고급객체지향프로그래밍 강의 보조 자료

이영구 교수님

고급객체지향프로그래밍

Data & Knowledge Engineering Laboratory
Kyung Hee University, Korea

Contents

■ Solution, Project 생성 방법

❖ Visual Studio에서는 솔루션과 프로젝트라는 개념적 컨테이너를 구현하여 IDE(통합 개발 환경)에서 광범위한 도구, 디자이너, 템플릿 및 설정을 적용할 수 있도록 합니다.

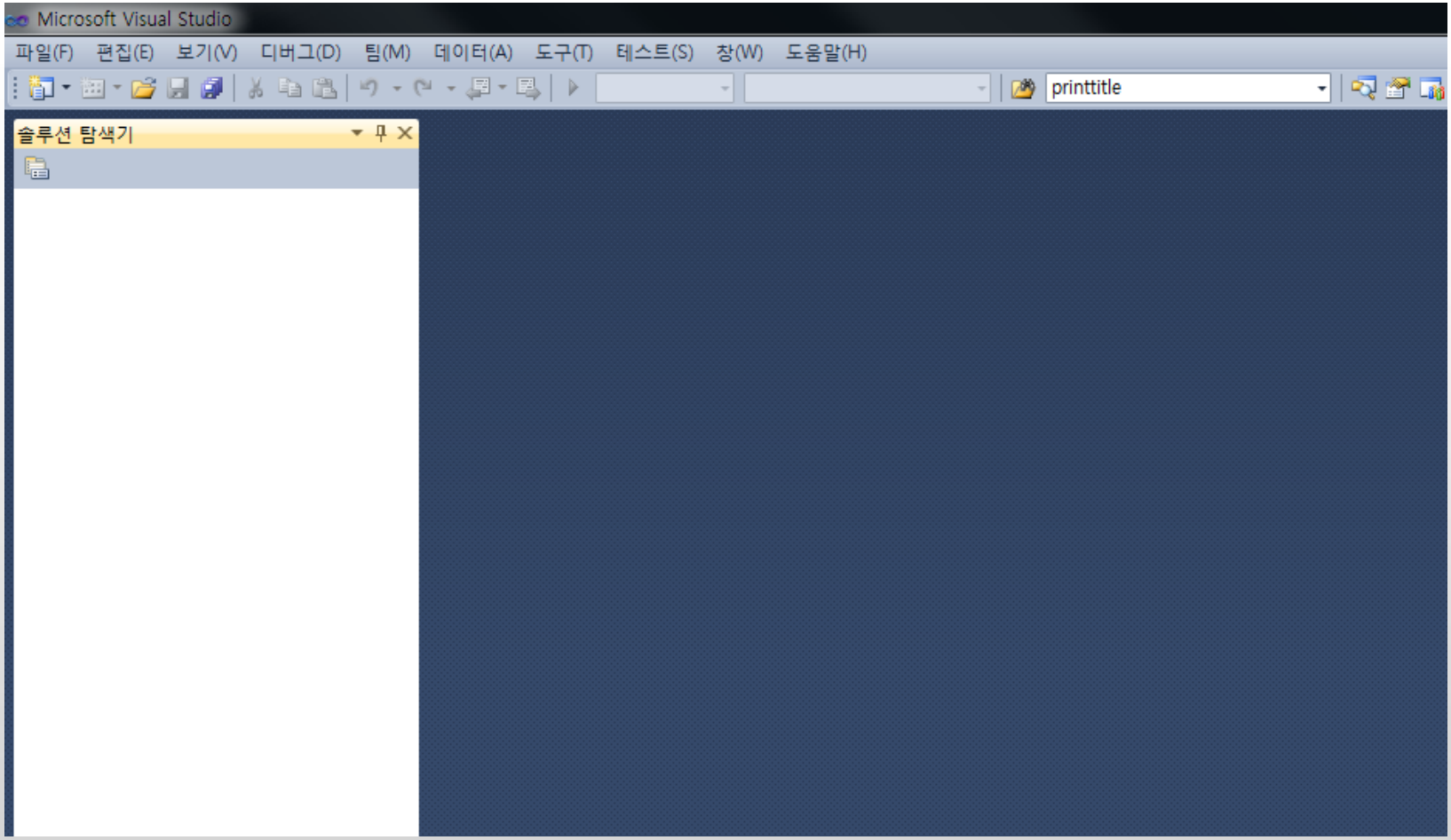
- Solution이란?
  - 솔루션은 **관련 프로젝트들의 그룹**으로 구성되어 폴더 형태로 제공이되며, 해당 프로젝트 그룹에서 작업을 수행할 수 있습니다. 솔루션에는 전체적으로 정의할 수 있는 메타데이터와 파일 및 하나 이상의 프로젝트가 포함됩니다.
- Project란?
  - **프로젝트**에는 **소스 파일 집합과 관련 메타데이터**(예: 구성 요소 참조, 빌드 지침)가 포함되고, 프로젝트를 빌드하면 일반적으로 하나 이상의 출력 파일이 만들어집니다.

[출처] http://msdn.microsoft.com/ko-kr/library/df8st53z.aspx

1-1. Solution, Project 생성 방법

■ Solution, Project 생성 방법

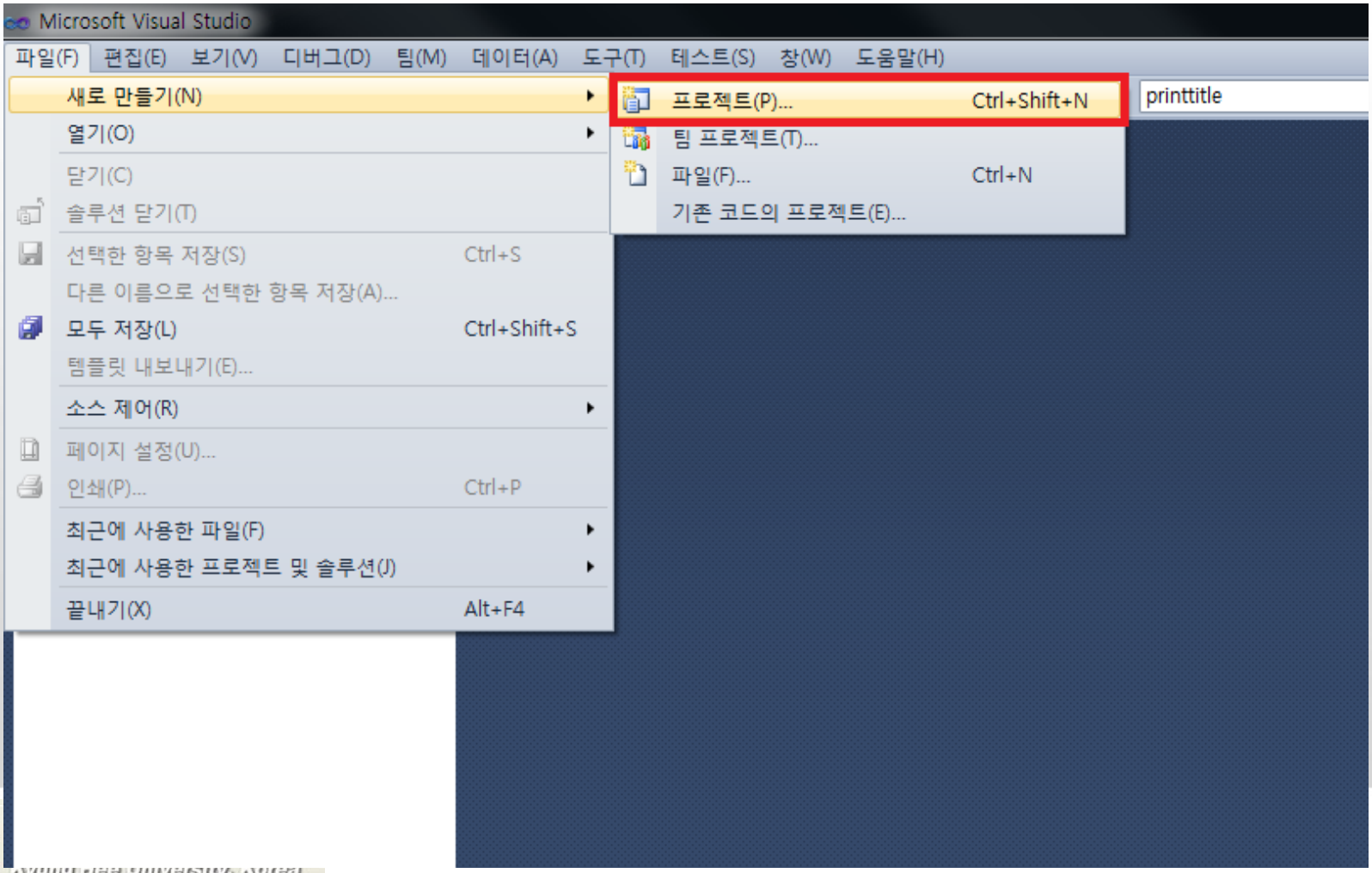
❖ Visual Studio 2010을 실행합니다.



1-1. Solution, Project 생성 방법

■ Solution, Project 생성 방법

❖ 프로젝트를 생성하기 위해
[파일]->[새로 만들기]->[프로젝트] 순으로 클릭합니다.

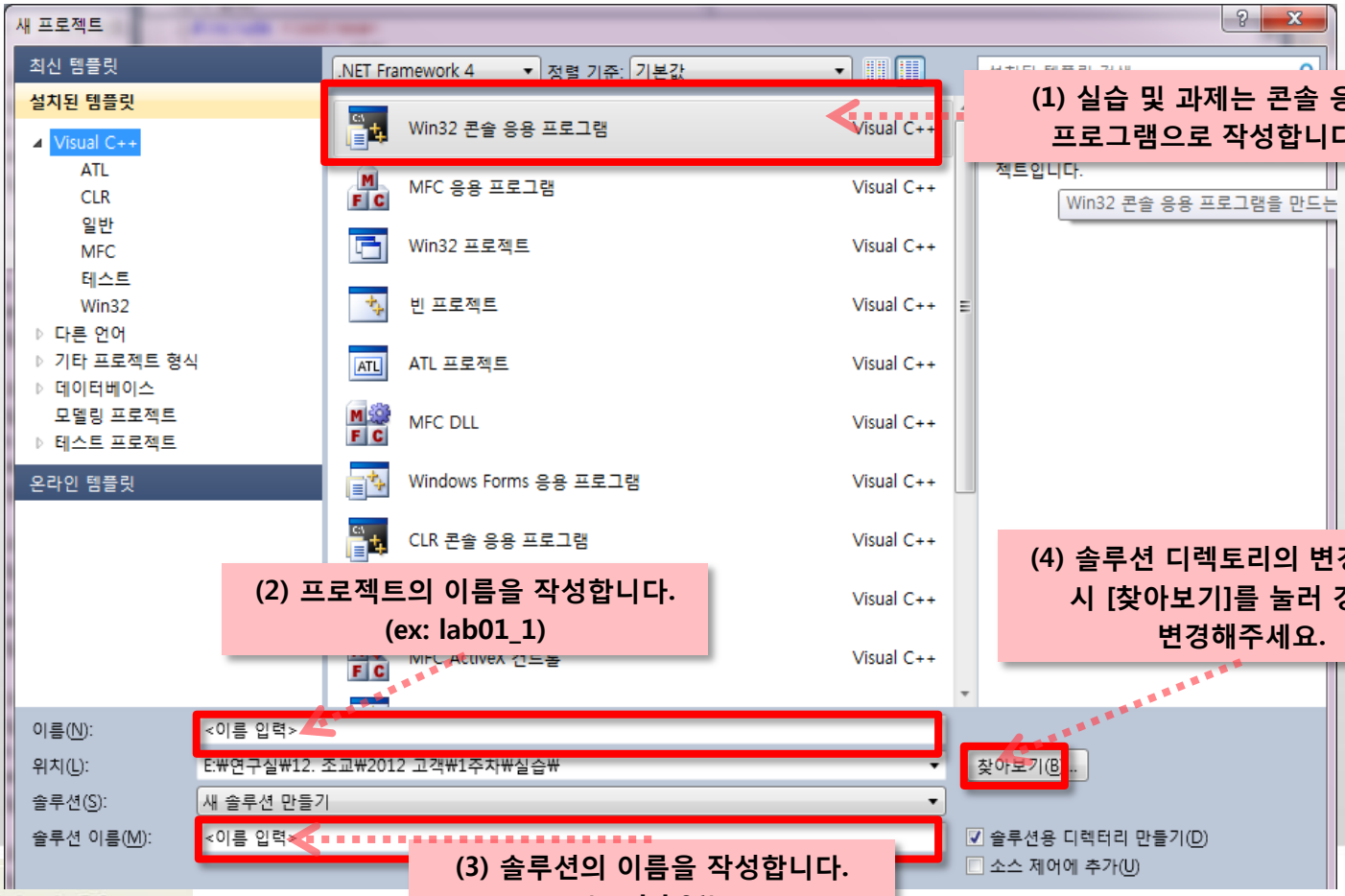


1-1. Solution, Project 생성 방법

■ Solution, Project 생성 방법

❖ '새 프로젝트' 만들기

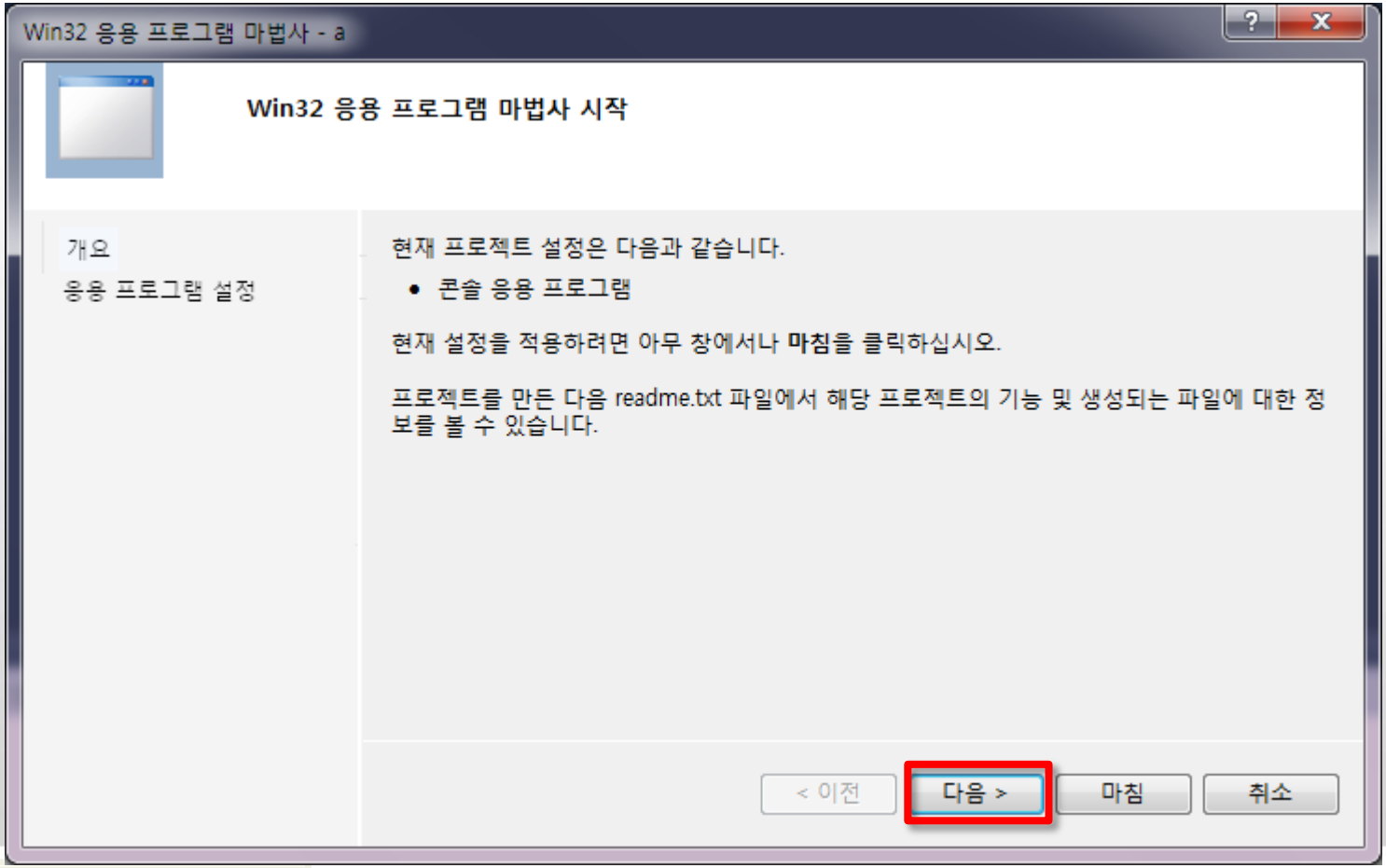
- 콘솔 응용프로그램이란? 명령 프롬프트와 같은 콘솔 창을 기반으로 작동하는 프로그램



1-1. Solution, Project 생성 방법

■ Solution, Project 생성 방법

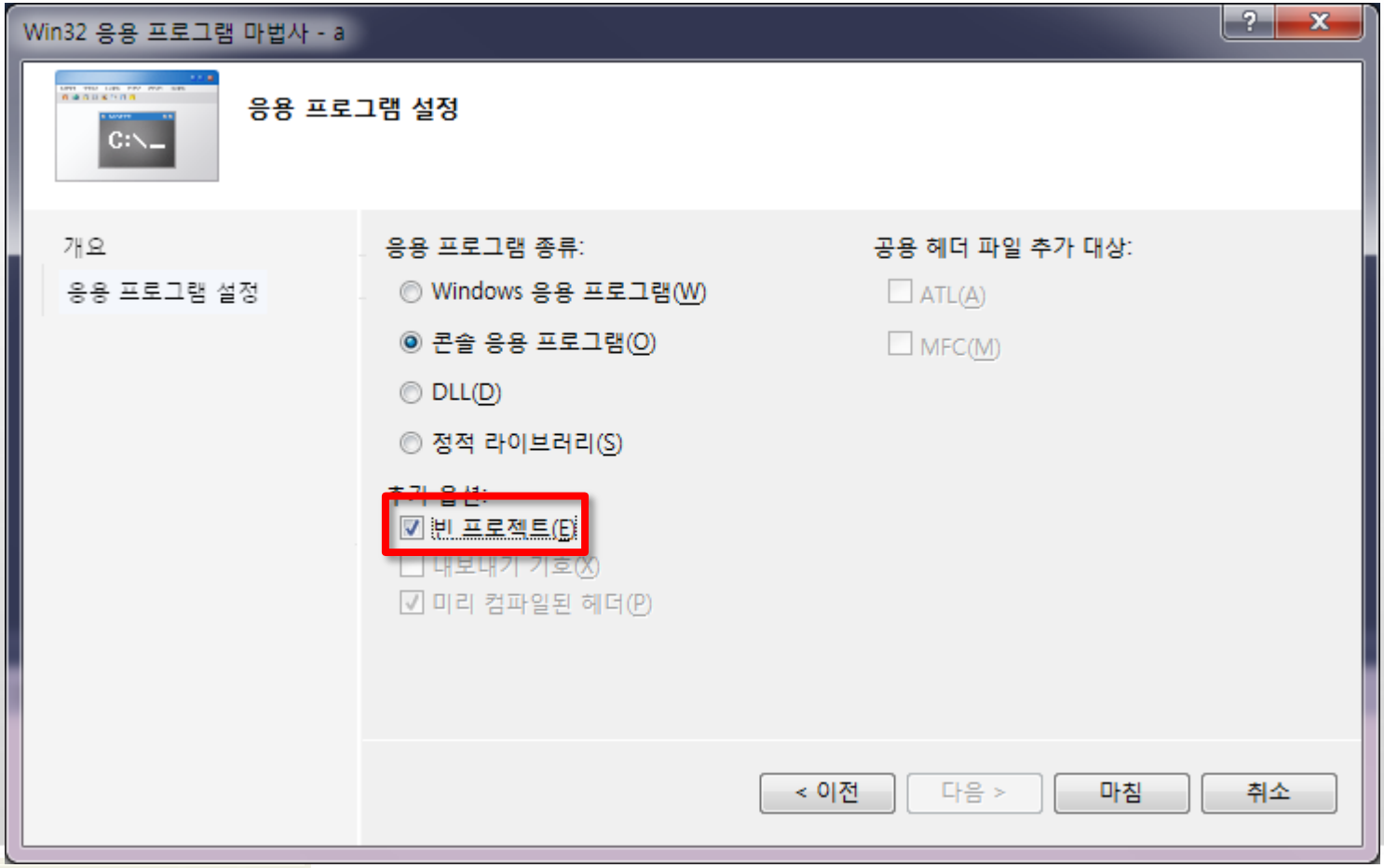
❖ 'Win32 응용 프로그램 마법사 – 솔루션이름' 창이 뜹니다.
[다음] 버튼을 누르세요.



1-1. Solution, Project 생성 방법

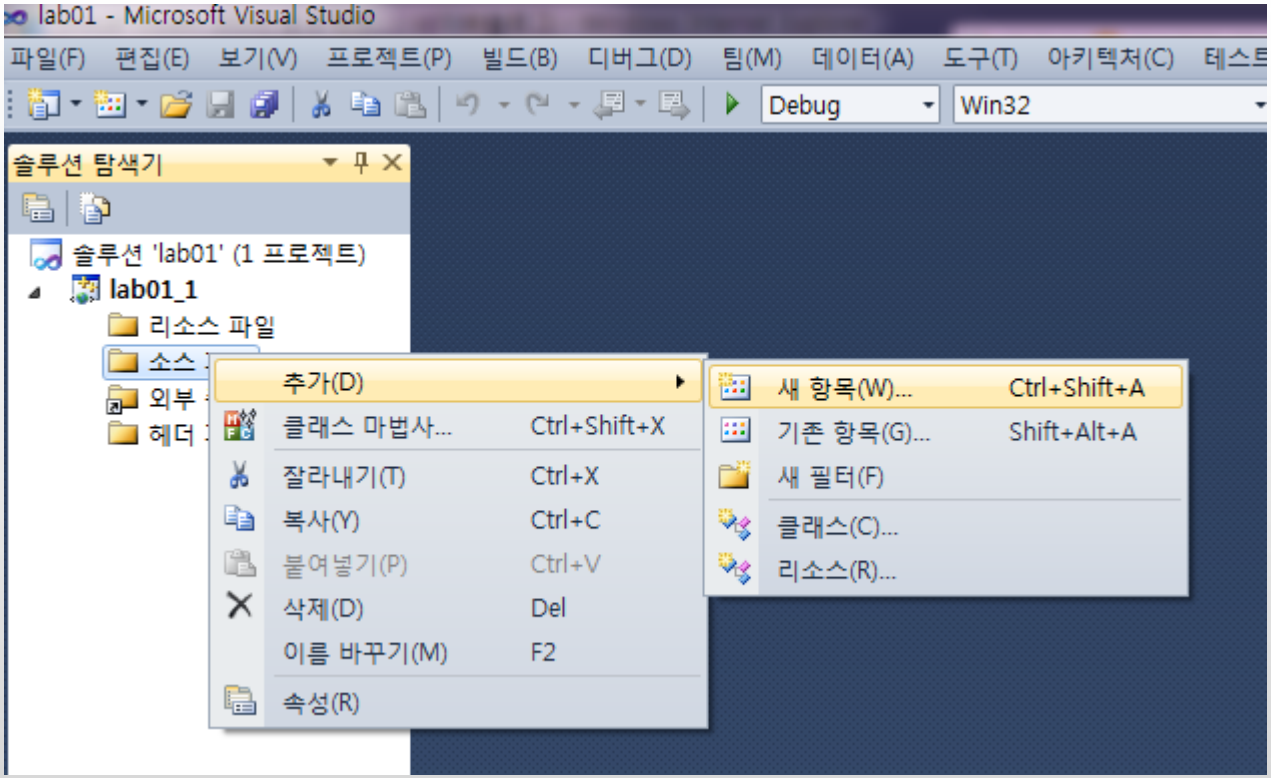
■ Solution, Project 생성 방법

❖ 추가 옵션 중, 빈 프로젝트 옵션에 체크해주세요.
그 후 마침을 누르면 프로젝트가 생성됩니다.



■ Solution, Project 생성 방법

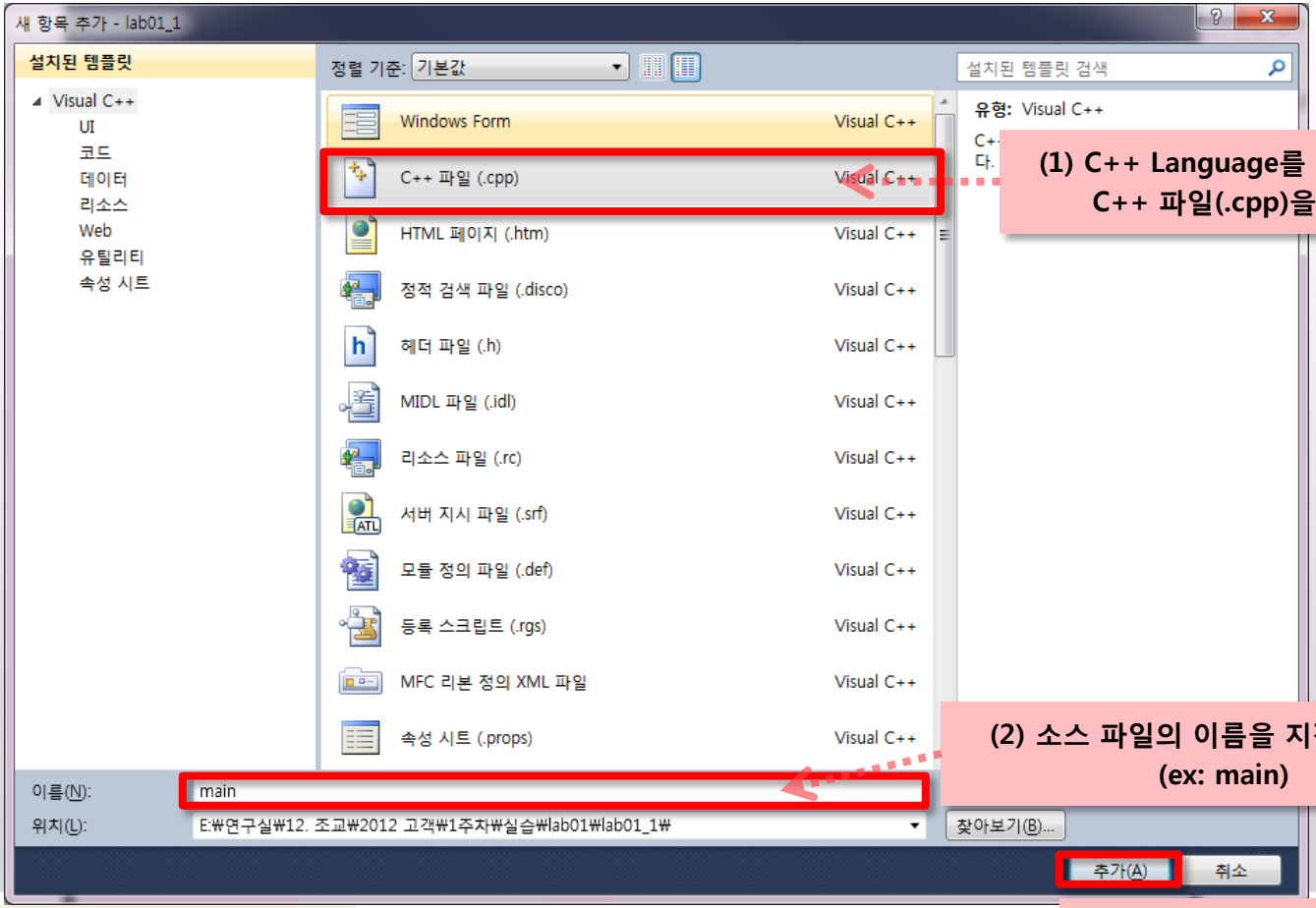
❖ 현재 프로젝트만 생성된 상태이므로, 소스코드를 추가하기 위해 [소스 파일]->[추가]->[새 항목]을 클릭합니다.



1-1. Solution, Project 생성 방법

■ Solution, Project 생성 방법

❖ 현재 프로젝트만 생성된 상태이므로, 소스코드를 추가하기 위해 [소스 파일]->[추가]->[새 항목]을 클릭합니다.



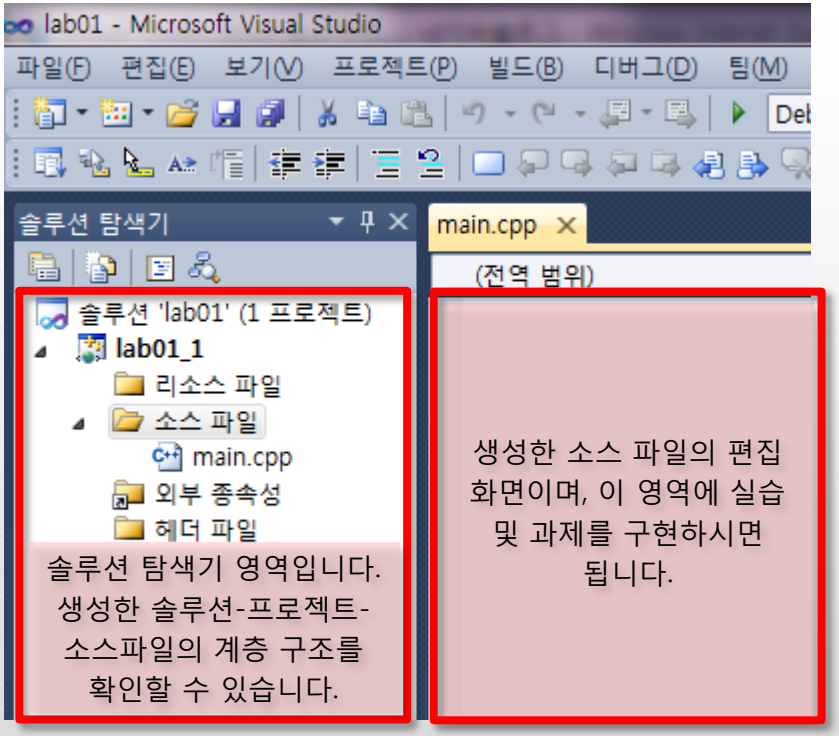
(1) C++ Language를 사용할 것이므로, C++ 파일(.cpp)을 선택합니다.

(2) 소스 파일의 이름을 지정해줍니다. (ex: main)

(3) 추가 버튼을 클릭합니다.

■ Solution, Project 생성 방법

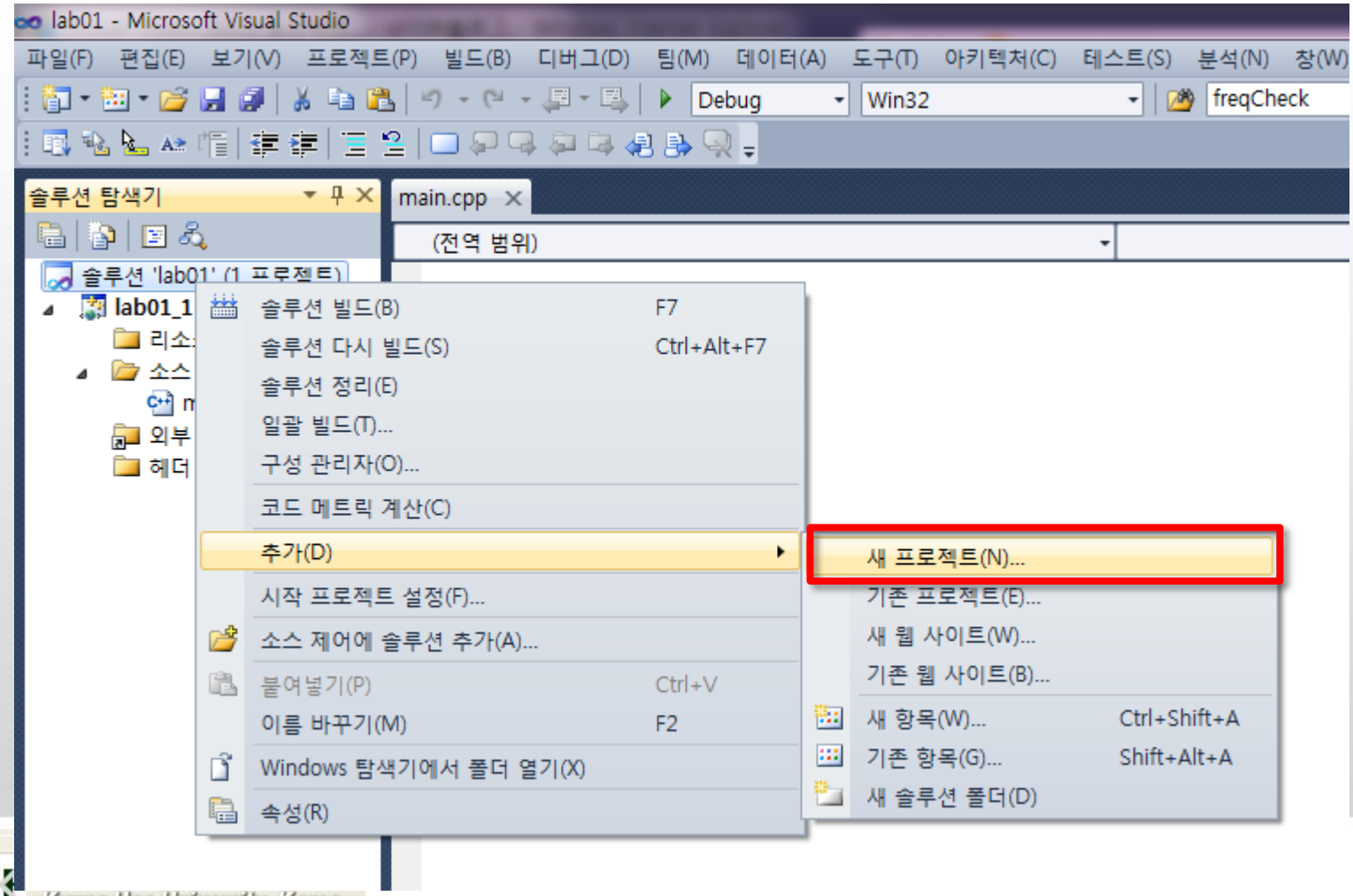
❖ 솔루션 과 프로젝트, 소스 파일까지 생성 완료.



1-1. Solution, Project 생성 방법

■ 이미 생성된 Solution 내에 새 프로젝트 생성하기

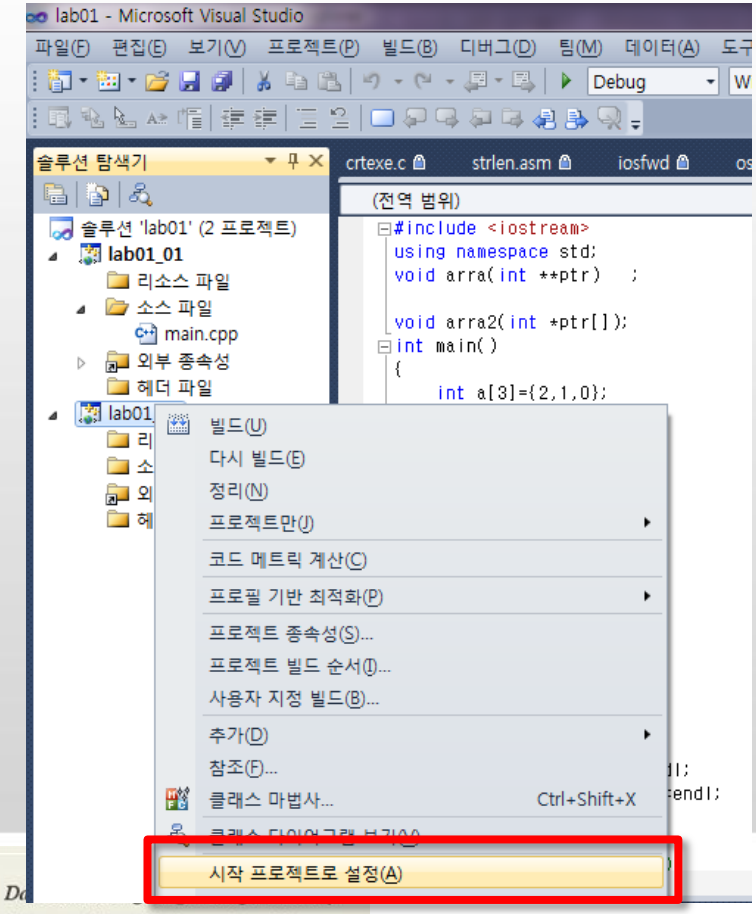
❖ 솔루션 'lab01'을 클릭한 후 마우스 우버튼 클릭->[추가]->[새 프로젝트] 순으로 클릭한 후, 5page를 참조하여 동일하게 작성합니다.



1-1. Solution, Project 생성 방법

■ 여러 프로젝트 중 실행할 프로젝트로 활성화 시키는 방법

- ❖ 실행하고자 하는 프로젝트의 이름에 마우스 오른쪽 버튼을 클릭
- ❖ [시작 프로젝트로 설정(A)]을 클릭하면 프로젝트명이 진한 글씨체로 변경되면서, 빌드 시 해당 프로젝트가 실행된다.



1-2. 빌드 & 실행 방법

■ 컴파일, 링크, 빌드 란?

❖ 컴파일(compile) 이란?

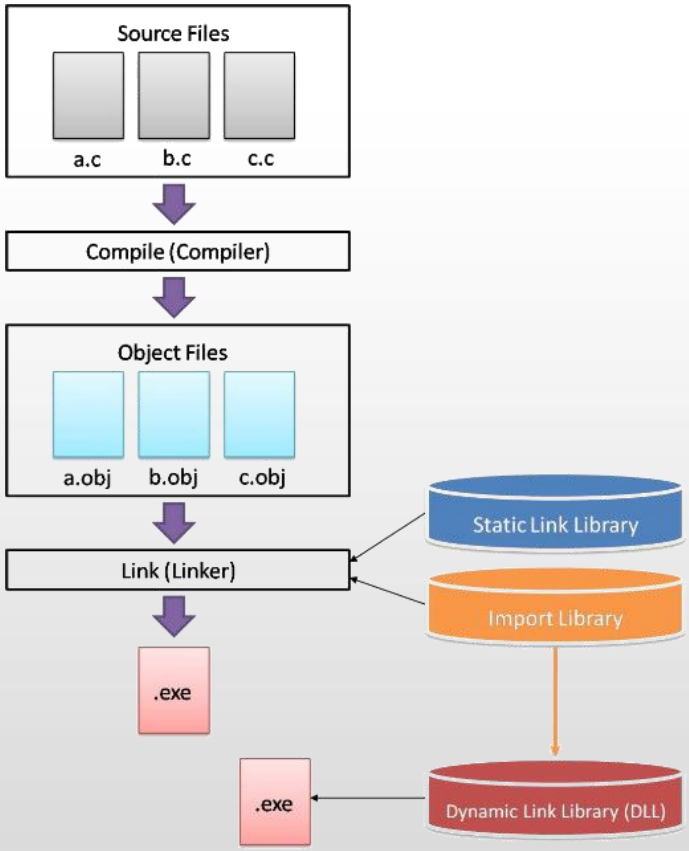
- 특정 프로그래밍 언어(C, C++등..)를 사용하여 컴퓨터가 이해할 수 있는 기계어 코드(이진코드)로 변환하는 작업

❖ 링크(link) 란?

- 대부분의 애플리케이션들은 반복적인 코드(기능) 등을 라이브러리화하여 사용한다.
- 이러한 라이브러리들을 구현한 애플리케이션과 연결해주는 작업이 링크이다.

❖ **빌드(build) 란?**

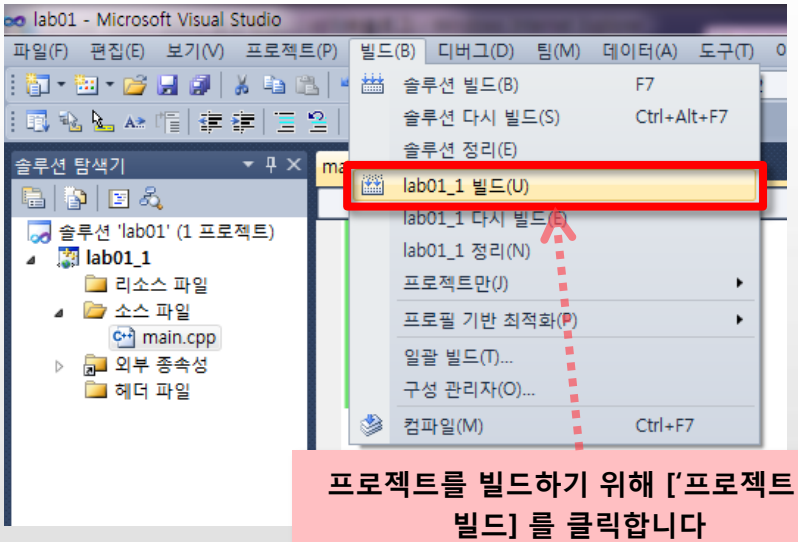
- 컴파일+링크
- 컴퓨터에서 실행 가능한 형태의 기계어 코드 출력



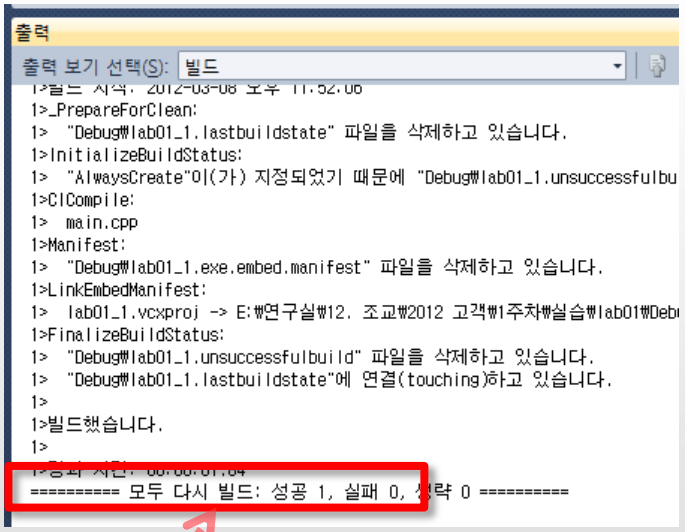
1-2. 빌드 & 실행 방법

■ 빌드 방법

- ❖ [빌드]->['프로젝트명' 빌드] 를 클릭합니다
- ❖ 빌드 단축키 F7



프로젝트를 빌드하기 위해 ['프로젝트명' 빌드] 를 클릭합니다

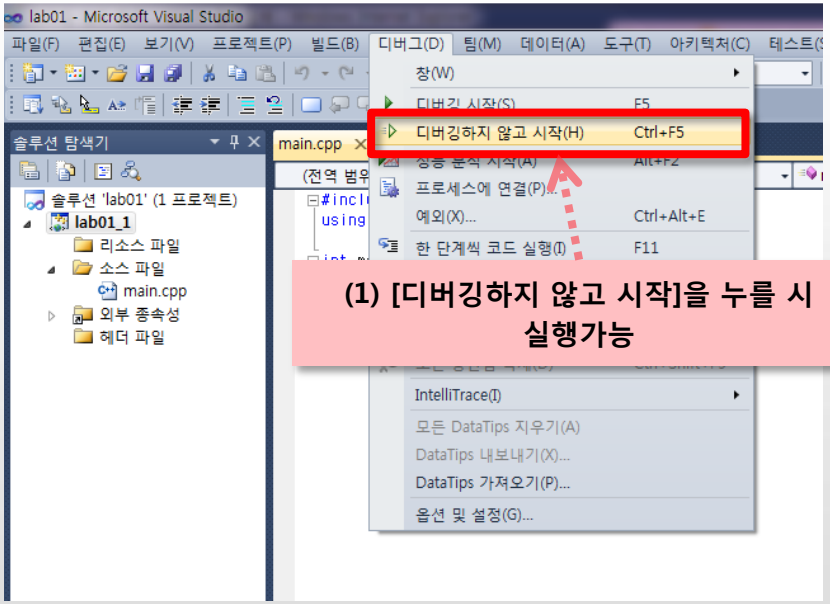


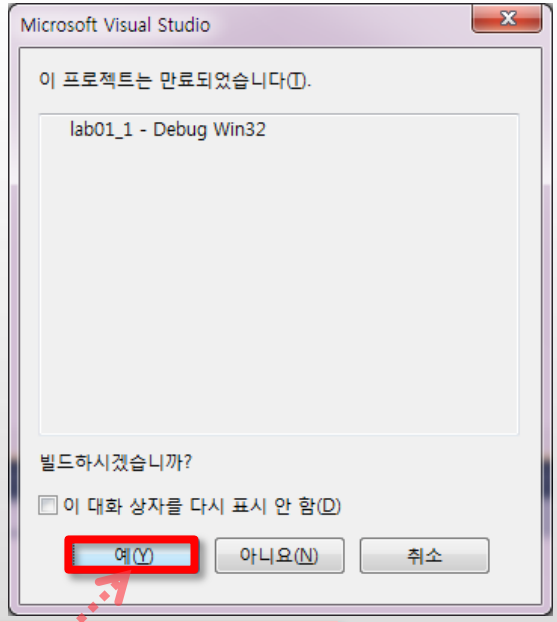
빌드 성공

1-2. 빌드 & 실행 방법

■ 실행 방법

- ❖ [디버그]->[디버깅하지 않고 시작]을 클릭합니다.
- ❖ 단축키 : Ctrl + F5



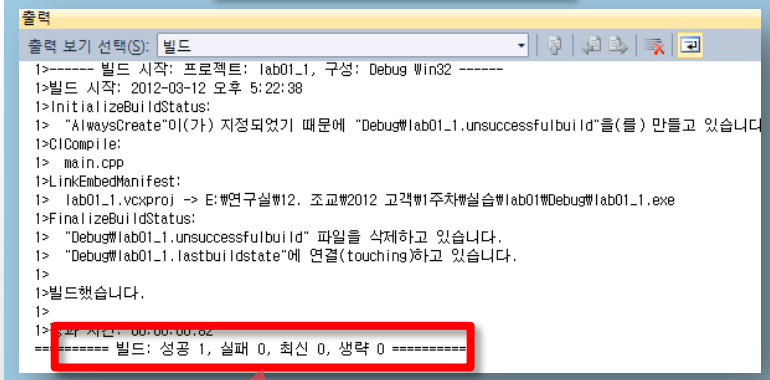


1-3. 빌드(컴파일) 시 에러 메시지로 이동, 확인하는 방법

■ 실행 방법

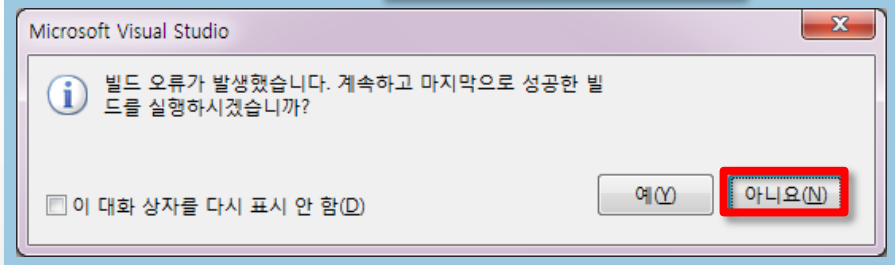
❖ 실행 성공과 실패의 예

실행 성공

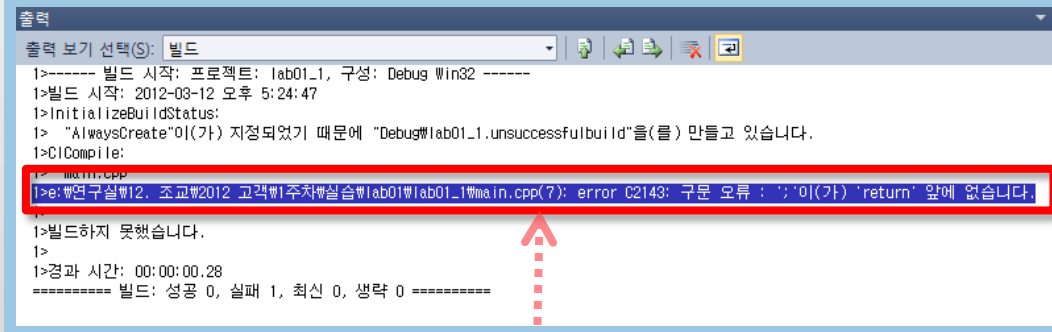


빌드 성공

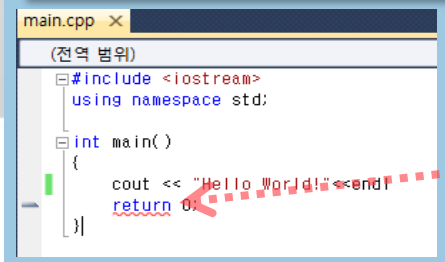
실행 실패



오류가 발생하여 빌드를 실패하였으므로, '아니요(N)' 버튼을 클릭



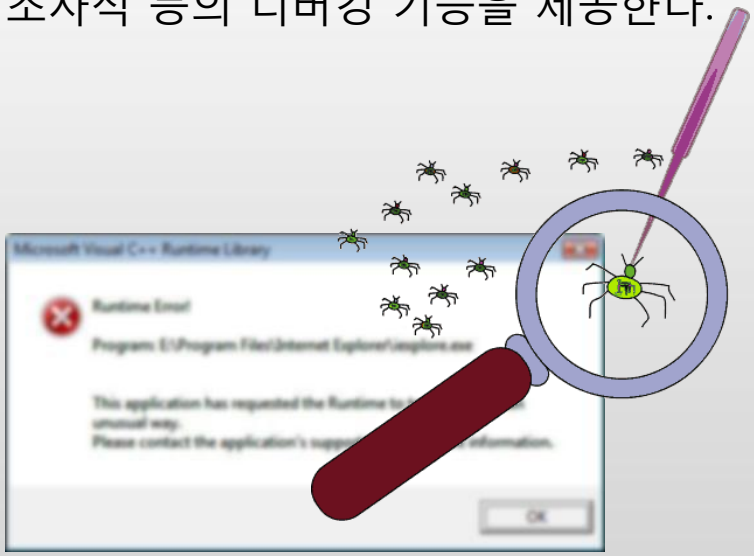
다음 에러 메시지로 이동(단축키 F4)를 이용하여 어떤 에러인지 확인 후 디버깅



수정

디버깅이란?

- 컴퓨터 프로그램에서 잘못된 부분, 즉 버그를 찾아서 수정하거나 또는 에러를 피해나가는 처리과정이다.
- 디버깅 도구를 사용하면 각 개발단계에서의 잘못된 코딩부분을 쉽게 찾아낼 수 있으며, 몇몇 프로그램 개발 패키지에는 프로그램 작성시 그때그때 잘못된 부분을 검사할 수 있는 기능이 포함되어 있다.
  - Ex) Visual studio는 디버거를 자체 내장하고 있는 통합 개발 환경이다. 중단점, 프로시저 단위로 실행, 조사식 등의 디버깅 기능을 제공한다.



[출처] http://www.wpclipart.com/computer/humour/debugging.png.html

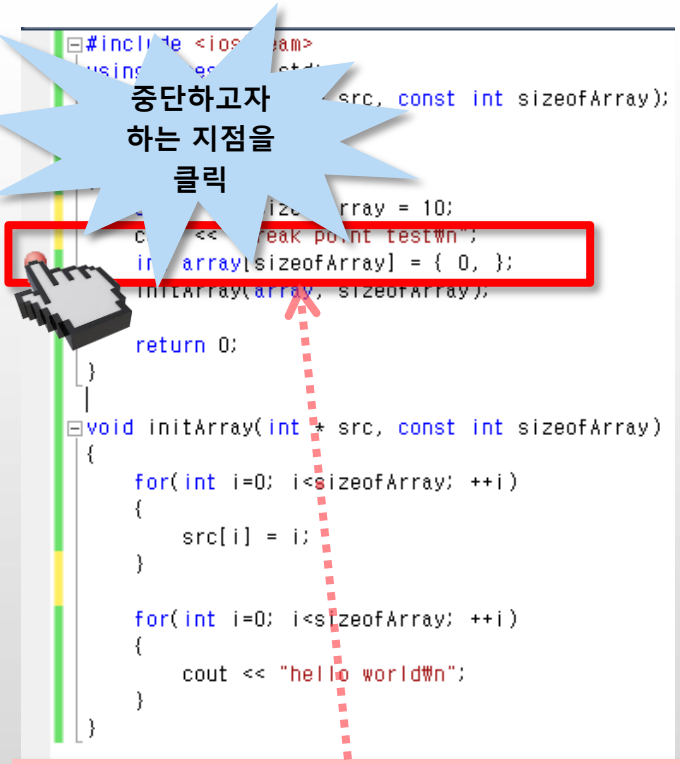
■ 디버깅 방법

❖ 중단점(breakpoint)으로 디버깅하기

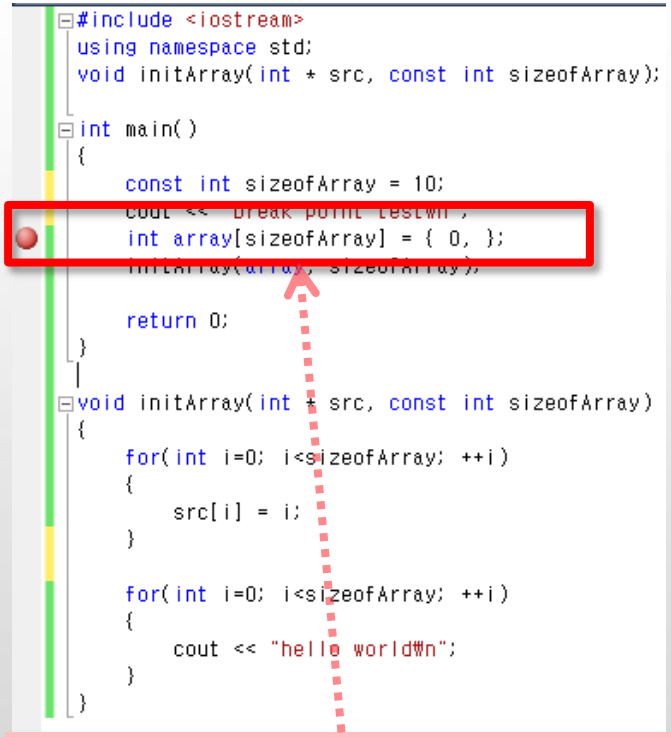
- 중단점이란?
  - 중단점이란 프로그램의 실행을 중지하고자 하는 지점을 말한다.
  - 프로그램의 동작을 관찰하려면 일단 실행을 멈추어야 하므로 중단점을 설정(단축키 F9)하는 일은 디버깅 작업의 시작이라 할 수 있다.
  - 중단점을 설정해 놓고 디버깅을 시작하면, 중단점에서 프로그램의 실행이 잠시 중단된다. 이 상태에서 단계 실행(단축키 F10, F11) 및 변수의 상태 확인, 변경을 할 수 있다.
  - 에러가 의심되는 부분에 중단점을 설정하고, 단계 실행을 통해 변수의 값이나 제어 구조의 흐름을 따라가면서 어디가 잘못되었는지 관찰할 수 있다.

■ 디버깅 방법

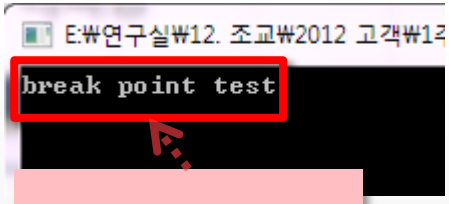
❖ 중단점(breakpoint) 설정 및 실행



(1) 중단시키고 싶은 지점을 클릭(혹은 단축키 F9)



(2) 디버깅 시작(F5)시 해당 지점에서 수행을 멈춤.

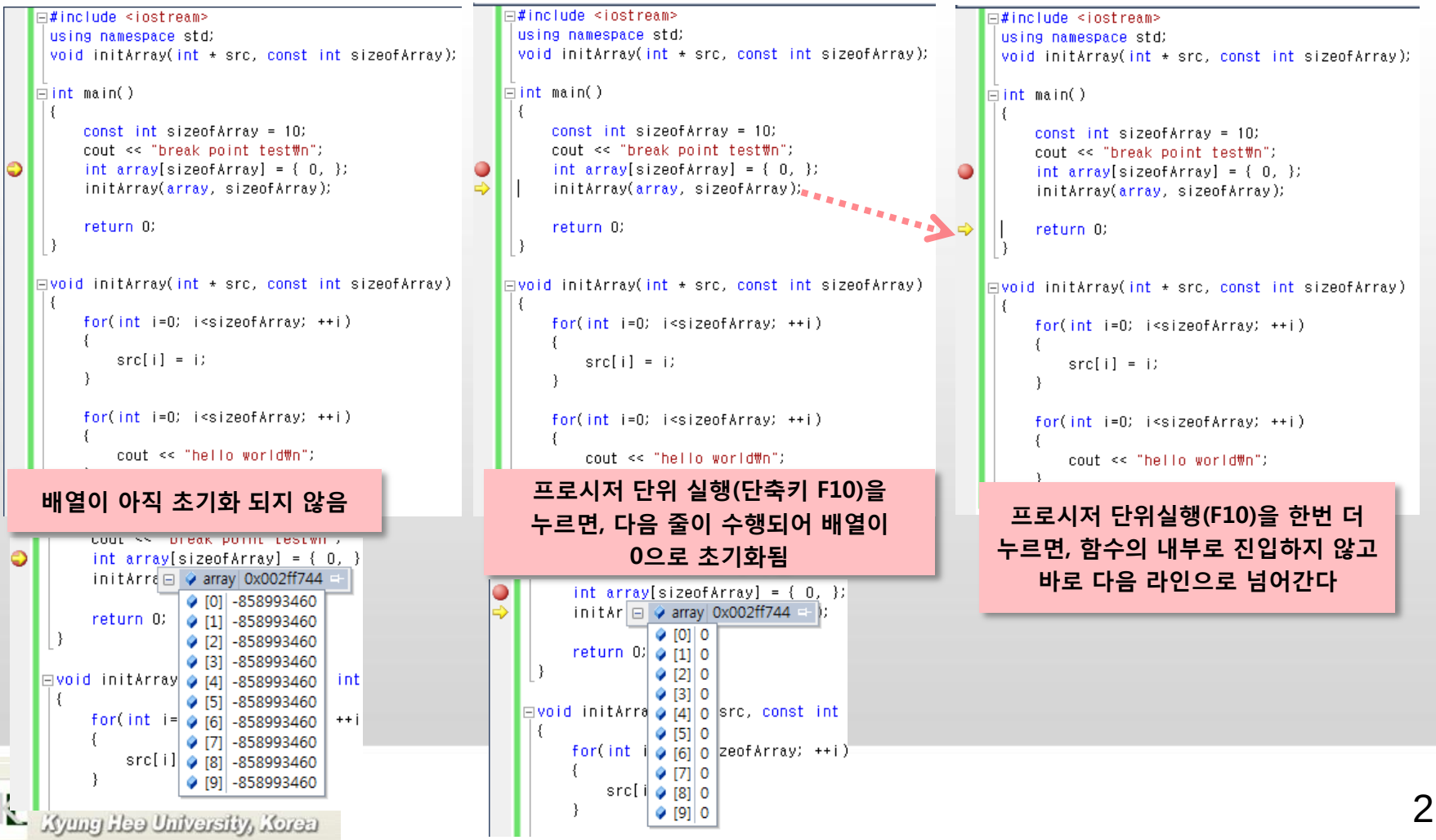


(3) 중단점까지 수행된 결과를 확인할 수 있음

■ 디버깅 방법

❖ 중단점 이후, 단계 실행(단축키 F10, F11, shift+F11)을 이용하여 디버깅

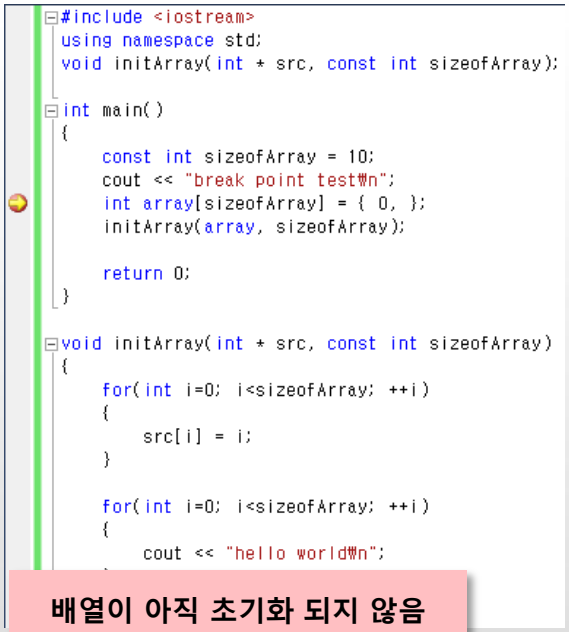
- 프로시저 단위 실행(F10)



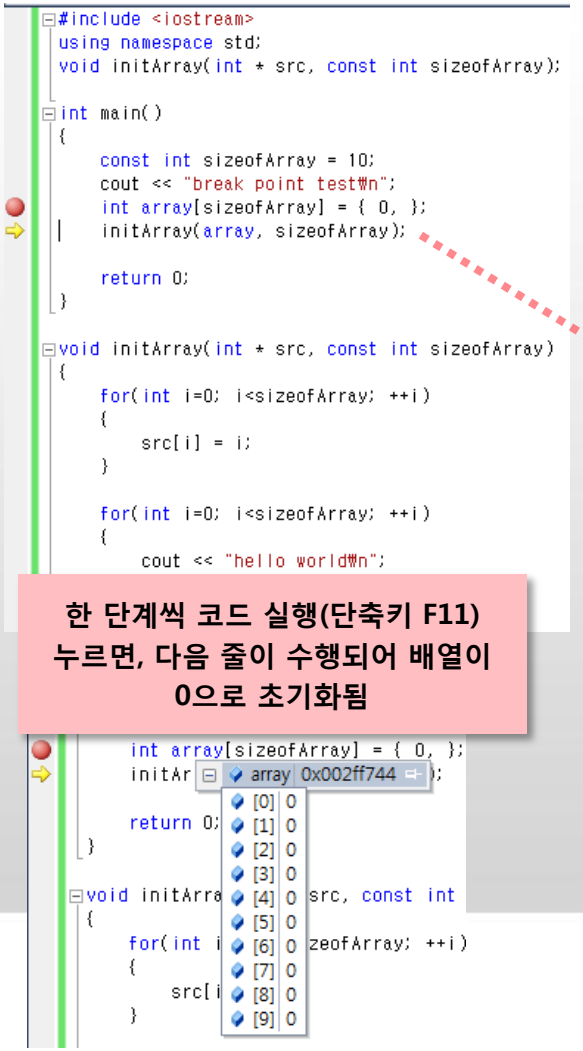
■ 디버깅 방법

❖ 중단점 이후, 단계 실행(단축키 F10, F11, shift+F11)을 이용하여 디버깅

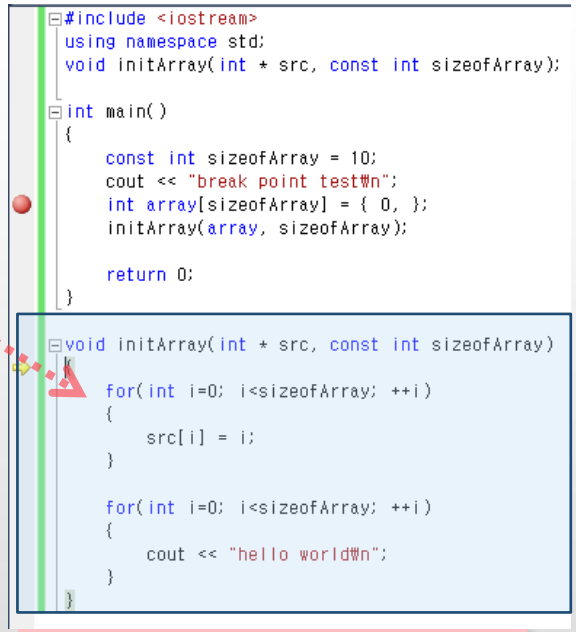
- 한 단계씩 코드 실행(F11)



배열이 아직 초기화 되지 않음



한 단계씩 코드 실행(단축키 F11) 누르면, 다음 줄이 수행되어 배열이 0으로 초기화됨

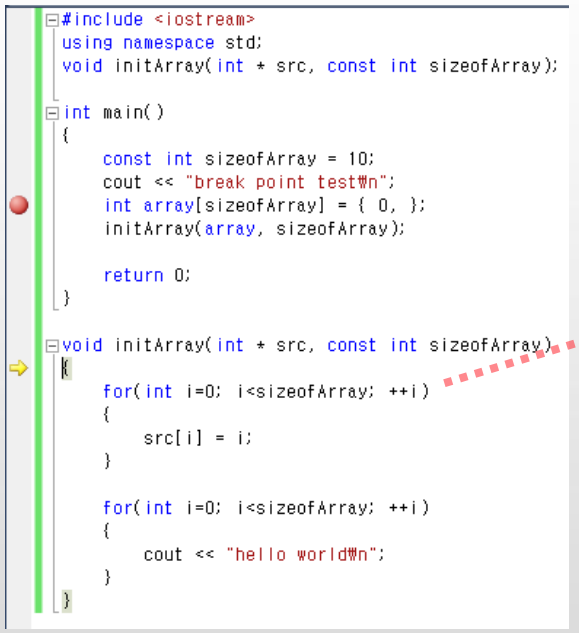


한 단계씩 코드 실행(단축키 F11) 수행 시, 함수의 내부로 진입하여 한 줄씩 수행한다.

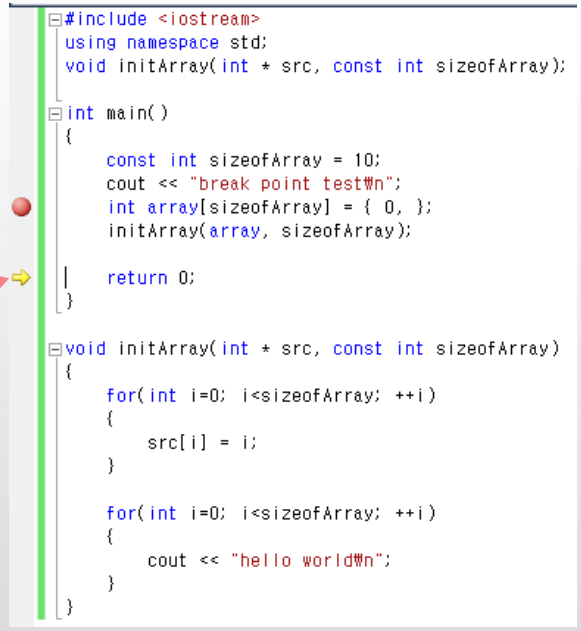
■ 디버깅 방법

❖ 중단점 이후, 단계 실행(단축키 F10, F11, shift+F11)을 이용하여 디버깅

- 프로시저로 진입하였을 때, 해당 함수를 빠져나가고 싶다면 프로시저 벗어나기(단축키 shift+F11)를 활용



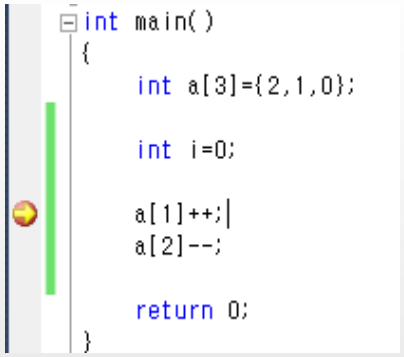
F11 수행 시, 함수의 내부로 진입하여 한 줄씩 수행한다.



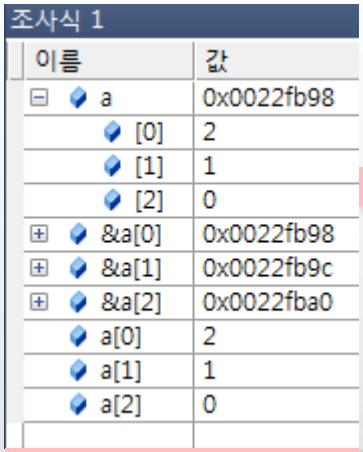
Shift+F11 수행 시, 바로 프로시저의 바깥으로 빠져나간다.

■ 조사식 사용법

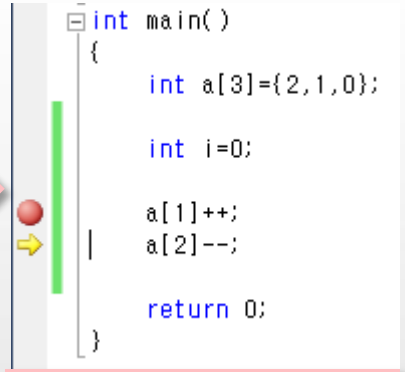
❖ 디버그 모드에서 해당 지점까지 수행되었을 때 변수 값, 변수의 메모리 번지 등을 확인하기 위해 조사식을 활용한다.



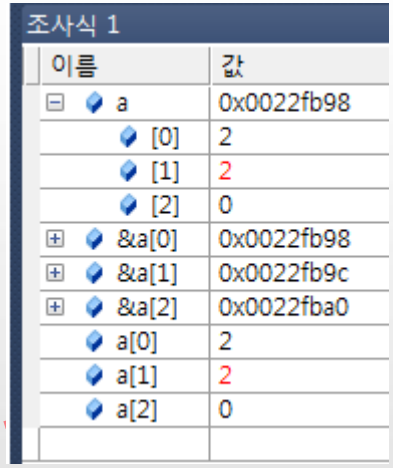
중단점을 설정하여 빌드 후 실행



작업창 하단의 조사식의 왼쪽 이름에 확인하고자 하는 변수의 이름 등을 적어 값을 확인 (수행 시점에서의 값을 확인할 수 있다)



단계실행을 통해 한 줄을 수행한다.



단계실행에 의해 값이 바뀐 경우, 값의 색이 빨간색으로 변경된다. a[1]++;줄이 수행되어 조사식에서 a[1]의 값이 변경됨.

1-5. 디버깅 모드와 Release 모드

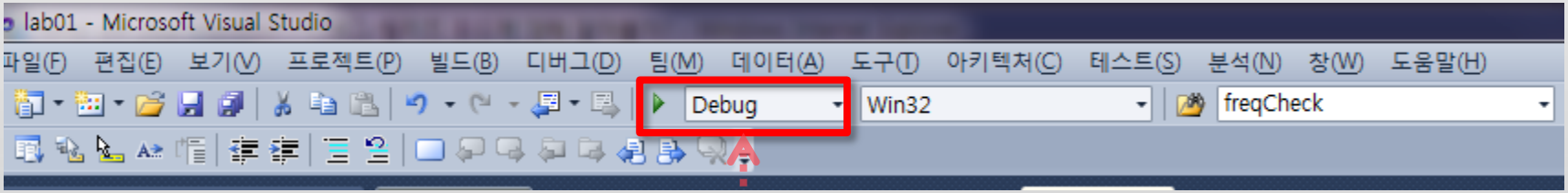
■ 디버그 모드 vs 릴리즈 모드?

❖ 디버그 모드

- 실행 파일에 디버깅 정보를 삽입하여 언제든지 디버깅을 할 수 있도록 하며, 프로젝트 디렉토리 아래에 Debug 서브 디렉토리에 실행 파일을 만들어 준다. 디버깅 정보가 들어가 있기 때문에 실행 파일의 상태를 확인할 수 있다.
  - [출처] http://jongkok4.net/10

❖ 릴리즈 모드

- 일체의 디버깅 정보를 삽입하지 않을 뿐만 아니라 코드를 최적화하여 실행 파일 크기를 최대한 줄여준다. 릴리즈 모드로 만든 실행 파일로는 디버깅을 할 수 없지만 속도나 크기면에서는 디버그 모드로 만든 실행 파일보다 월등히 유리하다. 모든 버그를 잡고 개발이 완료되었을 때는 릴리즈 모드로 컴파일해야 한다.
  - [출처] http://jongkok4.net/10



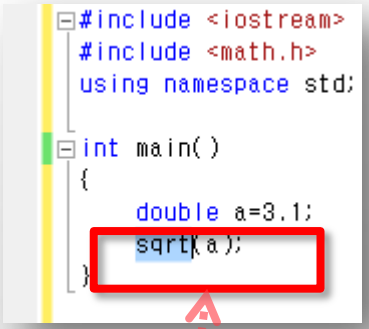
이 부분을 조작하여 Debug 모드와 Release 모드를 변경할 수 있다.

1-6. 함수들에 대한 reference 보는 방법

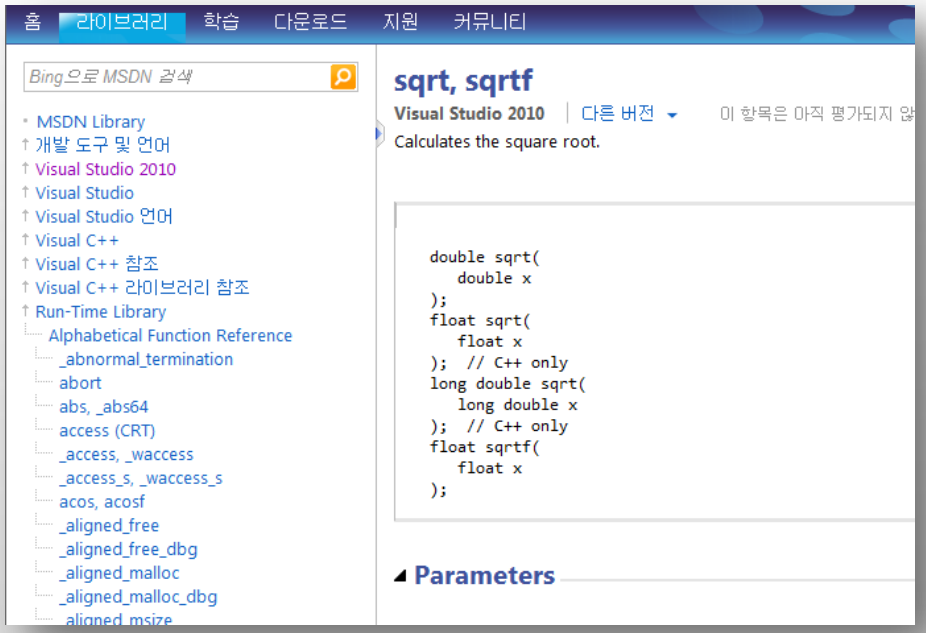
■ Reference 보는 방법

❖ 함수의 기능, 인자, 반환형, 사용법 등의 정보를 알고 싶을 때 msdn을 활용.

- 도움말(F1)



(1) 레퍼런스를 확인하고자 하는 함수명을 블락지정합니다.

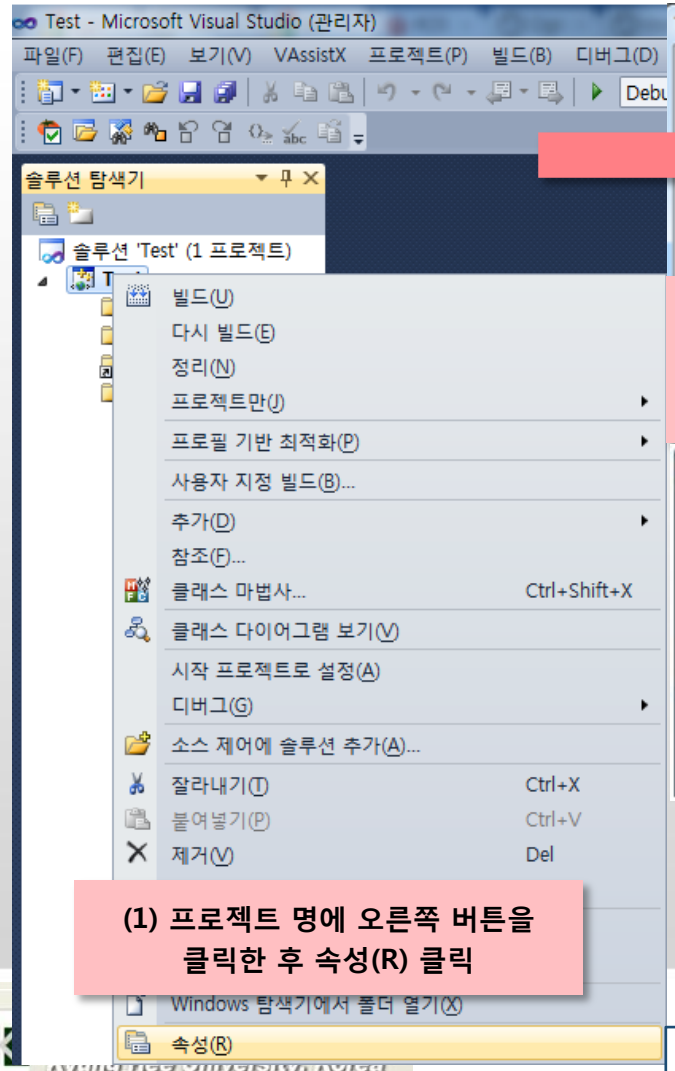


(2) F1을 누르면 해당 함수에 대한 레퍼런스가 열립니다. (네트워크 연결 필요)

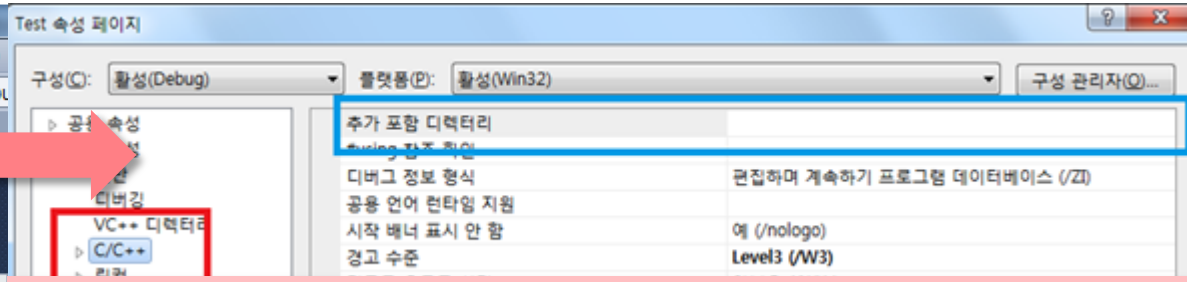
1-7. 외부 라이브러리 설정 방법

■ 빌드 전 필요한 설정

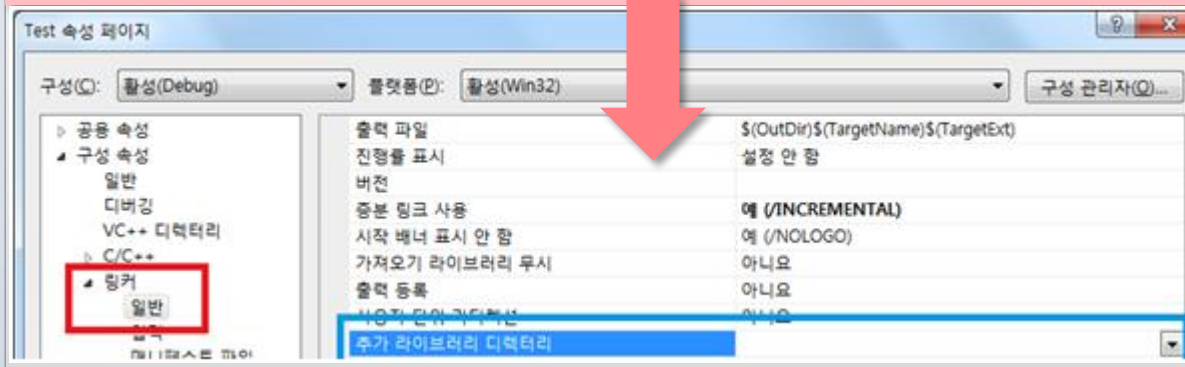
❖ 프로젝트에 라이브러리를 설정하는 방법



(1) 프로젝트 명에 오른쪽 버튼을 클릭한 후 속성(R) 클릭



(2) 속성 페이지에서 그림과 같이 빨간색으로 표시된 구성 속성의 C/C++ 메뉴를 선택 파란색으로 표시된 추가 포함 디렉터리에 .h파일(헤더파일)의 경로를 추가하고, 여러 개를 추가할 경우 세미콜론(;)으로 구분합니다.



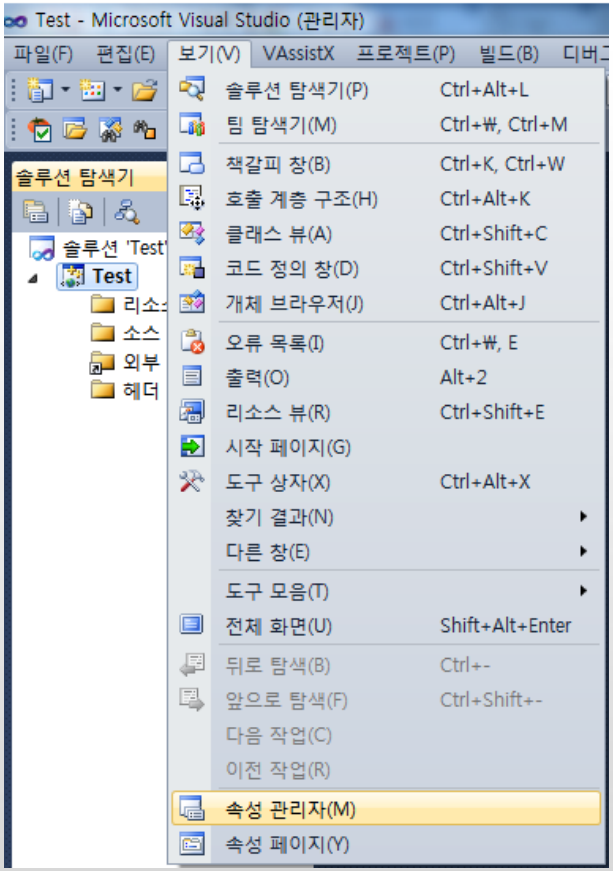
(3) 빨간색으로 표시된 구성속성->링커->일반메뉴에서 파란색으로 표시된 추가 라이브러리 디렉터리에 Include 폴더와 같이 추가해줍니다.

[출처] http://blog.secmem.org/69

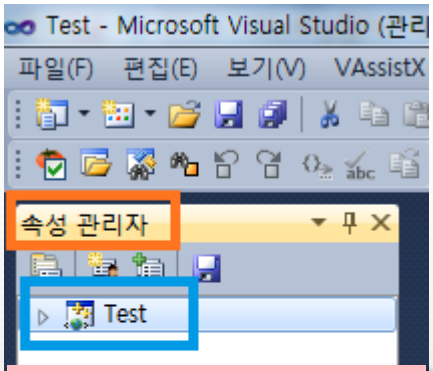
1-7. 외부 라이브러리 설정 방법

■ 빌드 전 필요한 설정

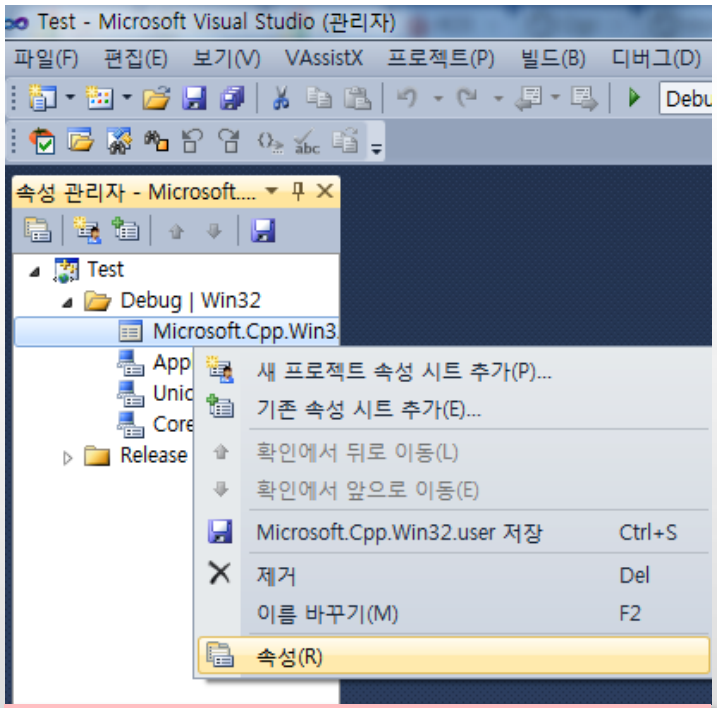
❖ 모든 프로젝트에 공통된 라이브러리를 설정하는 방법



(1) 보기(V) 메뉴의 속성 관리자(M) 선택



(2) 속성 관리자 탭이 나타나게 된다. 파란색으로 표시된 프로젝트를 선택하며, 왼쪽의 작은 삼각형을 클릭하여 Debug|Win32 폴더를 선택후 (디버그 모드의 경우) Microsoft.Cpp.Win32를 선택하고 우클릭하여 속성 메뉴를 선택합니다.



(3) 속성 페이지를 연 후, 27page의 내용을 참고하여 라이브러리의 경로를 설정합니다.

[출처] http://blog.secmem.org/69

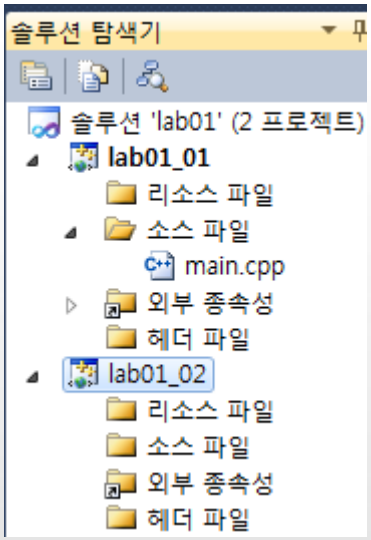
2-1. 실습 및 과제 제출 방법

■ 솔루션 및 프로젝트 생성 방법

❖ **1-1을 참조**하여 솔루션 및 프로젝트 생성하되, 가능하다면 솔루션 명과 프로젝트 명을 다음과 같이 지정한다.

- **솔루션 명** : lab#, hw# (#=순번)
  - Ex) lab01 (실습 1번의 모든 문제를 포함하는 솔루션의 이름)
- **프로젝트 명** : lab#_@, hw#_@ (@=실습 내 문제 번호)
  - Ex) lab01_03 (실습 1번의 3번째 문제)

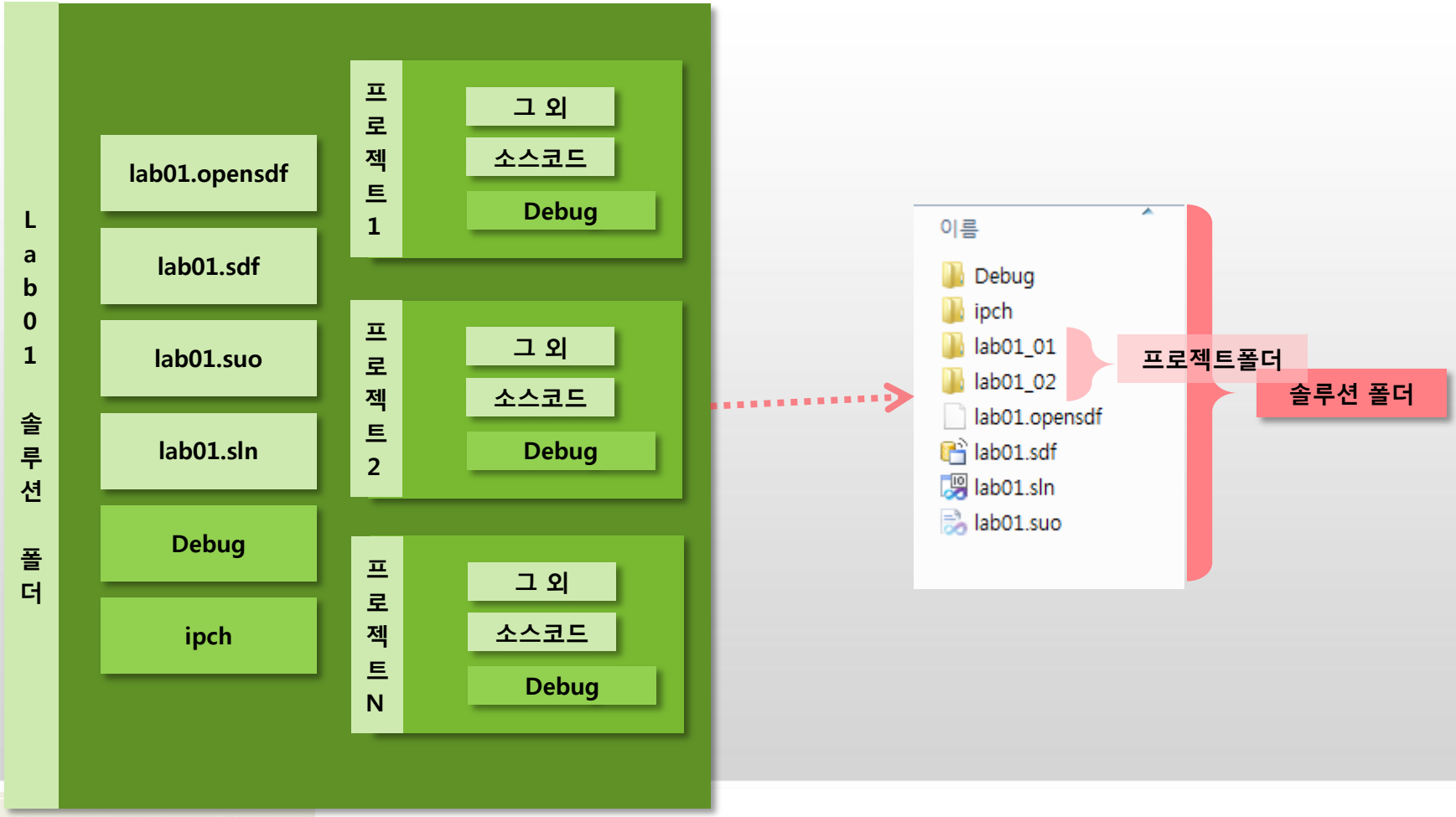
❖ 매주 진행되는 실습은 **한 솔루션 안에 여러 개의 프로젝트로 구분하여 작성한다.** (오른쪽 그림과 같이)



2-1. 실습 및 과제 제출 방법

■ 솔루션 구성

❖ 솔루션 이름을 lab01로 가정하였을 때, 솔루션 폴더는 다음과 같은 파일들로 구성됩니다. (Visual Studio 2010 기준)



2-1. 실습 및 과제 제출 방법

■ 솔루션 구성

❖ X 표시된 파일들을 삭제한 후 솔루션 폴더를 압축합니다.