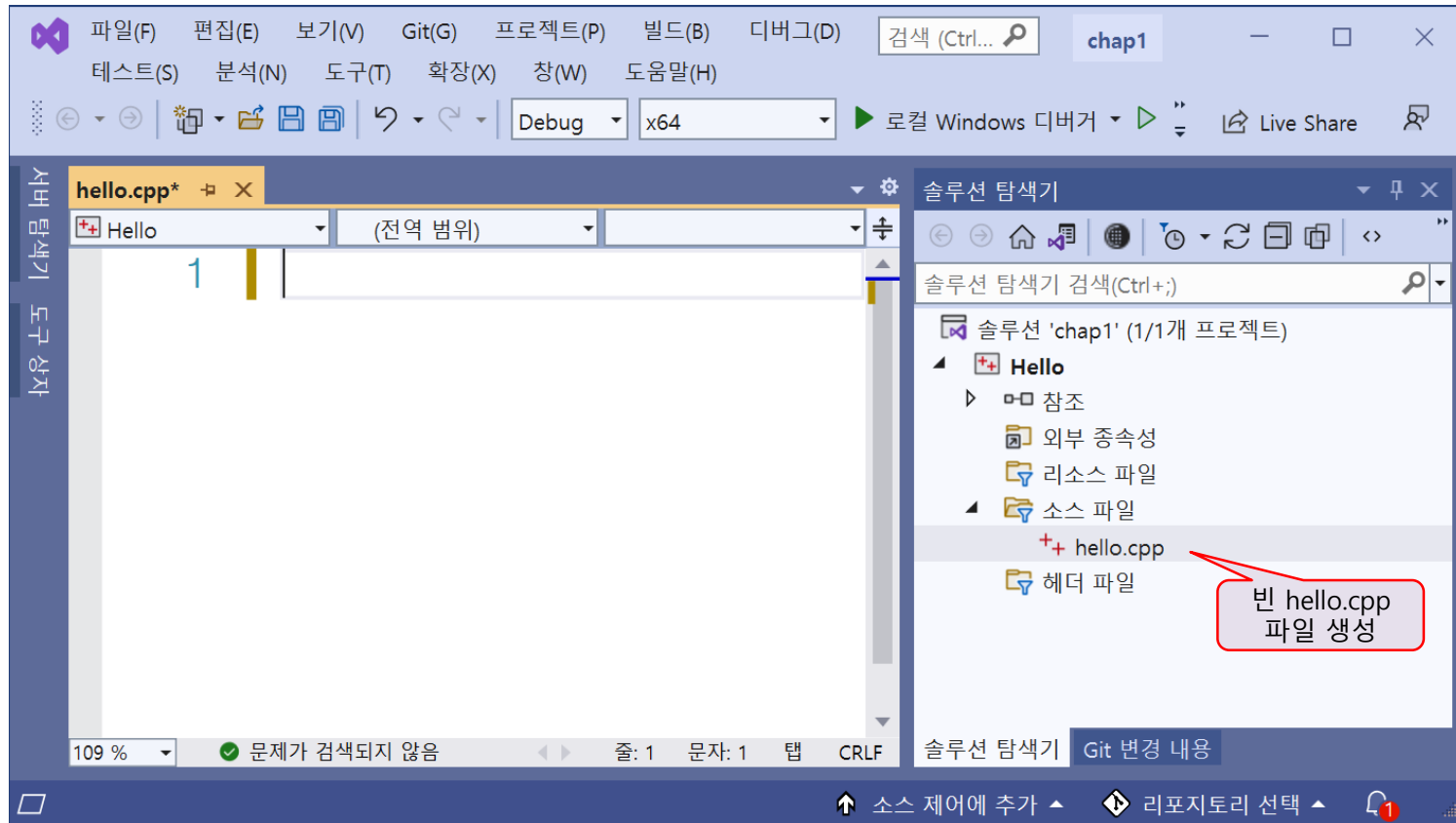
# 새 항목 만드는 메뉴 선택



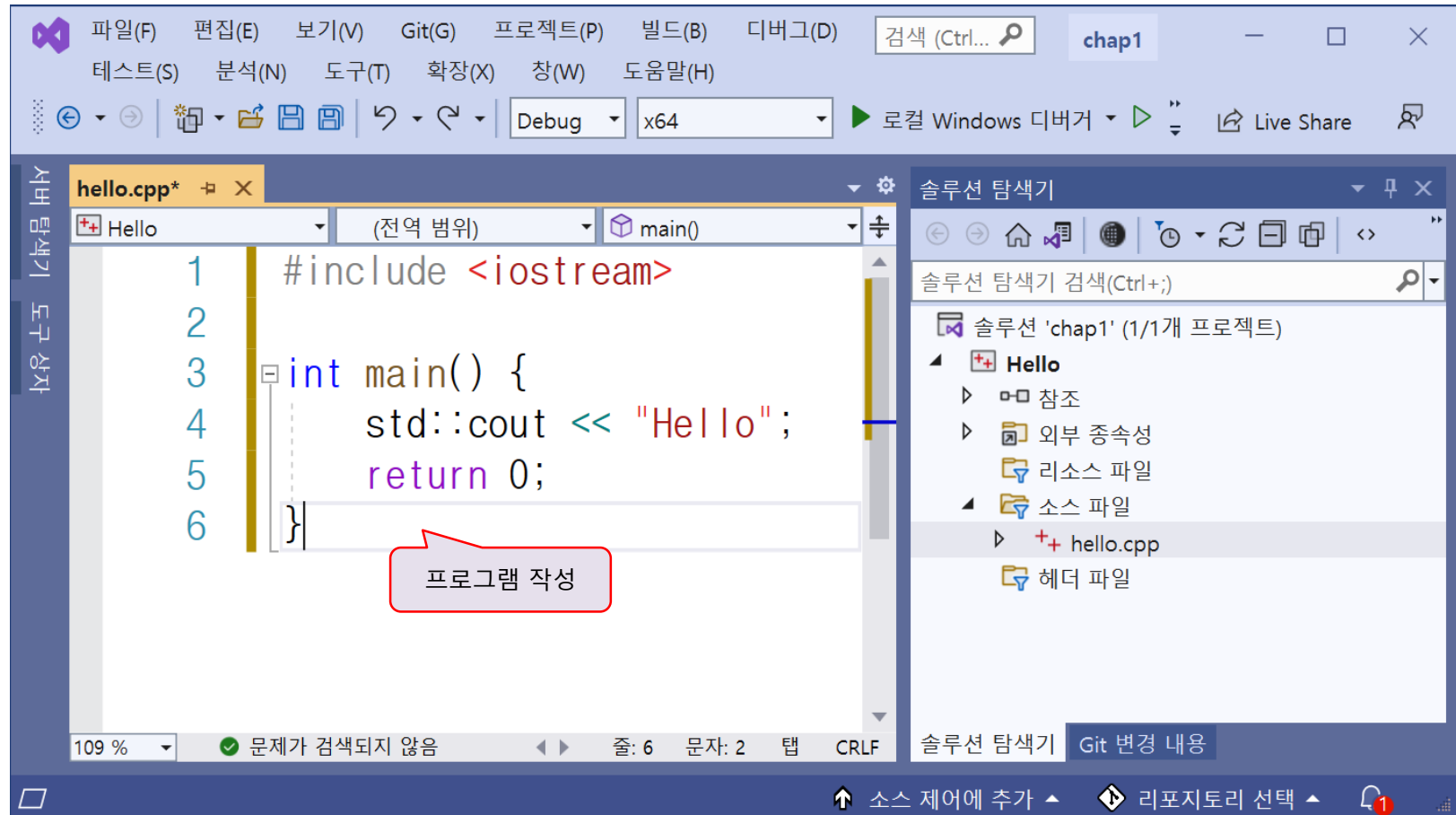
# hello.cpp 소스 파일 생성



# hello.cpp 파일이 생성된 초기 모습

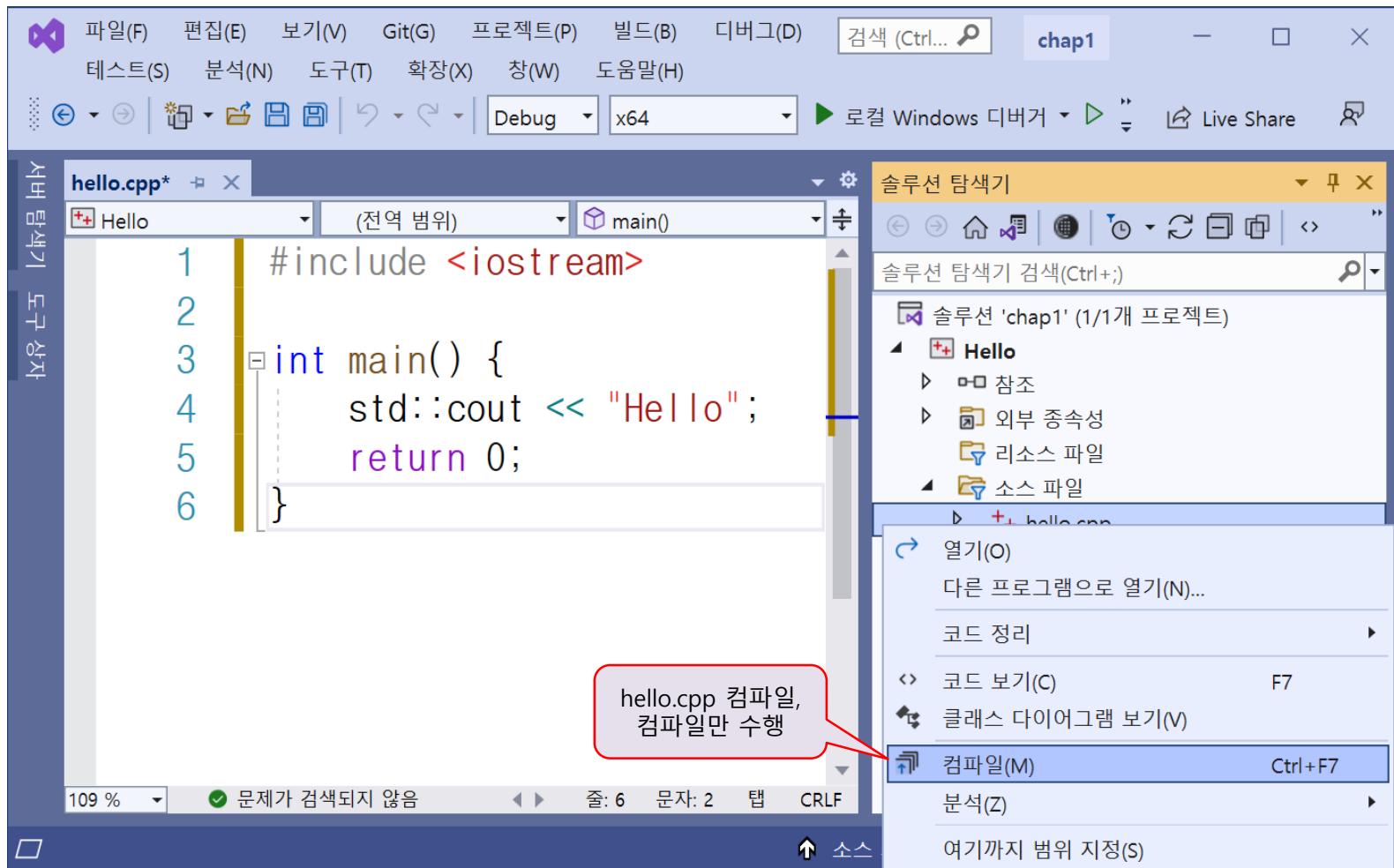


# hello.cpp 작성

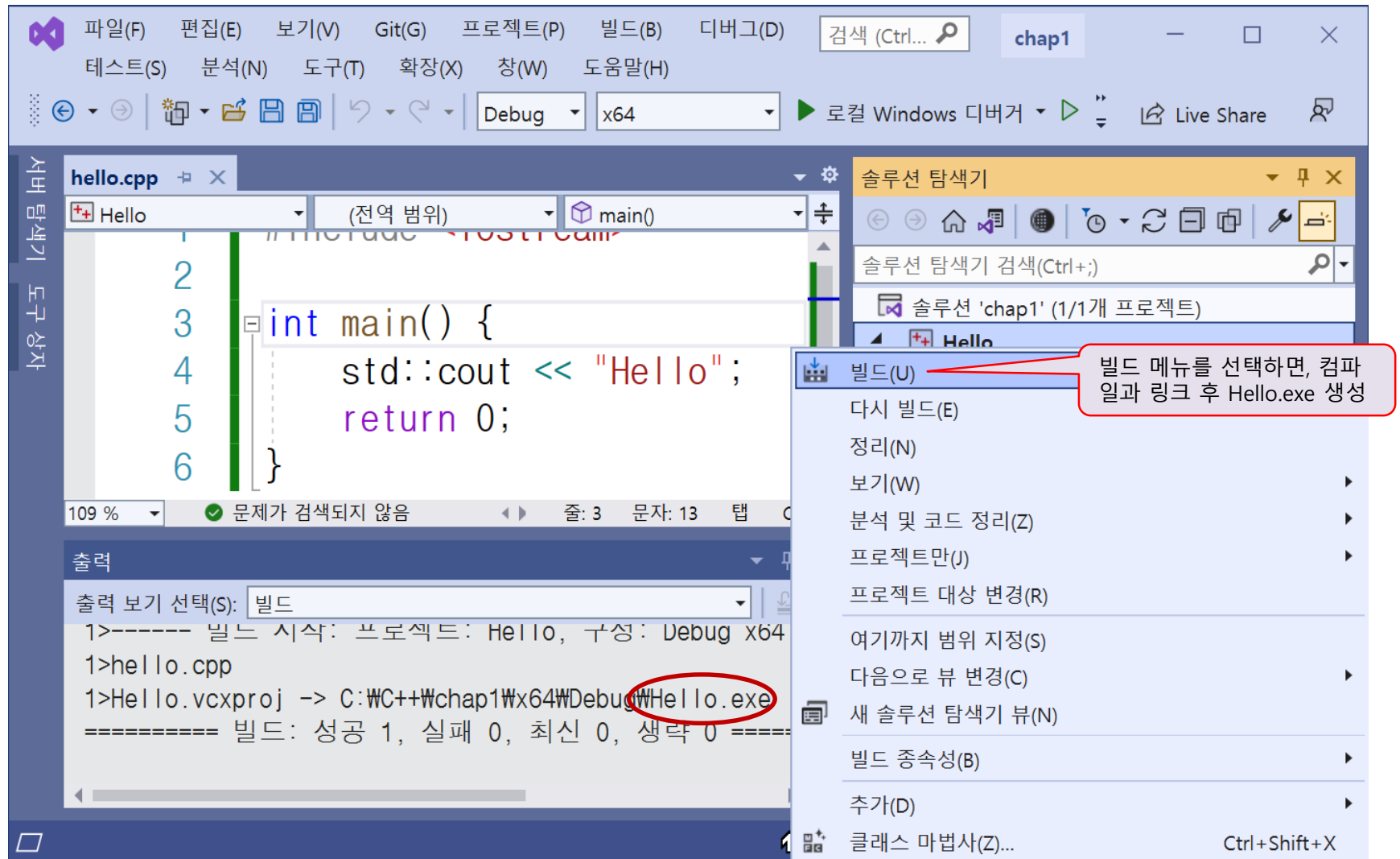




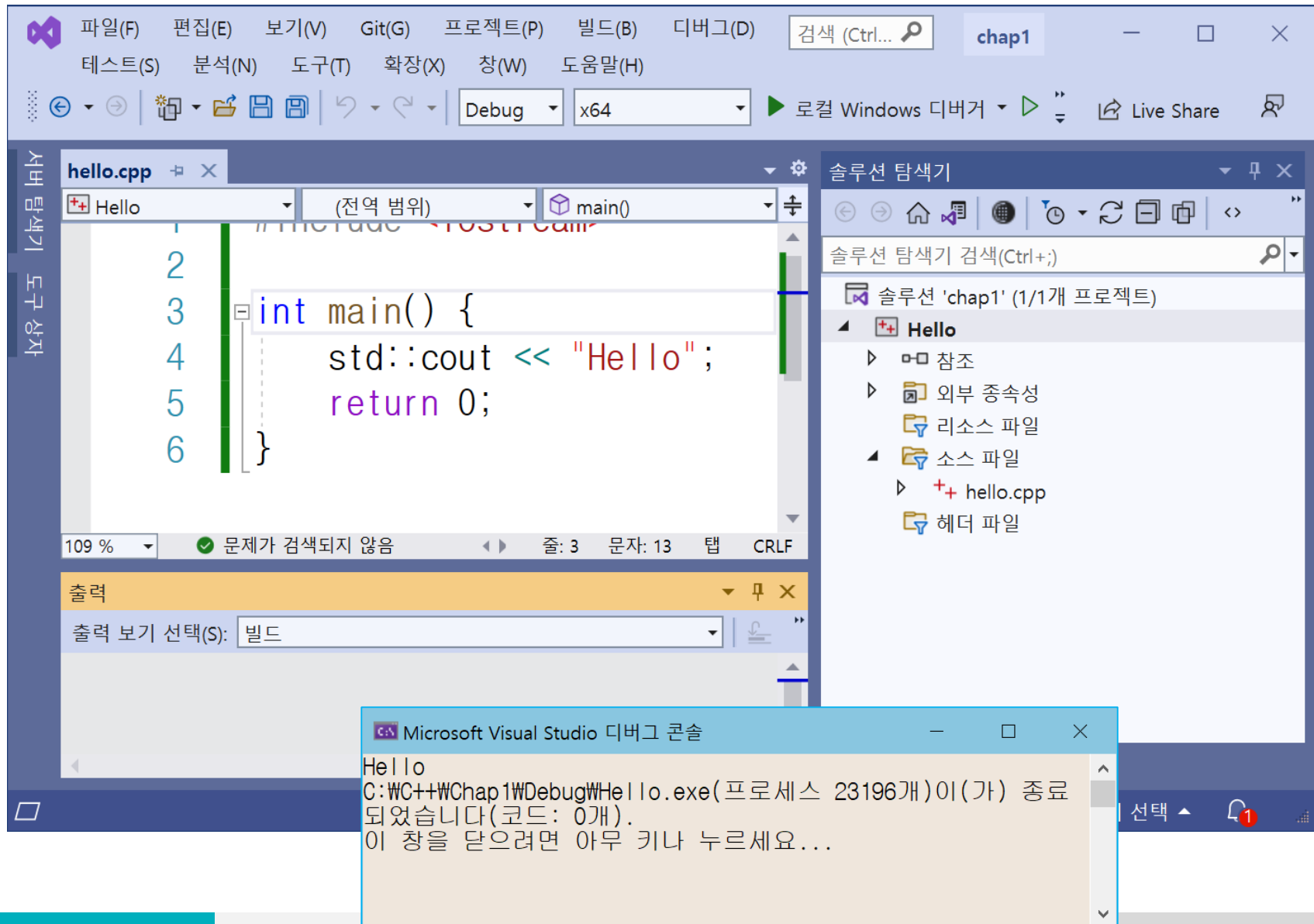
# 솔루션 탐색기에서 컴파일 메뉴 선택



# Hello 프로젝트의 빌드로 Hello.exe 생성



# Hello 프로젝트가 실행되는 화면



# 첫 번째 프로그램의 분석

hello.cpp

01 // 첫 번째 예제 프로그램

주석

02 #include <iostream>

헤더파일

03 using namespace std;

이름 공간 설정

04

05 int main()

함수 선언

06 {

07 cout << "Hello World!"<< endl;

화면에 문자열 출력

08 return 0;

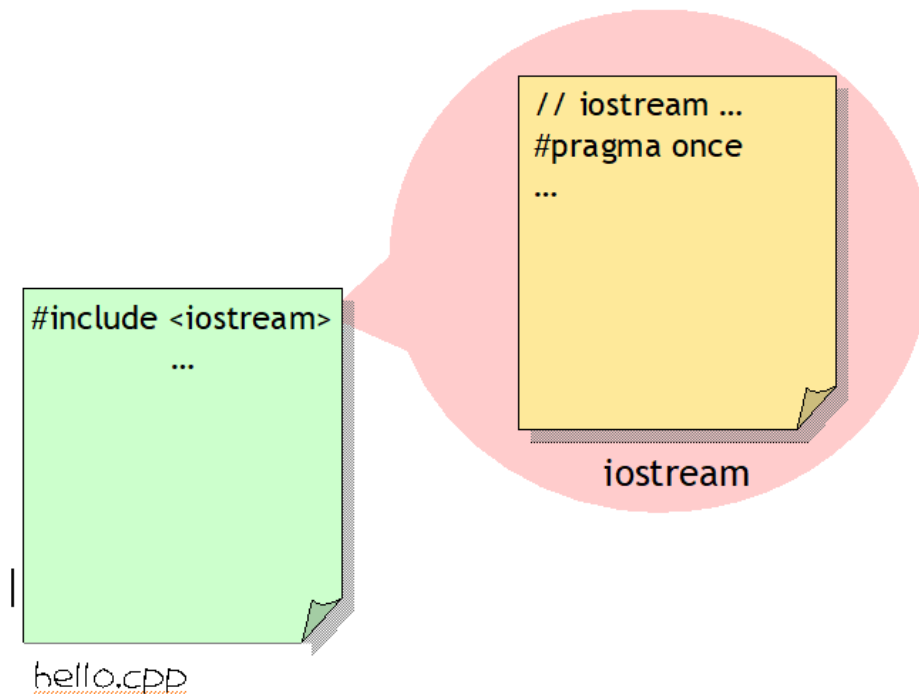
09 }

# // 첫 번째 예제 프로그램

- 주석(comment)이란 코드를 설명하는 글
- 2가지 종류의 주석
  - `/* ... */`
  - `// ...`

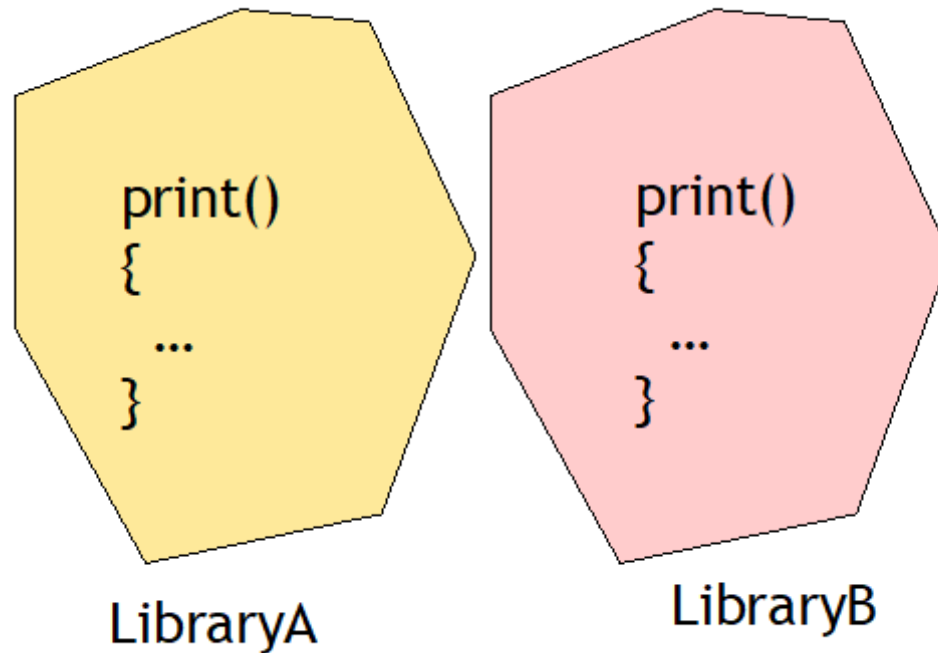
# #include <iostream>

- 현재의 위치에 `iostream`이라는 헤더 파일을 포함



using namespace std;

- 변수 이름이나 함수 이름과 같은 수많은 이름(식별자)들은 이름 공간 (name space)이라고 하는 영역으로 분리되어 저장



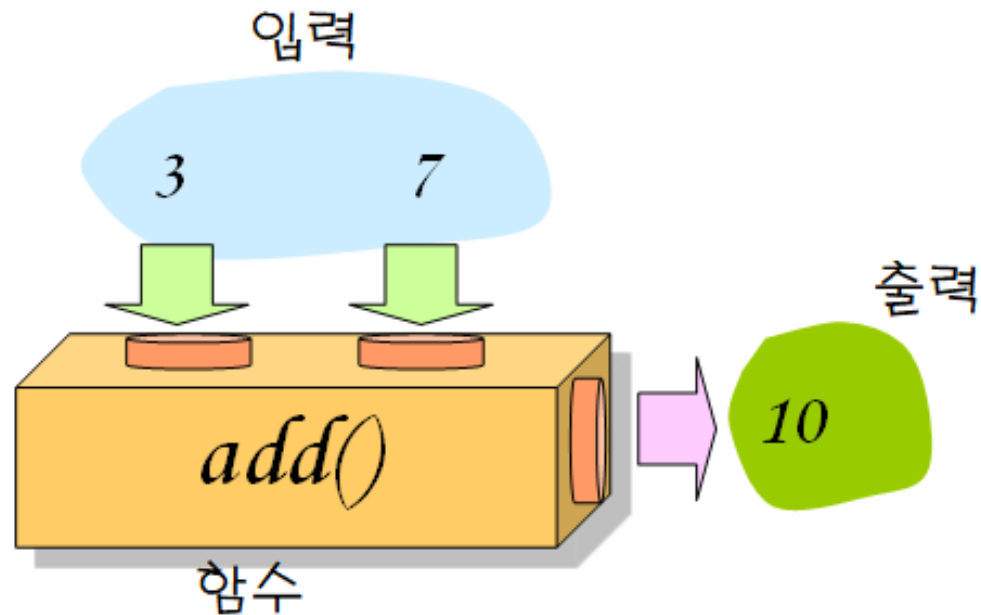
## 식별자를 사용하는 2가지 방법

1. `std::cout << "Hello World!" << std::endl;`
2. `using namespace std;`  
`cout << "Hello World!" << endl;`



```
int main()
```

- 위의 문장은 `main()` 함수를 정의하는 문장



```
cout << "Hello World! " << endl ;
```

- 콘솔 화면에 "Hello World!"라는 문자열을 출력하고 이어서 endl(end of line)을 출력하는 문장



```
return 0;
```

- 이 문장이 실행되면 `main( )` 함수는 작업을 끝내고 외부로 0값을 반환

# 학습 정리

1. C++ 언어의 역사와 특징에 대하여 설명할 수 있다.
2. C++ 프로그램의 구조를 이해할 수 있다.
3. 학습 환경(비주얼 스튜디오 등)을 설치할 수 있다.
4. 간단한 예제 프로그램을 컴파일하고 실행할 수 있다.
5. 문자열을 출력할 수 있다.



# Visual Studio 설치

# 1. 실습할 수 있는 Visual Studio 버전

표 1-1 윈도우에 설치 가능한 Visual Studio

운영체제	설치 가능한 유료 버전	설치 가능한 무료 버전
Windows 7(sp1) Windows 8 Windows 8.1	Visual Studio 2008, 2010, 2012, 2013, 2015, 2017, 2019	Visual Studio Express 2010 Visual Studio Express 2012, 2013, 2015 for Windows Desktop Visual Studio Community 2013, 2015, 2017, 2019
Windows 10 Windows 11	Visual Studio 2012, 2013, 2015, 2017, 2019, 2022	Visual Studio Express 2012, 2013, 2015 for Windows Desktop Visual Studio Community 2013, 2015, 2017, 2019, 2022

## 2. Visual Studio 설치

01 <https://visualstudio.com/ko> 에서 Visual Studio Community 설치 파일을 다운로드

Microsoft | Visual Studio 개발자 도구 다운로드 구입 구독 무료 Visual Studio

Microsoft 전체 검색 Sign in

# 소프트웨어를 만드는 방법

오늘은 무엇을 [코딩, 빌드, 디버그, 배포, 공동 작업, 분석, 학습]하시겠어요?

Visual Studio가 할 수 있습니다.

## Visual Studio 제품군을 만나보세요

### Visual Studio | Windows

Windows에서 .NET 및 C++ 개발자를 위한 가장 포괄적인 IDE입니다. 소프트웨어 개발의 모든 단계를 향상시키고 개선할 수 있는 다양한 도구와 기능이 완벽하게 포함되어 있습니다.

자세히 보기 →

Visual Studio 다운로드 ▾

- Community 2022
- Professional 2022
- Enterprise 2022

### Visual Studio for Mac | macOS

macOS에 기본 제공되는 .NET 개발자를 위한 포괄적인 IDE입니다. 웹, 클라우드, 모바일 및 게임 개발을 위한 최고 수준의 지원을 포함합니다.

자세히 보기 →

다음에 대해 자세히 알아보기 라이선스 활성화

Visual Studio for Mac 다운로드

### Visual Studio Code | Windows, macOS, Linux

Windows, macOS 및 Linux에서 실행되는 독립 실행형 소스 코드 편집기입니다. JavaScript 및 웹 개발자를 위한 최고의 선택이며, 거의 모든 프로그래밍 언어를 지원할 수 있는 확장 기능을 제공합니다.

자세히 보기 →

Visual Studio Code를 사용하면 다음에 동의하는 것입니다. 라이선스 & 개인정보 처리방침

Visual Studio 코드 다운로드 ▾

① C... 고 싶으신가요? 저희가 도와드릴게요.

파드백

## 2. Visual Studio 설치

### 02 다운로드된 파일 실행

### 03 비주얼 스튜디오 설치 진행



그림 1-9 설치 초기 화면

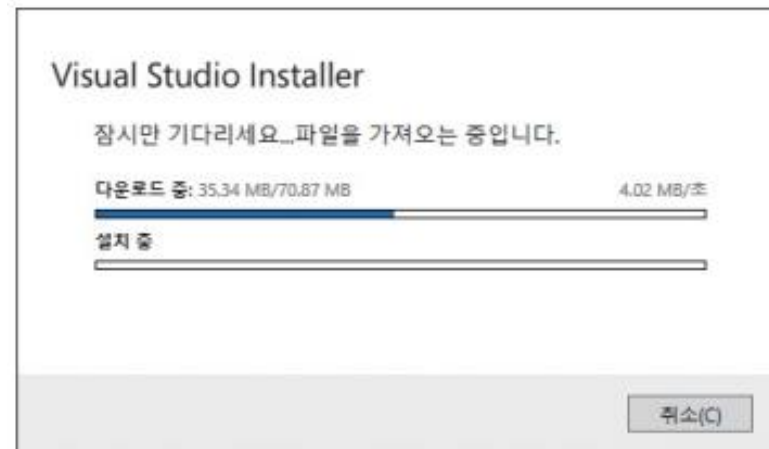


그림 1-10 설치 초기 진행



## 2. Visual Studio 설치

### 04 [워크로드] - [C++를 사용한 데스크톱 개발]만 체크한 뒤 <설치>를 클릭

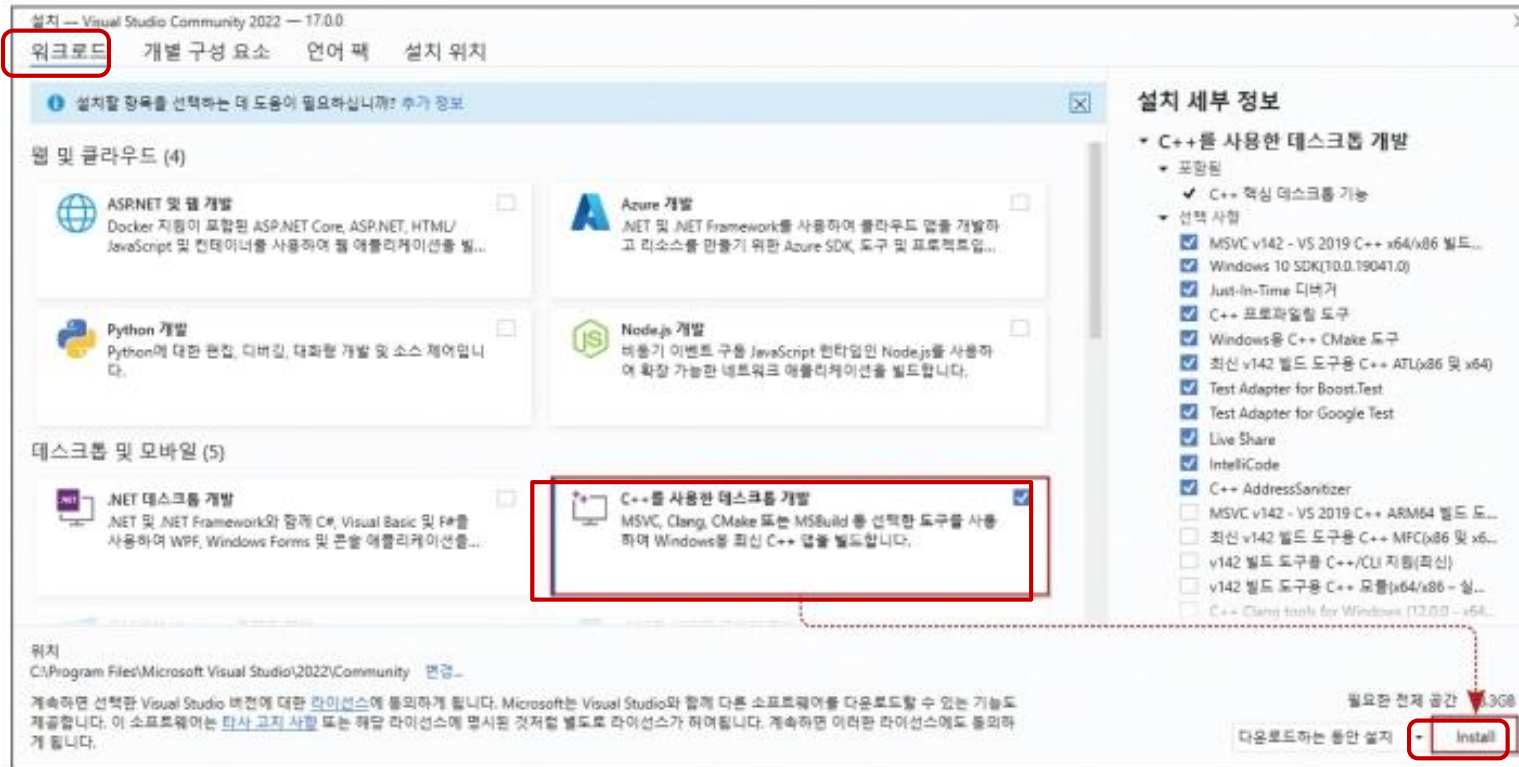


그림 1-11 설치할 기능 선택

**TIP |** 'C++을 사용한 데스크톱 개발' 항목이 보이지 않으면 화면을 아래로 스크롤,  
프로그램을 설치하려면 10GB 정도의 하드디스크 여유 공간이 필요

## 2. Visual Studio 설치

**05** '설치 후 시작'이 체크되어 있으면 설치 완료 후에 Visual Studio가 자동으로 시작



그림 1-12 설치 후 자동 시작 체크

## 2. Visual Studio 설치

### 06 설치가 진행(컴퓨터 성능에 따라서 수 분에서 수십 분이 걸림)

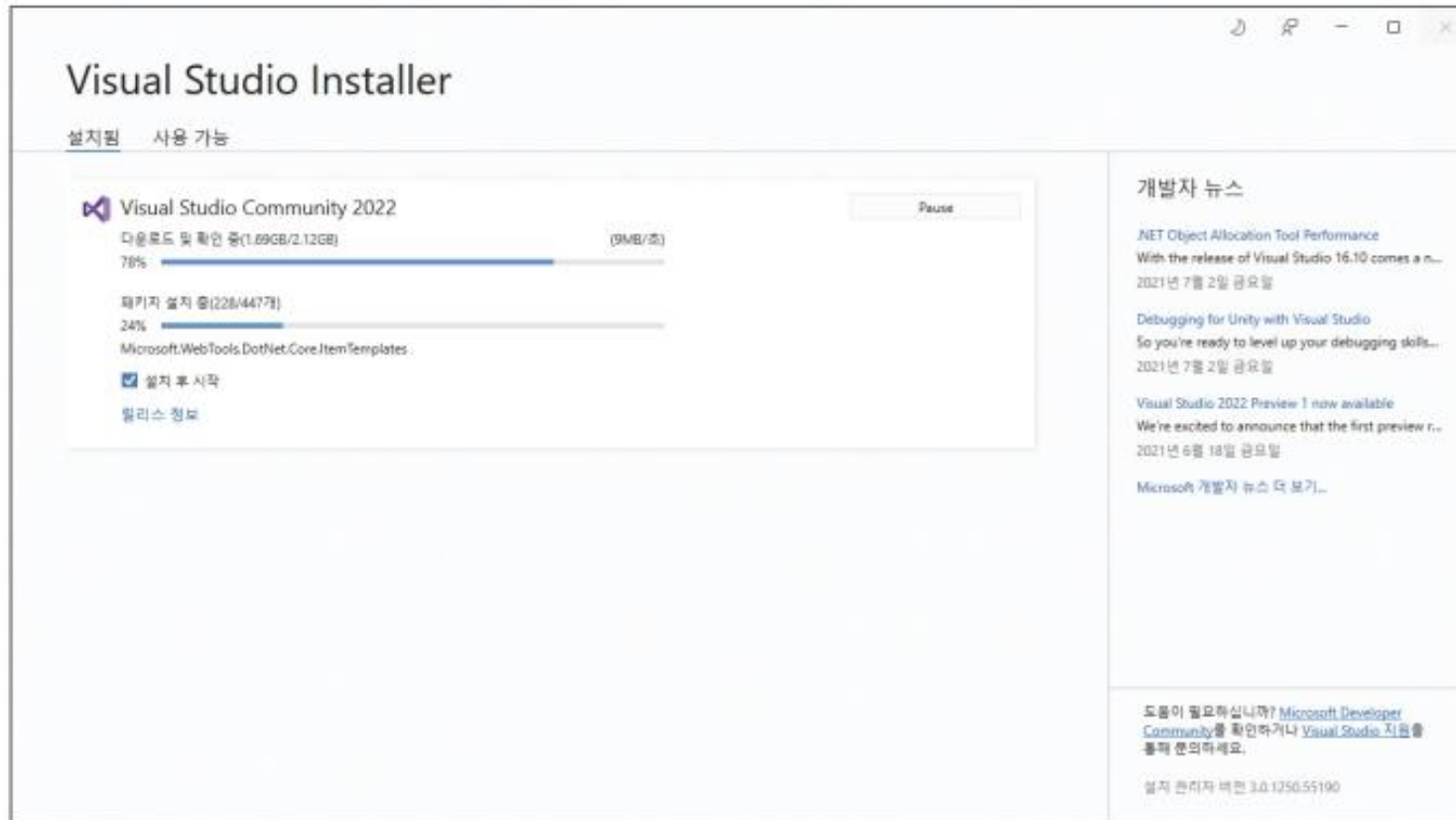


그림 1-13 설치 진행

## 2. Visual Studio 설치

**07** 설치 완료 후, 다시 부팅하라는 메시지 창이 나오면 컴퓨터를 재부팅

**08** 윈도우의 [시작] 클릭 → <모든 앱> 클릭 → [V] 입력 → [Visual Studio 2022 Currents]  
또는 [Visual Studio 2022]를 클릭해서 Visual Studio 실행



그림 1-14 Visual Studio 시작

## 2. Visual Studio 설치

**09** [나중에 로그인] 클릭, '친숙한 환경에서 시작' 창에서 원하는 화면 색상을 선택 후 [Visual Studio]을 클릭

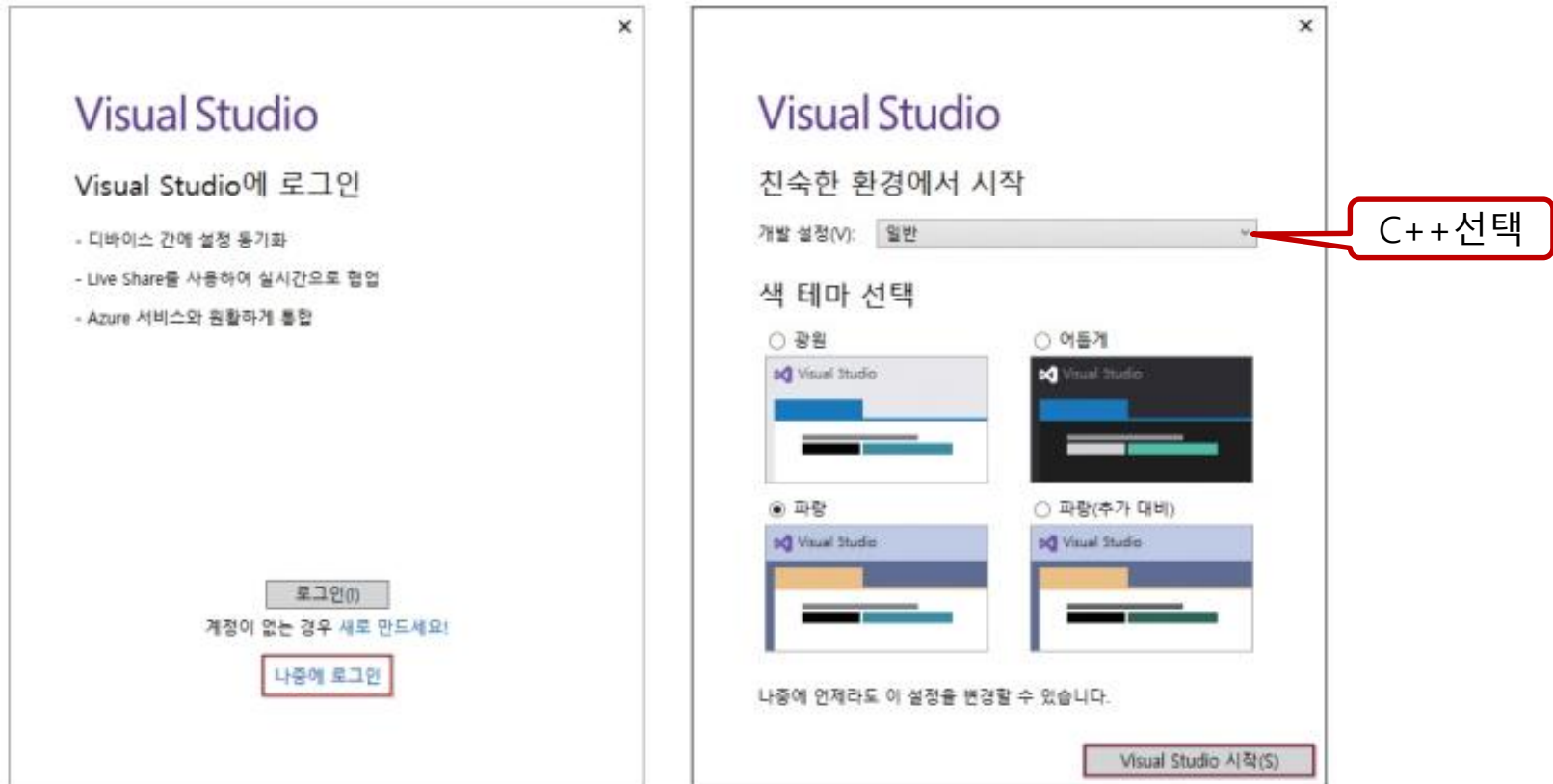


그림 1-15 로그인 및 친숙한 환경에서 시작

## 2. Visual Studio 설치

### 10 Visual Studio 초기 화면이 나옴



그림 1-16 Visual Studio 초기 화면

## 2. Visual Studio 설치

**11** Visual Studio Community는 마이크로소프트 계정으로 로그인해야 기간에 제한 없이 무료로 사용할 수 있음(별도의 로그인 절차 없이 30일간 무료로 사용할 수 있으나, 30일이 지나면 다시 로그인해야 됨)

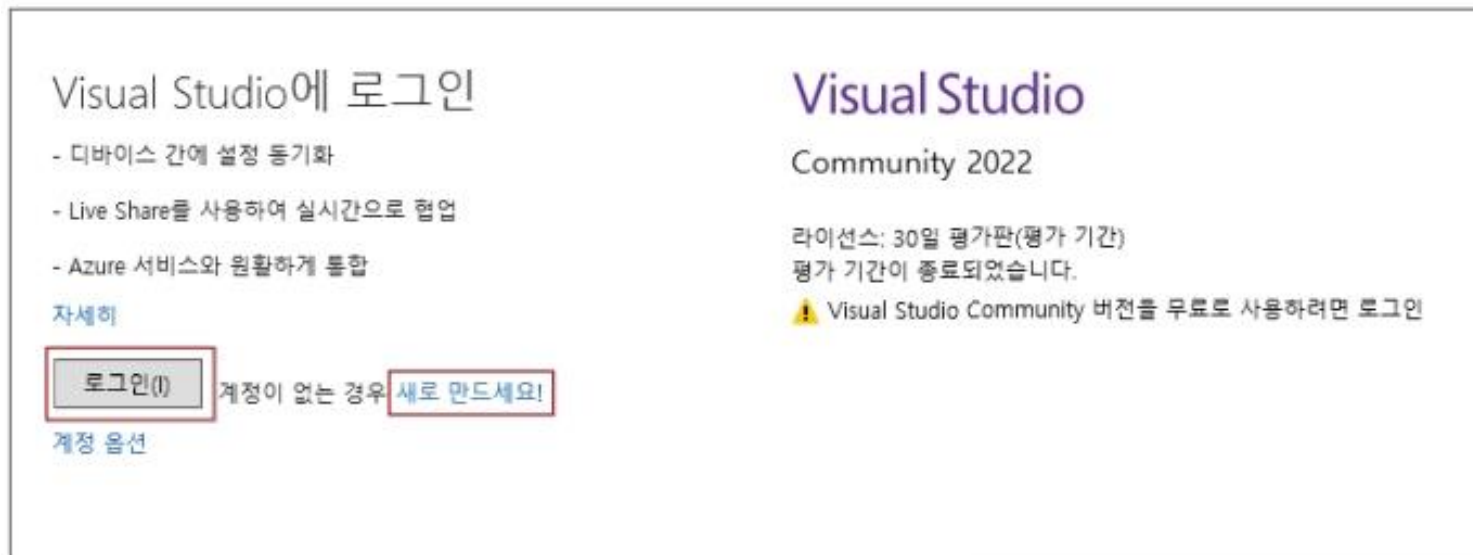


그림 1-17 [Visual Studio에 로그인] 화면

### 3. 간단한 프로그램 실습

#### ■ Visual Studio

- 강력하고 편리한 통합 개발 환경을 제공
- 코드를 입력 시 잘못 입력한 글자를 찾아주고
- 컴파일과 링크를 한 번으로 해결해준다.

#### 프로젝트와 소스 파일의 상관관계

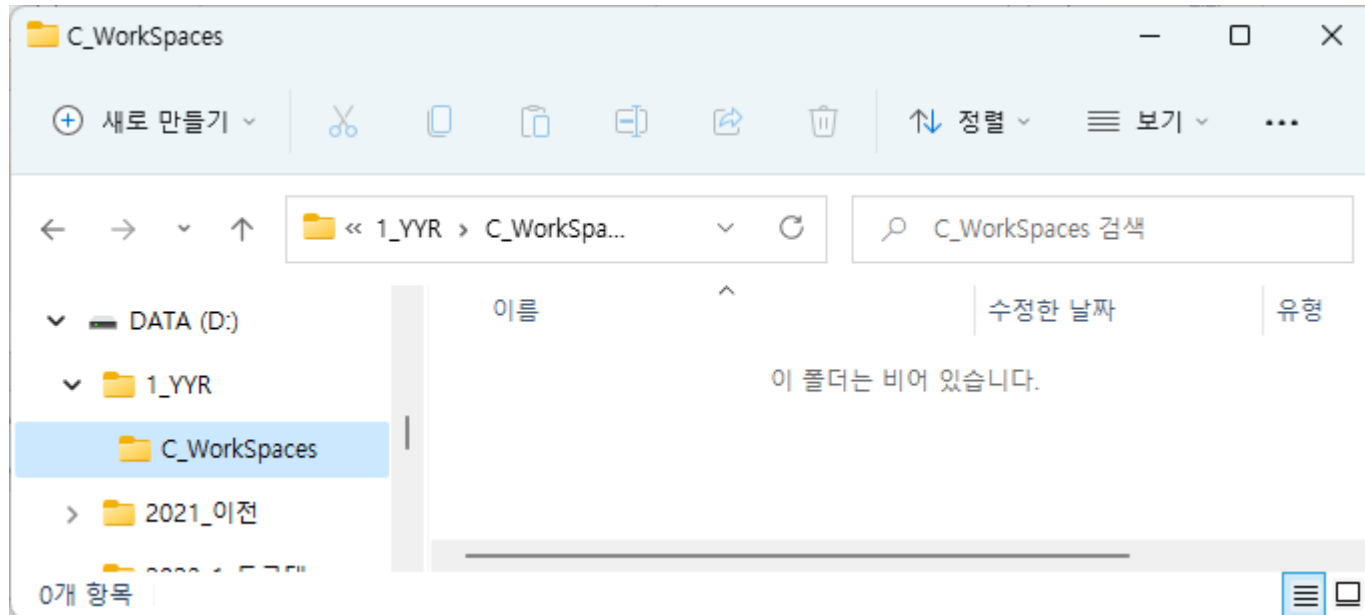
- 단순한 C++ 컴파일러(예를 들어 Turbo C++ 등)를 사용한다면 C++ 소스 파일을 하나만 작성해서 컴파일과 링크를 수행하면 됨
- 하지만 Visual C++로 프로그래밍을 하려면 단순히 소스 파일만 필요한 것이 아니라 프로젝트를 먼저 생성해야 함
- 프로젝트는 지금 공부하는 C++뿐만 아니라 고급 프로그래밍에서 중요한 개념
- 프로젝트의 개념은 여러 개의 C++ 프로그램을 담아놓는 그릇이라고 보면 됨



### 3. 간단한 프로그램 실습

- 프로젝트 생성

01 실습 폴더 생성 : 'D:\YYR\CPP\_WorkSpace'



### 3. 간단한 프로그램 실습

02 윈도우의 [시작] - [Visual Studio 2022]를 클릭해서 다시 실행

03 [새 프로젝트 만들기] 클릭

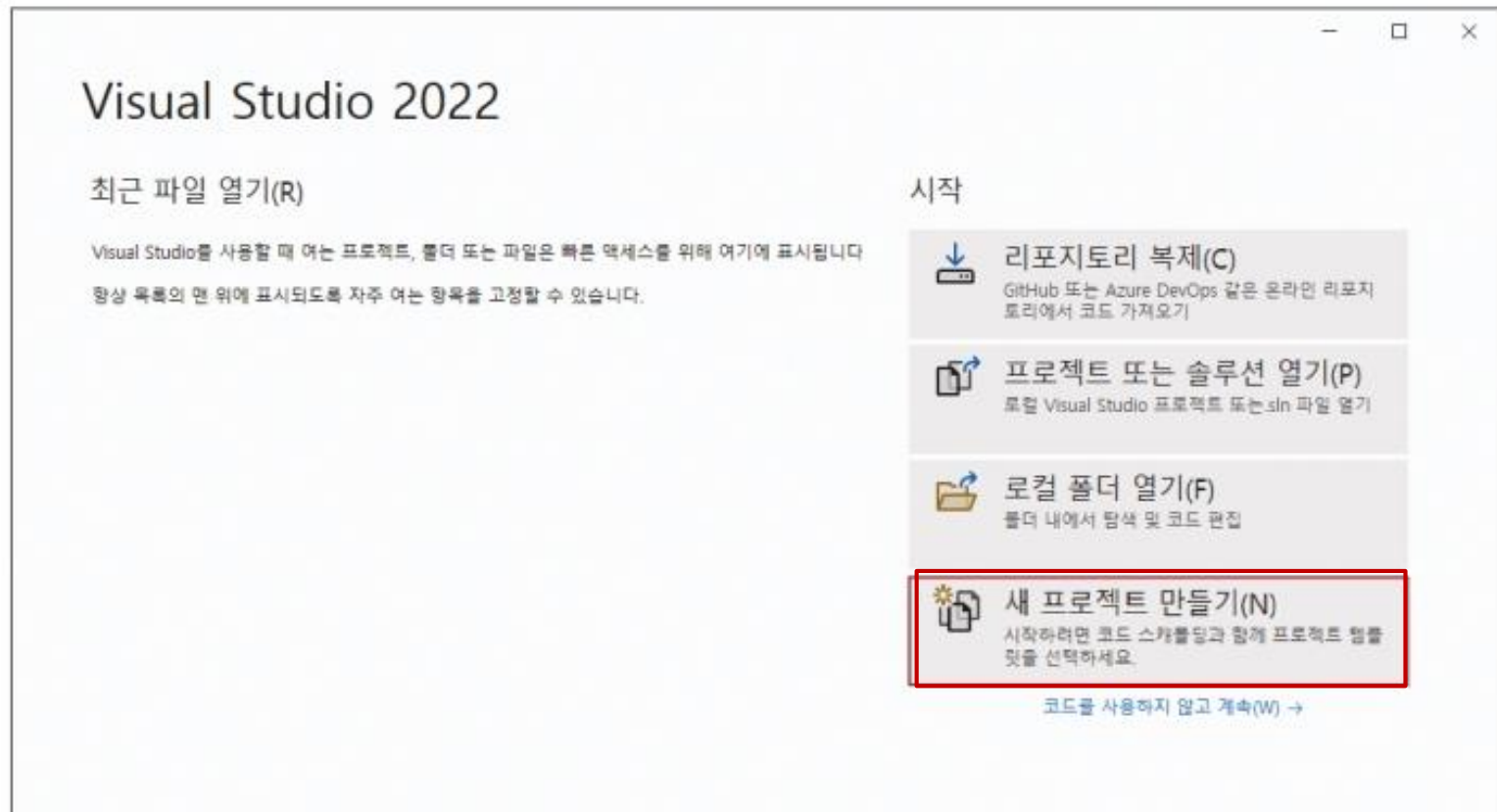


그림 1-19 새 프로젝트 생성 1

### 3. 간단한 프로그램 실습

**04** [모든 언어]를 [C++]로 선택하고, [Windows 데스크톱 마법사]를 선택 후 [다음]



그림 1-20 새 프로젝트 생성 2

# 3. 간단한 프로그램 실습

## 05 새 프로젝트 구성

새 프로젝트 구성

Windows 데스크톱 마법사 C++ Windows 데스크톱 콘솔 라이브러리

프로젝트 이름(I)  
First

위치(L)  
D:\1\_YR\WC\_WorkSpaces\

솔루션 이름(M) ⓘ  
First

☒ 솔루션 및 프로젝트를 같은 디렉터리에 배치(D)

뒤로(B) 만들기(C)

### 3. 간단한 프로그램 실습

**06** 애플리케이션 종류는 '콘솔 애플리케이션(.exe)'으로 되어 있을 것, 추가 옵션에서 '빈 프로젝트'에 체크하고 <확인>을 클릭

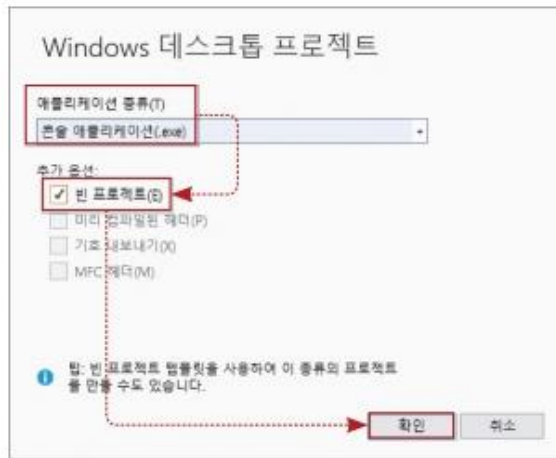


그림 1-22 설정

**07** 빈 프로젝트가 완성

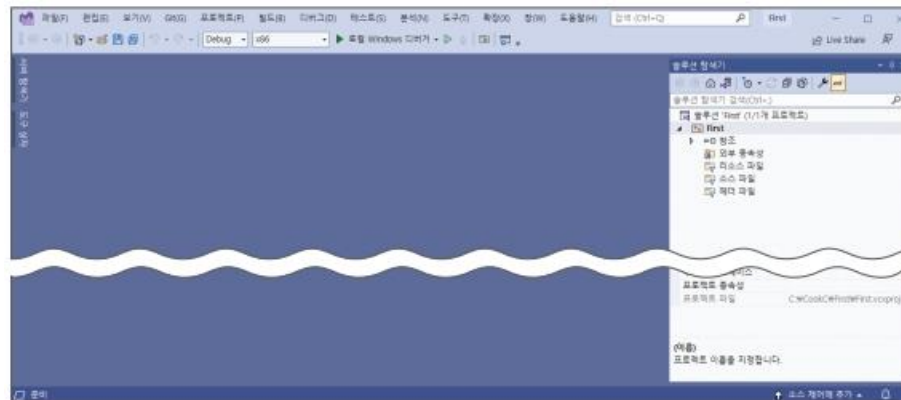
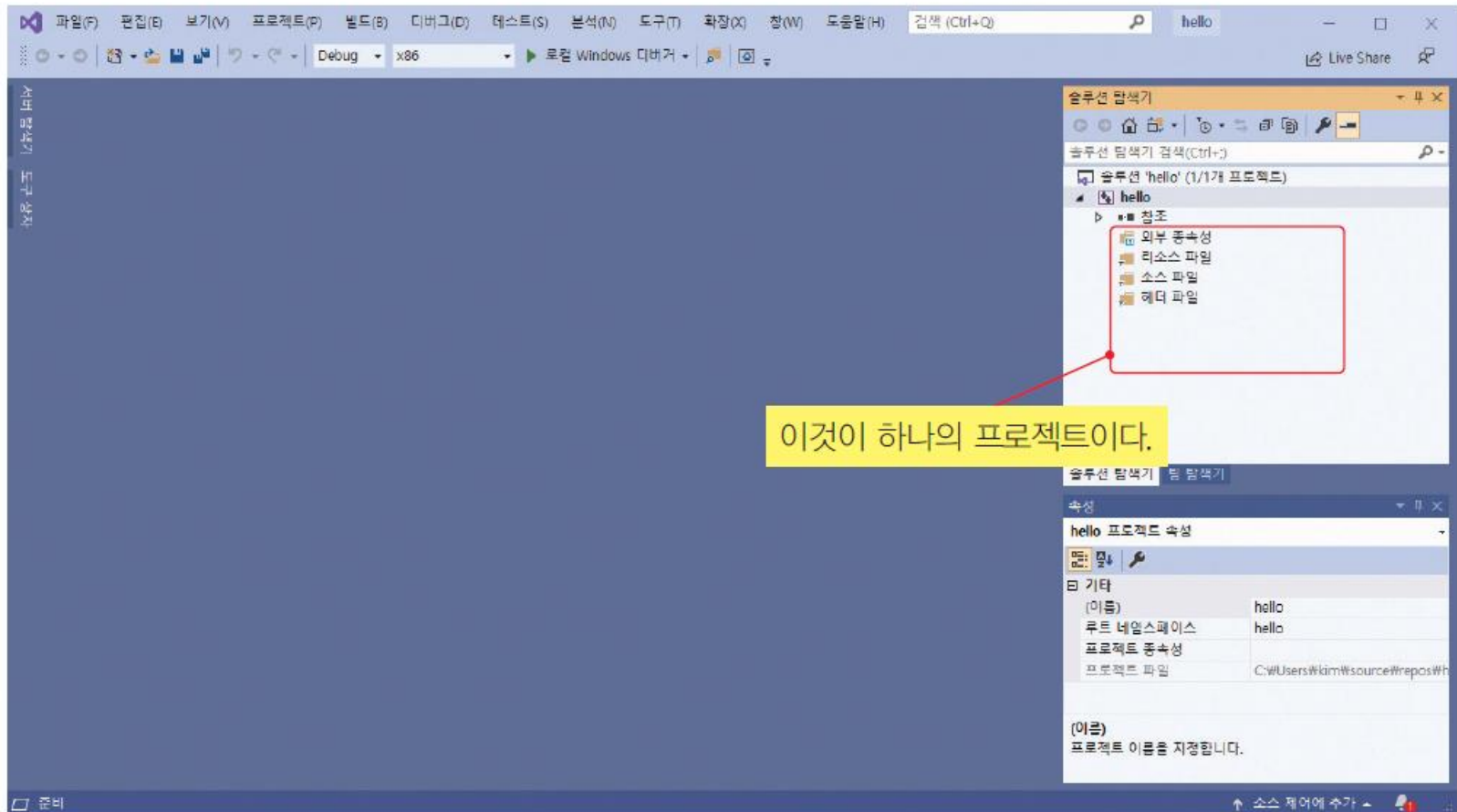


그림 1-23 프로젝트 생성 화면

### 3. 간단한 프로그램 실습

#### 07 빈 프로젝트 완성



### 3. 간단한 프로그램 실습

#### ■ C 프로그램 코딩

- 01 '소스 파일' 폴더에서 마우스 오른쪽 버튼을 클릭  
[추가] - [새 항목] 선택

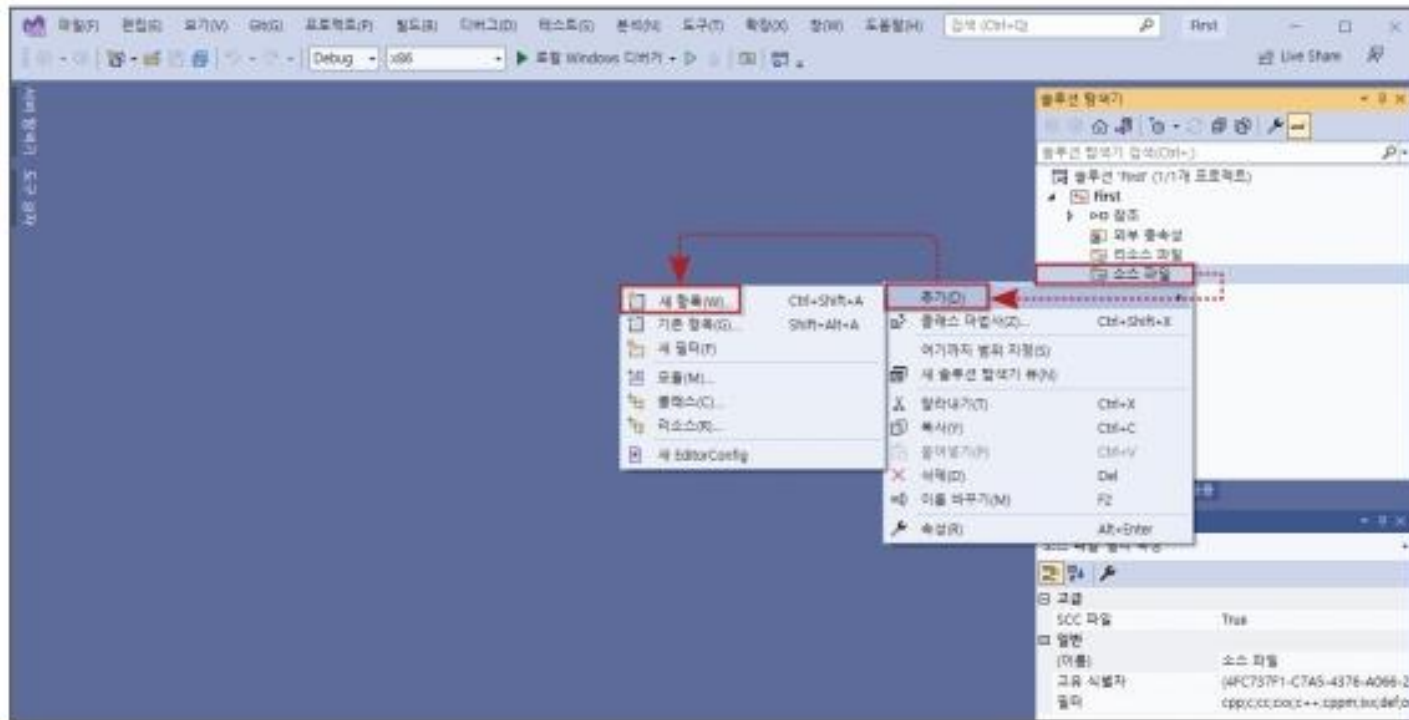


그림 1-24 프로젝트에 소스 파일 추가

### 3. 간단한 프로그램 실습

**02** [새 항목 추가] 창에서 왼쪽의 [Visual C++]를 선택하고, 'C++ 파일(.cpp)'을 선택한 뒤 이름에 'First.c'를 입력하고 <추가>를 클릭

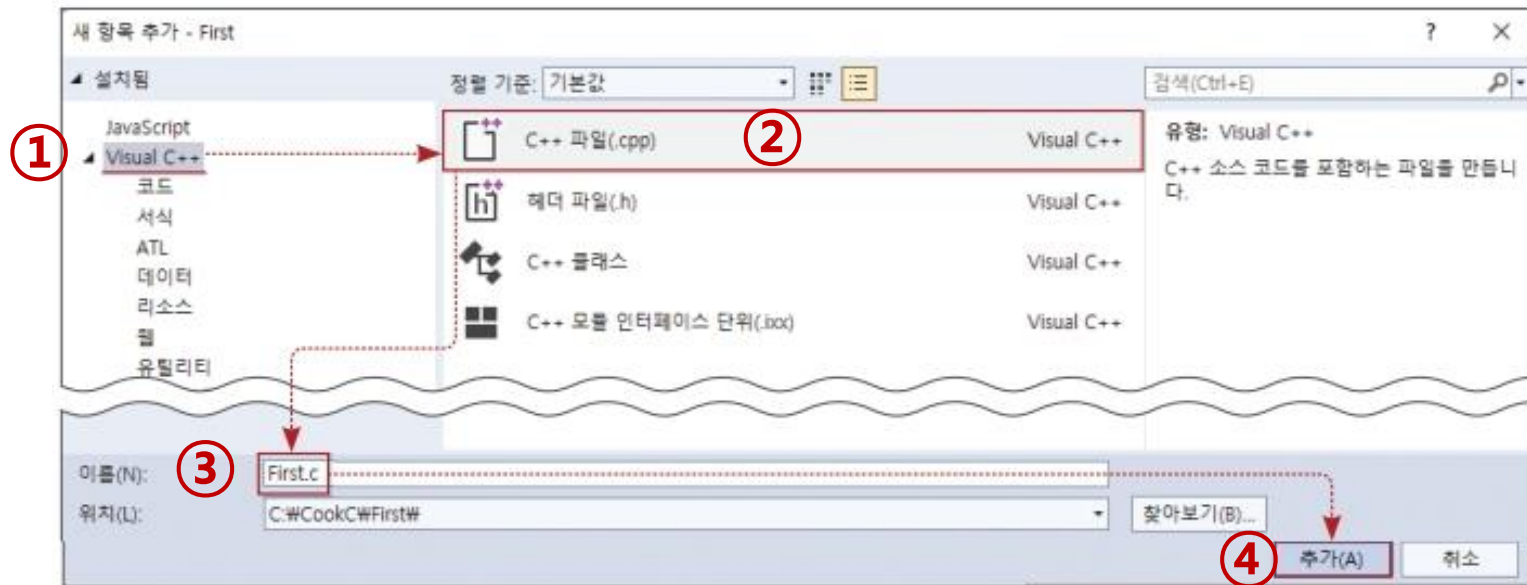


그림 1-25 First.c 파일 생성



### 3. 간단한 프로그램 실습

#### 03 왼쪽에 빈 창에 소스 코드 입력 - [파일] - [First.c 저장]

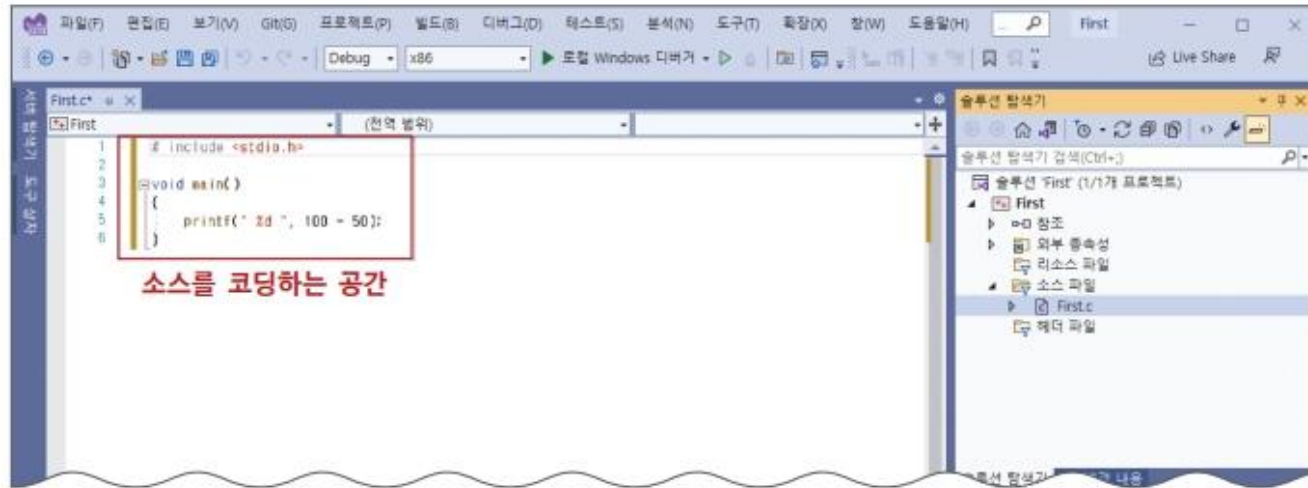


그림 1-26 C 소스 입력

```
01 #include<stdio.h>
02
03 int main( )
04 {
05     printf("%d", 100 - 50);
06     return 0;
07 }
```

### 3. 간단한 프로그램 실습

#### ■ 빌드(컴파일 및 링크)

01 [빌드] - [솔루션 빌드], 단축키인 [Ctrl]+[Shift]+[B]

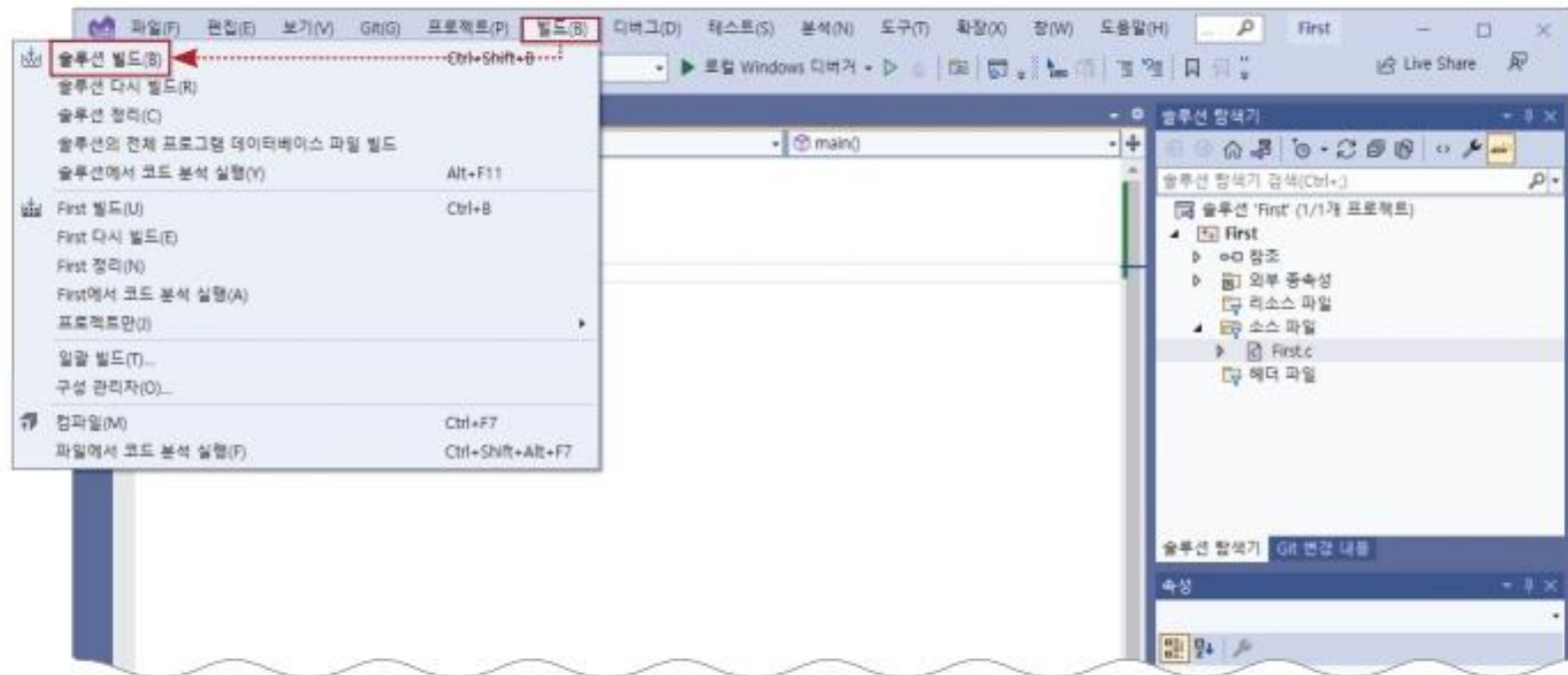
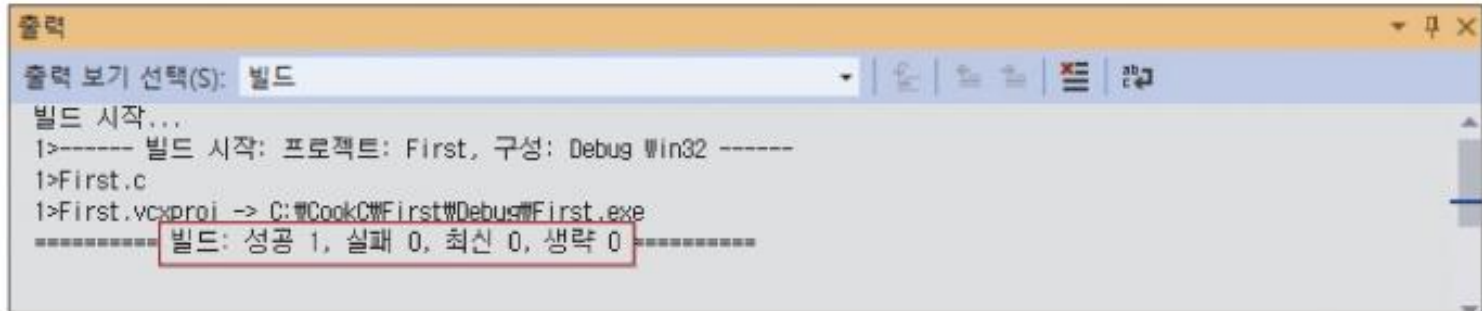


그림 1-27 빌드 실행

### 3. 간단한 프로그램 실습

#### 02 소스에 오류가 없을 경우



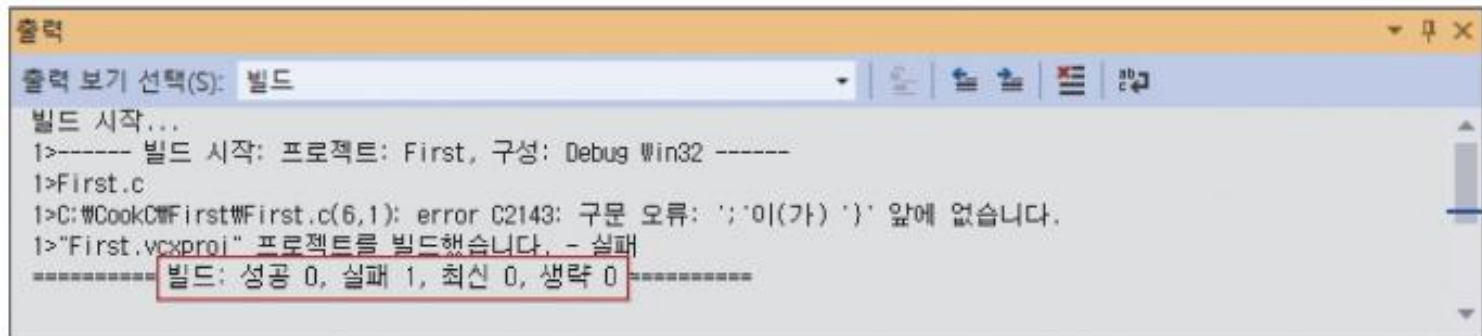
The screenshot shows the 'Output' window in Visual Studio with the 'Build' filter selected. The text in the window is as follows:

```
출력
출력 보기 선택(S): 빌드
빌드 시작...
1>----- 빌드 시작: 프로젝트: First, 구성: Debug Win32 -----
1>First.c
1>First.vcxproj -> C:\CookC\First\Debug\First.exe
===== 빌드: 성공 1, 실패 0, 최신 0, 생략 0 =====
```

그림 1-28 빌드 결과: 이상 없음

#### 03 소스에 오류가 있을 경우

- 5행에서 세미콜론 (;)을 지우고, [F7]을 눌러서 다시 빌드한다.



The screenshot shows the 'Output' window in Visual Studio with the 'Build' filter selected. The text in the window is as follows:

```
출력
출력 보기 선택(S): 빌드
빌드 시작...
1>----- 빌드 시작: 프로젝트: First, 구성: Debug Win32 -----
1>First.c
1>C:\CookC\First\First.c(6,1): error C2143: 구문 오류: ';'이(가) ')' 앞에 없습니다.
1>"First.vcxproj" 프로젝트를 빌드했습니다. - 실패
===== 빌드: 성공 0, 실패 1, 최신 0, 생략 0 =====
```

그림 1-29 빌드 결과: 오류 발생

### 3. 간단한 프로그램 실습

- 04 [오류 목록] 탭 클릭 - 오류가 있는 부분을 마우스로 더블클릭  
→ 해당 위치로 바로 이동

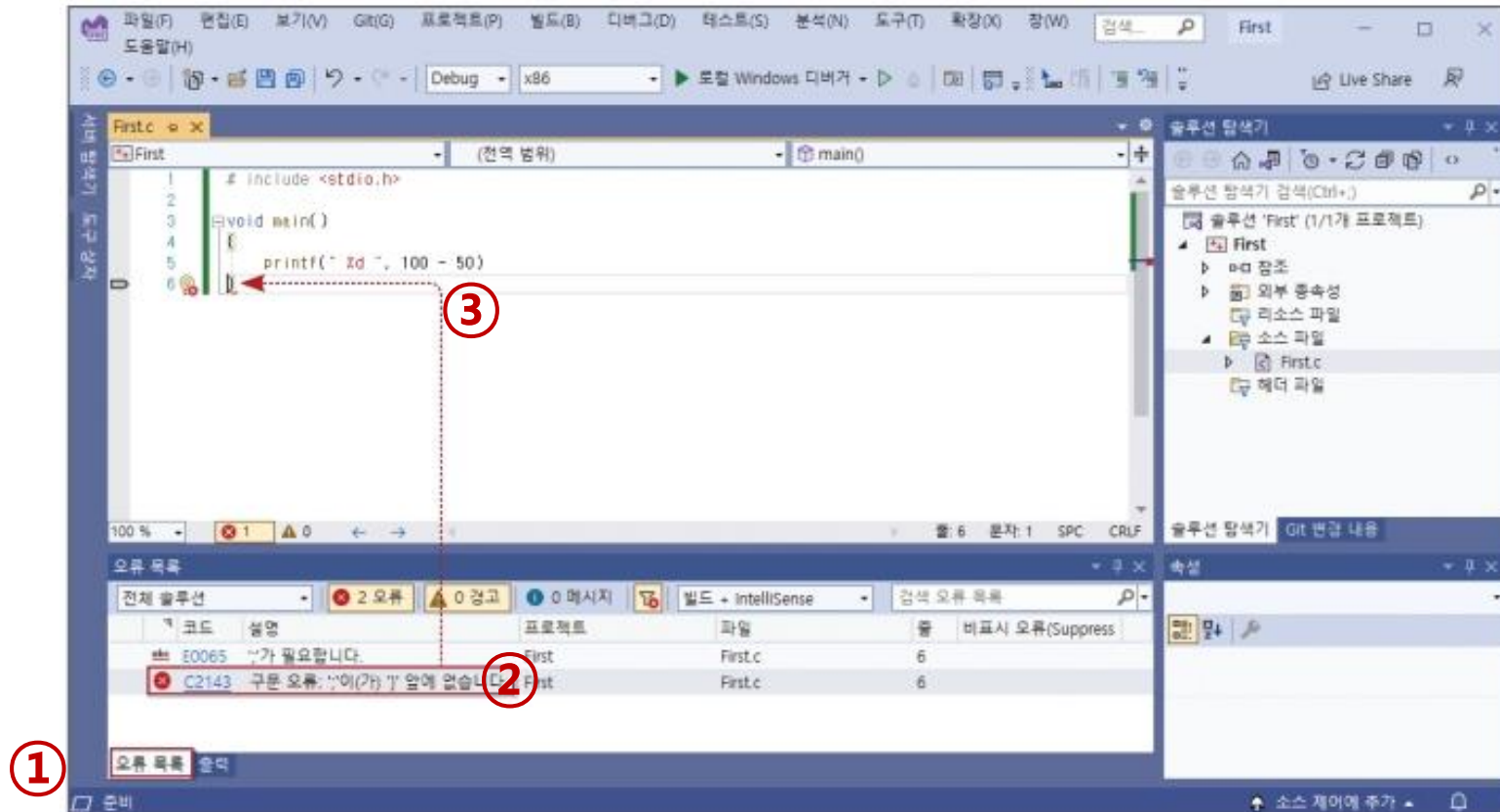


그림 1-30 소스에서 틀린 부분 찾기

### 3. 간단한 프로그램 실습

05 오류 수정 후 파일을 저장하고 [Ctrl]+[Shift]+[B]를 눌러서 다시 빌드

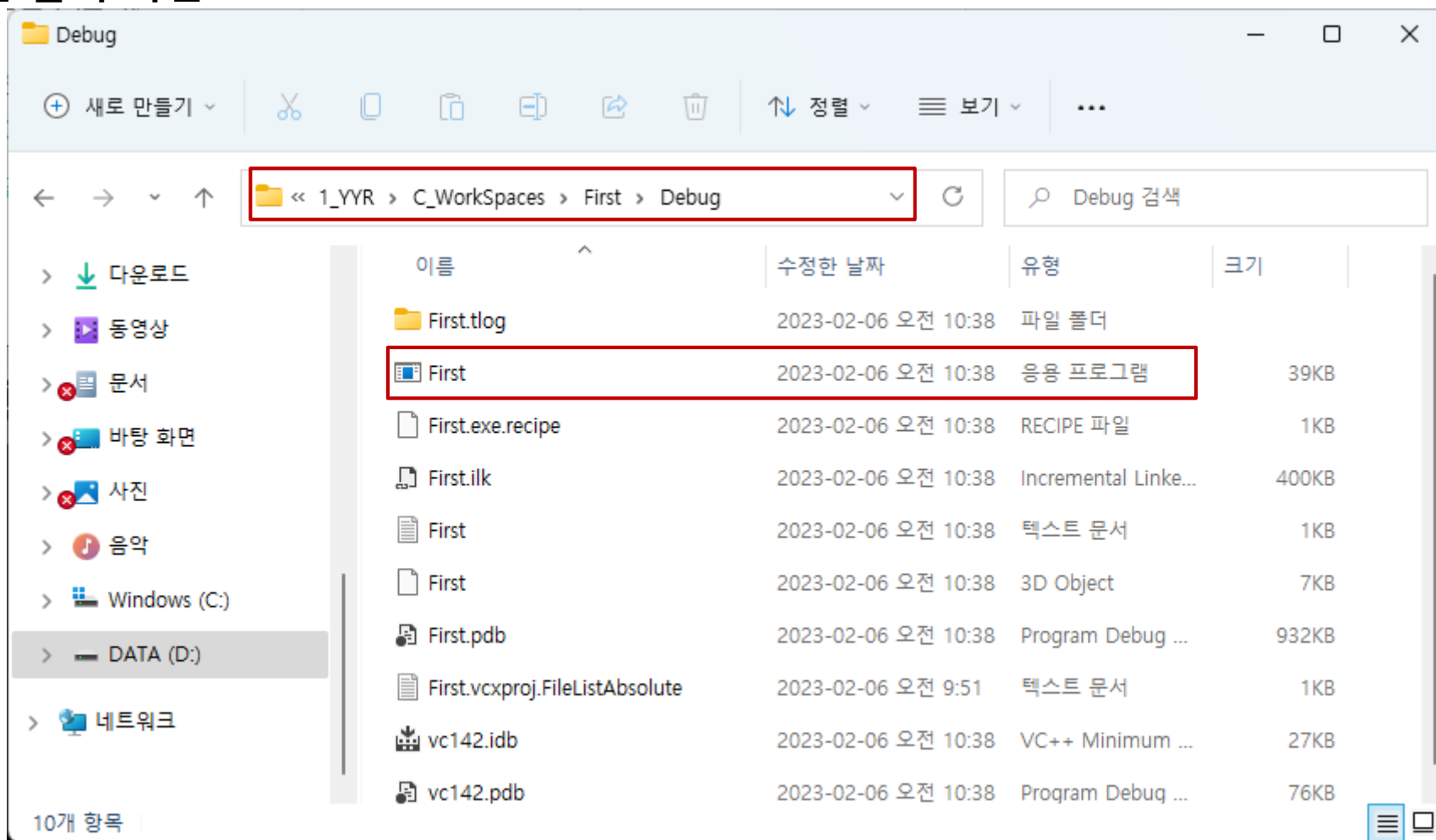
Q / U / I / Z | 세미콜론이 빠지면 세미콜론이 빠진 줄에 오류가 표시된다. (○ ×)

#### C 소스의 줄 바꿈

- 문장을 끝내는 방법
  - ① 세미콜론(;
  - ② 닫는 중괄호( } ) 등을 만나야 하나의 문장이 종료
- 다음 네 문장의 의미
  - ① `printf("%d", 100-50);`
  - ② `printf("%d",`  
`100-50);`
  - ③ `printf("%d",`  
`100-50)`  
`;`
  - ④ `printf`  
`("%d", 100-50);`

### 3. 간단한 프로그램 실습

#### ■ 작업 폴더 확인



### 3. 간단한 프로그램 실습

#### ■ 파일 실행

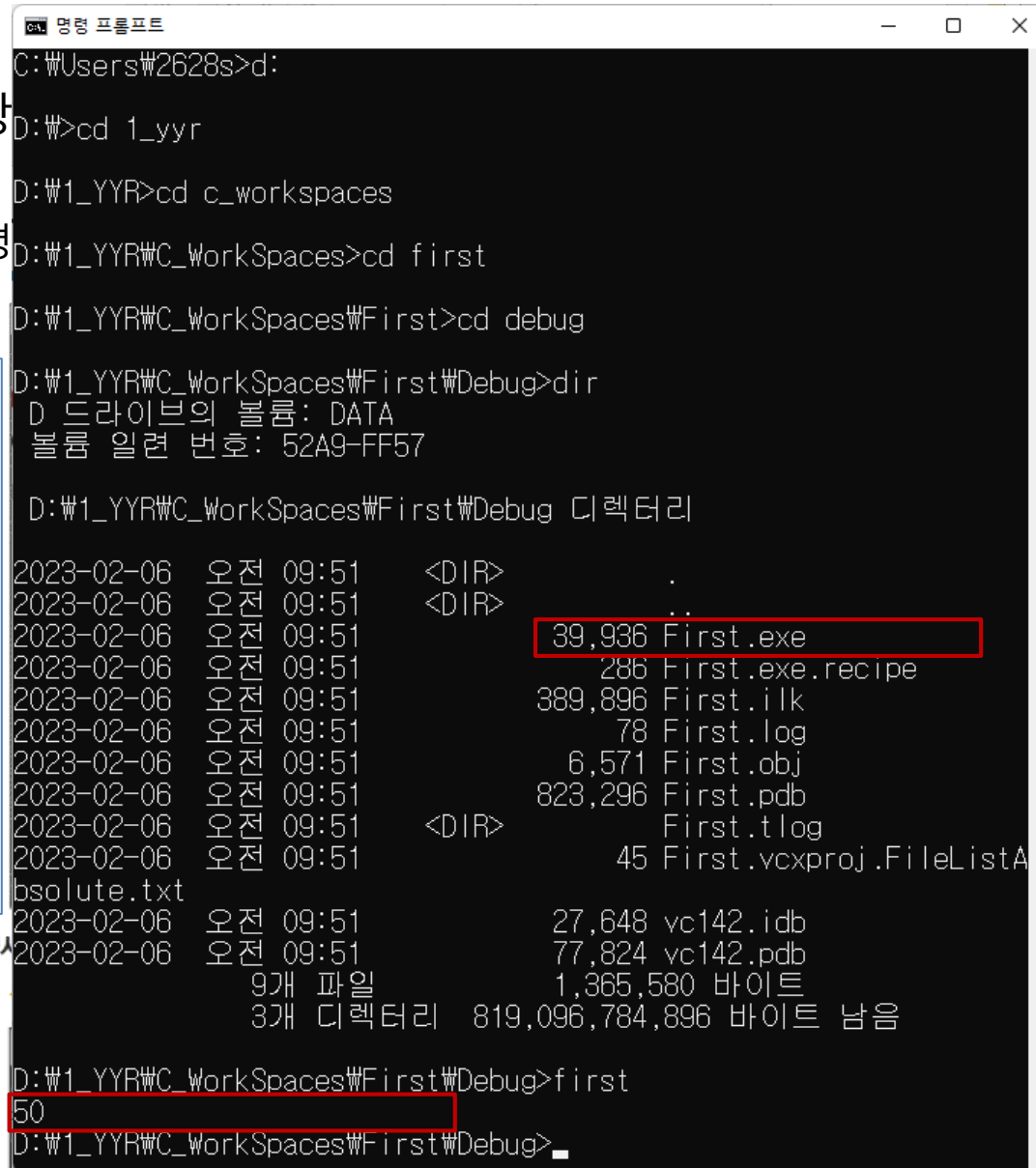
- 명령 프롬프트 창에서 실행하는 방법

**01** 명령 프롬프트 창에서 직접 실행

Windows키+[R] → cmd 입력 → [Enter] → 명령  
다음 명령을 차례대로 실행

```
> D:
> cd YYR
> cd C_WorkSpace
> cd first
> cd Debug
> dir
> First
```

그림 1-31 명령 프롬프트 창에서



```
cmd 명령 프롬프트
C:\Users\2628s>d:
D:\>cd 1_yyr
D:\1_Yyr>cd c_workspaces
D:\1_Yyr\C_WorkSpaces>cd first
D:\1_Yyr\C_WorkSpaces\First>cd debug
D:\1_Yyr\C_WorkSpaces\First\Debug>dir
D 드라이브의 볼륨: DATA
볼륨 일련 번호: 52A9-FF57

D:\1_Yyr\C_WorkSpaces\First\Debug 디렉터리

2023-02-06 오전 09:51 <DIR> .
2023-02-06 오전 09:51 <DIR> ..
2023-02-06 오전 09:51 39,936 First.exe
2023-02-06 오전 09:51 286 First.exe.recipe
2023-02-06 오전 09:51 389,896 First.ilc
2023-02-06 오전 09:51 78 First.log
2023-02-06 오전 09:51 6,571 First.obj
2023-02-06 오전 09:51 823,296 First.pdb
2023-02-06 오전 09:51 <DIR> First.tlog
2023-02-06 오전 09:51 45 First.vcxproj.FileListAbsolute.txt
2023-02-06 오전 09:51 27,648 vc142.idb
2023-02-06 오전 09:51 77,824 vc142.pdb
9개 파일 1,365,580 바이트
3개 디렉터리 819,096,784,896 바이트 남음

D:\1_Yyr\C_WorkSpaces\First\Debug>first
50
D:\1_Yyr\C_WorkSpaces\First\Debug>
```

### 3. 간단한 프로그램 실습

**02** '100-99'의 결과가 나오도록 소스를 수정하여 결과를 확인하시오.

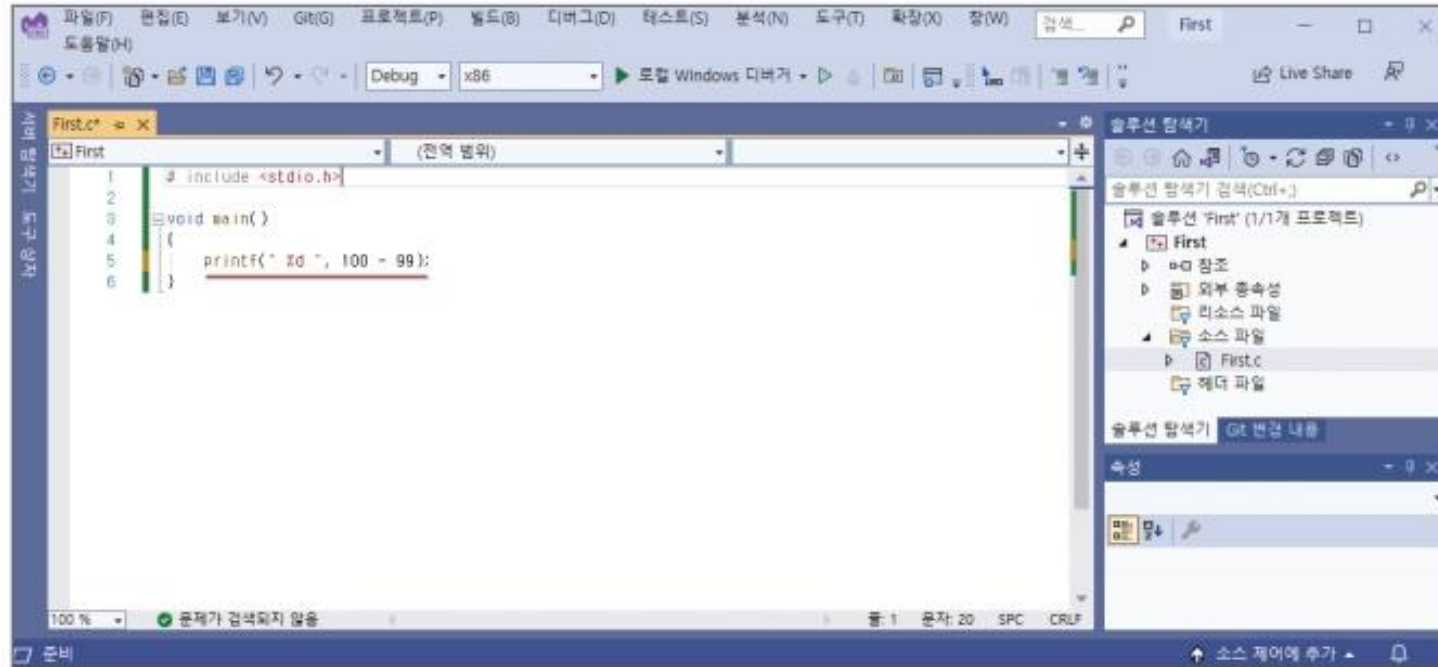


그림 1-32 소스 수정

**03** 새로운 결과를 보기 위해서 다시 명령 프롬프트 창에서 실행  
실행 결과▼





### 3. 간단한 프로그램 실습

**04** 결과가 바뀌지 않은 이유를 설명하고 해결하시오.

실행 결과▼



```
C:\Windows\system32\cmd.exe
C:\CookCM\First\wx64\Debug> First
50
C:\CookCM\First\wx64\Debug> First
50
C:\CookCM\First\wx64\Debug> First
1
C:\CookCM\First\wx64\Debug>
```

#### 기본 도스 명령

명령(대·소문자 구분 없음)	설명
DIR	현재 폴더의 파일 목록을 보여준다.
CD [폴더 이름]	폴더로 이동한다.
COPY [원본] [사본]	원본 파일을 사본 파일로 복사한다.
TYPE [파일 이름]	텍스트 파일의 내용을 화면에 출력한다.
DEL [파일 이름]	파일을 삭제한다.
MD [폴더 이름]	새로운 폴더를 생성한다.
RD [폴더 이름]	폴더를 삭제한다(단, 폴더가 비어 있어야 한다).
CLS	화면을 깨끗이 지운다.

### 3. 간단한 프로그램 실습

- Visual Studio 환경에서 바로 실행하는 방법

**01** [디버그]-[디버그하지 않고 시작]을 선택하거나, 단축키인 [Ctrl]+[F5]를 선택  
만약 프로젝트가 완료되었다는 창이 나오면 <예(Y)>를 클릭

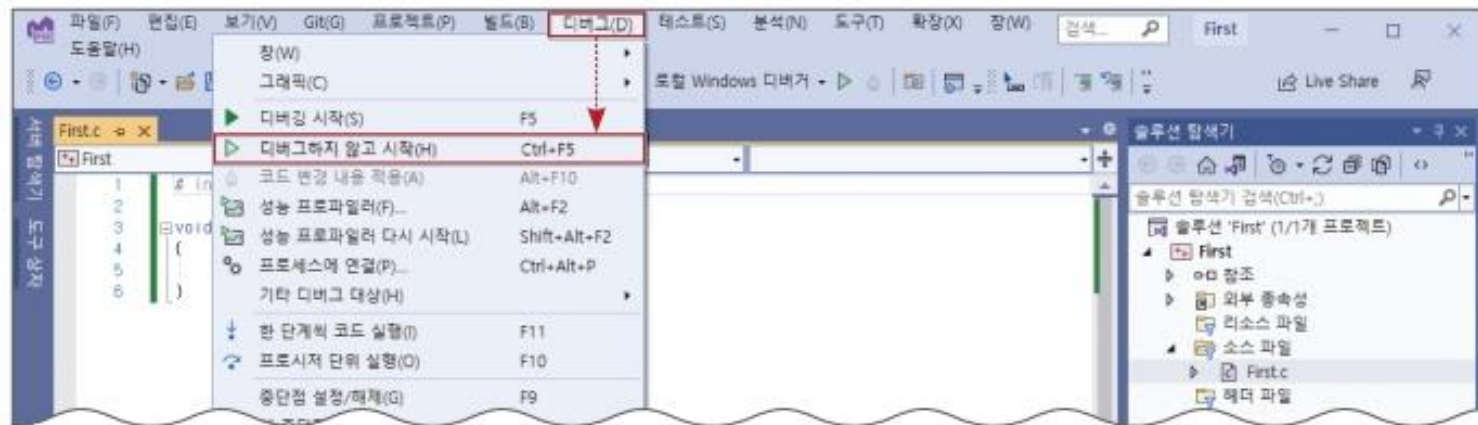
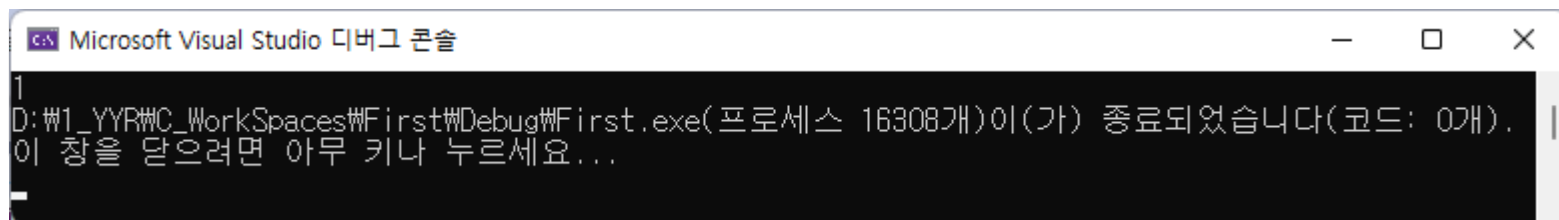


그림 1-33 Visual Studio에서 실행

### 3. 간단한 프로그램 실습

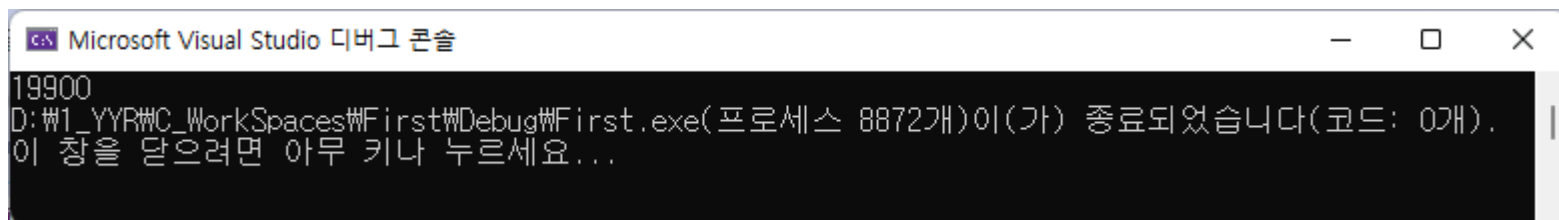
**02** 명령 프롬프트 창으로 실행 결과가 보인다.

실행 결과▼




```
Microsoft Visual Studio 디버그 콘솔
1
D:\₩1_YYR₩C_WorkSpaces₩First₩Debug₩First.exe(프로세스 16308개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

**03** '100\*99'의 결과도 함께 보이시오.



```
Microsoft Visual Studio 디버그 콘솔
19900
D:\₩1_YYR₩C_WorkSpaces₩First₩Debug₩First.exe(프로세스 8872개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

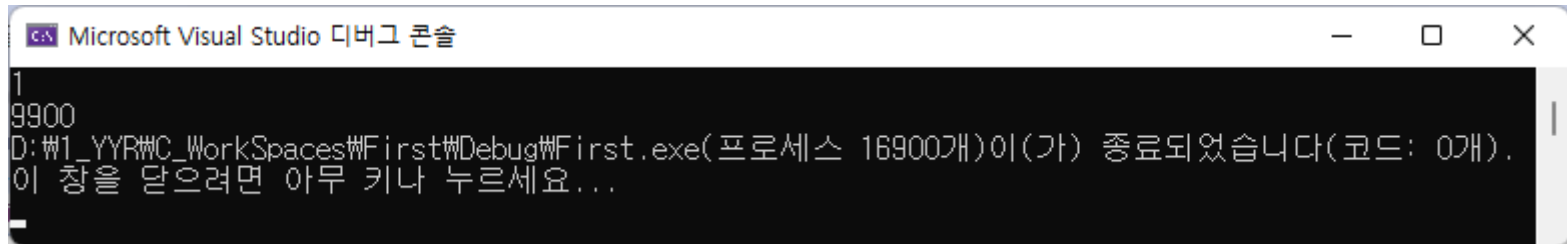


```
Microsoft Visual Studio 디버그 콘솔
1
9900
D:\₩1_YYR₩C_WorkSpaces₩First₩Debug₩First.exe(프로세스 16900개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

### 3. 간단한 프로그램 실습

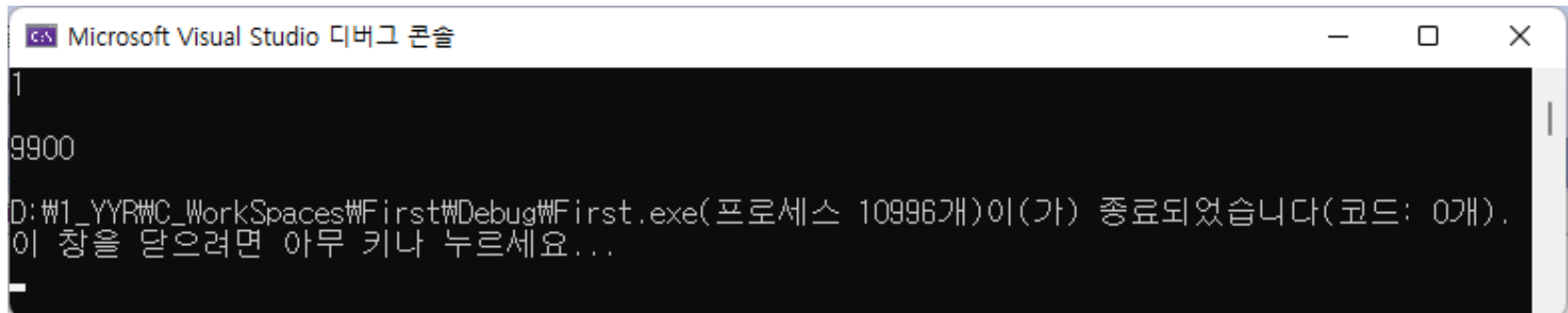
**04** 두 결과값을 구분하기 위해 줄을 바꾸어 출력하시오.

실행 결과▼



```
Microsoft Visual Studio 디버그 콘솔
1
9900
D:\₩1_YYR₩C_WorkSpaces₩First₩Debug₩First.exe(프로세스 16900개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

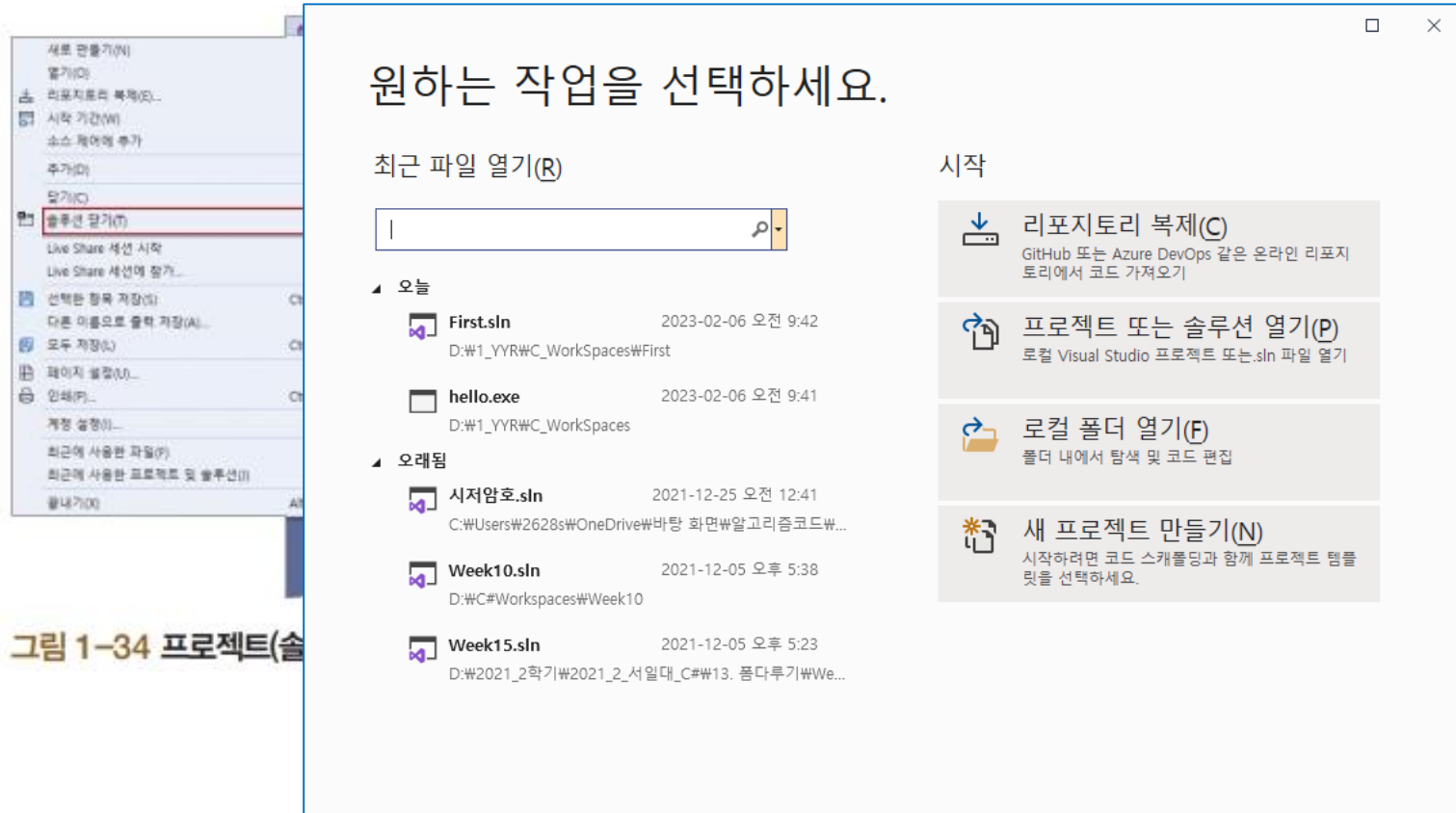
hint :



```
Microsoft Visual Studio 디버그 콘솔
1
9900
D:\₩1_YYR₩C_WorkSpaces₩First₩Debug₩First.exe(프로세스 10996개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

### 3. 간단한 프로그램 실습

#### 05 현재 작업 중인 프로젝트 종료 : [파일] → [솔루션 닫기]



#### 06 [원하는 작업을 선택하세요] 화면이 나오면 오른쪽 위 <x>를 클릭 → [파일] - [끝내기]

### 3. 간단한 프로그램 실습

- 기존의 C 소스 다시 열기(1)

01 Visual Studio 실행 → [프로젝트 또는 솔루션 열기] →

'D:\W1\_YYRWC\_WorkSpaces\First' 폴더의 'First.sln' 파일을 선택

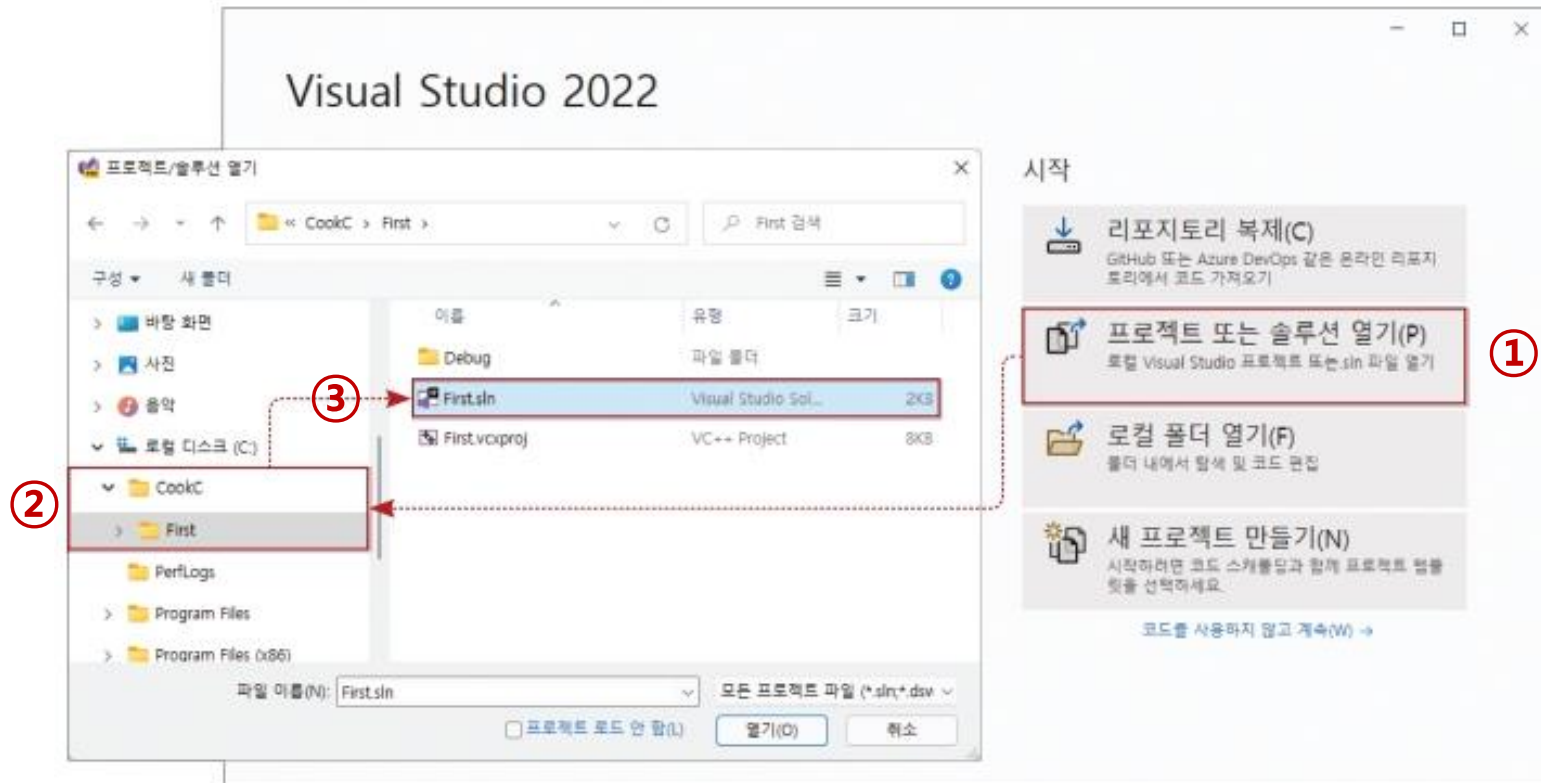


그림 1-35 기존 프로젝트 열기

### 3. 간단한 프로그램 실습

- 02** First.c 파일이 보일 것, 만약 C 소스가 보이지 않는다면 왼쪽 아래의 [솔루션 탐색기] 탭을 클릭한 후 '소스 파일'를 클릭해서 확장하고, F'irst.c'를 더블클릭

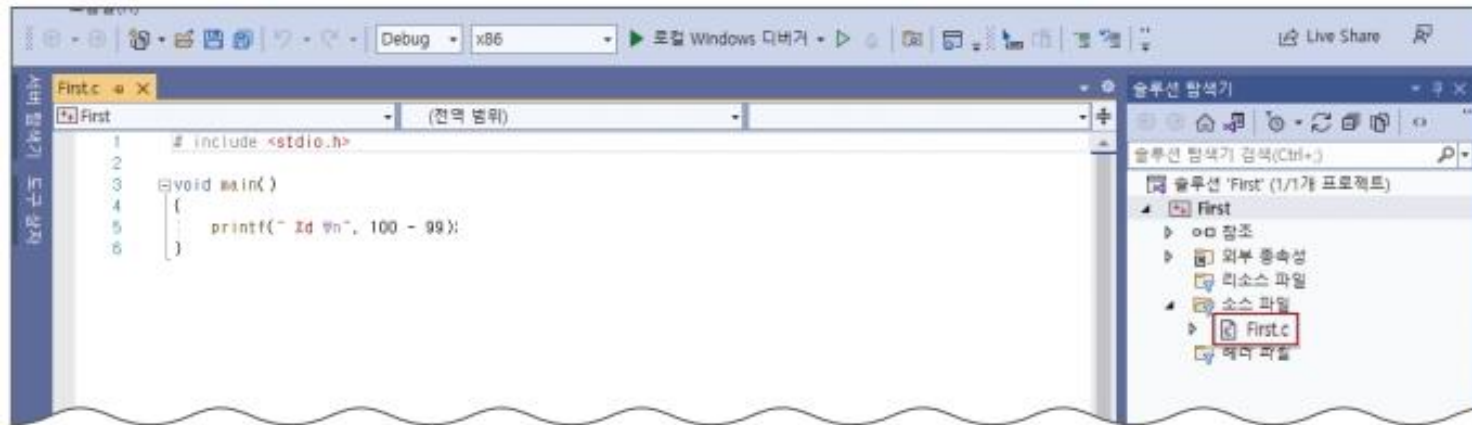


그림 1-36 기존 프로젝트 열기 완료

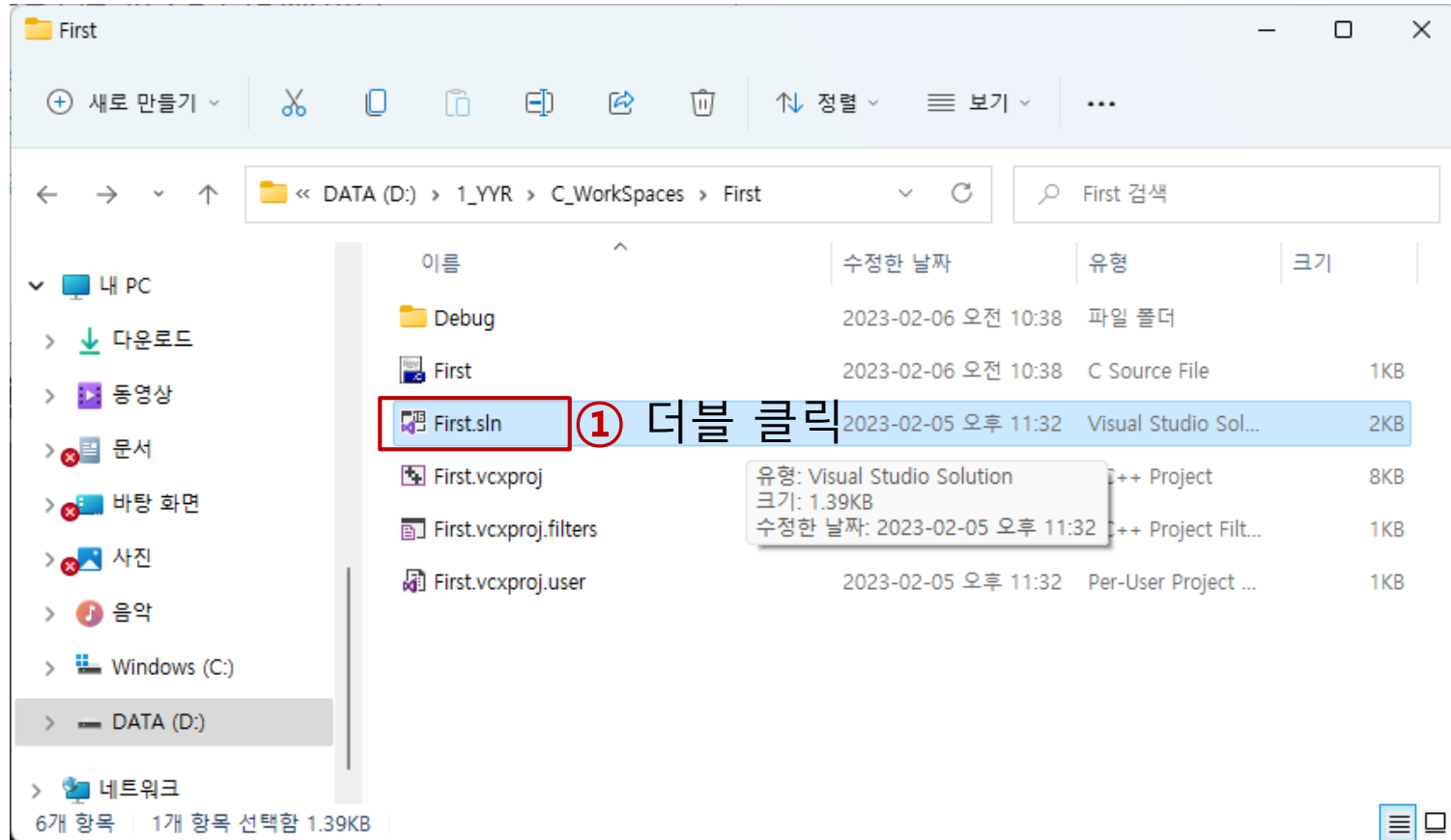
**여기서 잠깐 윈도우 11에서 파일의 확장명이 보이지 않는 경우**

- [파일 탐색기]를 실행하고 [레이아웃 및 보기 옵션] 아이콘을 클릭한 후, [표시] >> [파일 확장명]을 선택
- 참고로 윈도우 10에서는 [파일 탐색기]의 [보기] - [파일 확장명]을 체크

### 3. 간단한 프로그램 실습

- 기존의 C 소스 다시 열기(2)

- 작업 폴더 'D:\W1\_YYR\WC\_WorkSpaces\First' 에서 'First.sln' 파일 더블 클릭





# [기본 1-1] C 프로그램 맛보기

## 1. 프로그램 코딩

```
01 #include<stdio.h>
02
03 int main( )
04 {
05     int result;
06     result = 100 - 50;
07     printf("%d", result);
08 }
```

- 100에서 50을 뺀 결과를 모니터에 출력하는 전형적인 C 프로그램

### ✓ Slef study

- 위의 프로그램을 수정하여 123과 456을 곱한 결과를 출력하는 프로그램을 작성하라.