

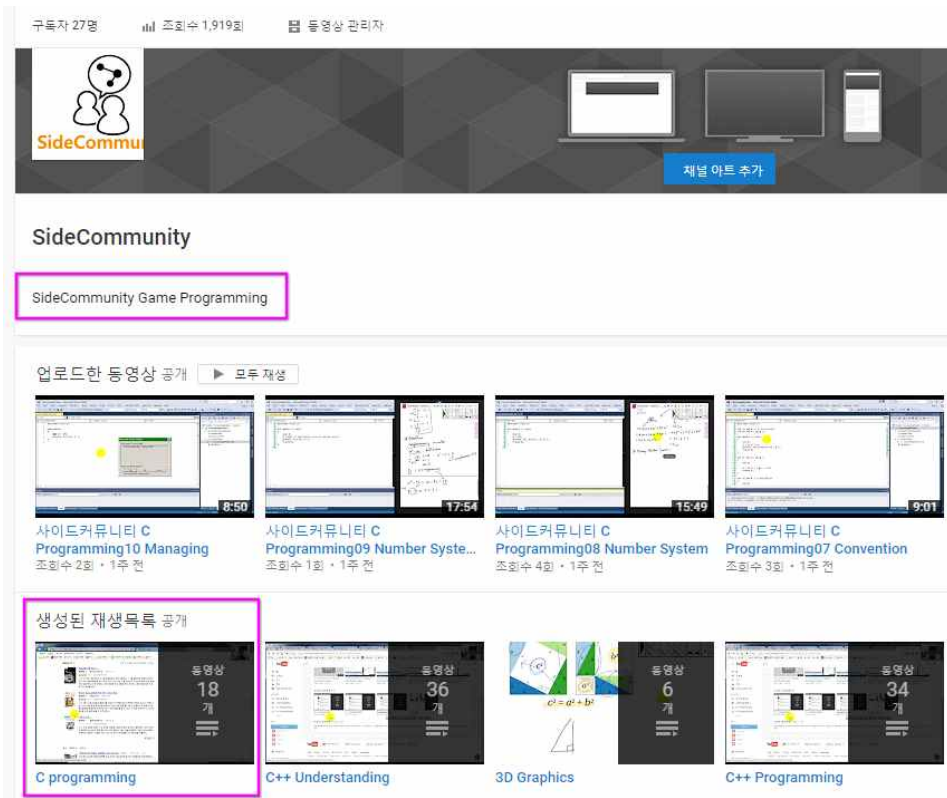


1. 들어가기 전에

- ISO는 1998년 C++98로 알려진 표준을 확립
- 그 뒤의 표준은 2003년 정해진 C++03
- 그 이후로 C++0x라고 알려진 표준이 계속해서 갱신
- 이름이 C++0x라고 알려진 이유는 200x년에는 정해질 것을 기대했기 때문이었는데, 이 표준은 2011년에야 정해지게 되었음
- C++11이 정해진 후, 2014년에는 C++14가 정해짐
- 2017년에는 C++17이 정해짐
- Visual Studio 2013은 C++11의 표준을 모두 지원하지는 않음

- 저자의 동영상 강의를 유튜브에서 시청할 수 있음
- 동영상 강좌는 sidecommunity game programm채널에 준비되어 있음.

<https://www.youtube.com/playlist?list=PLrrTotxaO6kxIHovqDNTdSFZ8tcAvQkO>



The image shows a YouTube channel page for 'SideCommunity'. The channel has 27 subscribers, 1,919 videos, and is managed by '동영상 관리자'. The banner features the SideComm logo and a blue button that says '채널 아트 추가'. Below the banner, the channel name 'SideCommunity' is displayed, followed by the playlist name 'SideCommunity Game Programming' which is highlighted with a pink box. The main content area shows a grid of video thumbnails. The first row contains four videos related to C programming, each with a duration and a title. The second row shows a '생성된 재생목록 공개' (Generated Playlist Public) section with four playlists: 'C programming' (18 videos, highlighted with a pink box), 'C++ Understanding' (36 videos), '3D Graphics' (6 videos), and 'C++ Programming' (34 videos).

구독자 27명 1,919회 동영상 관리자

SideComm

채널 아트 추가

SideCommunity

SideCommunity Game Programming

업로드한 동영상 공개 모두 재생

사이드커뮤니티 C Programming10 Managing
조회수 2회 · 1주 전 8:50

사이드커뮤니티 C Programming09 Number System
조회수 1회 · 1주 전 17:54

사이드커뮤니티 C Programming08 Number System
조회수 4회 · 1주 전 15:49

사이드커뮤니티 C Programming07 Convention
조회수 3회 · 1주 전 9:01

생성된 재생목록 공개

C programming 동영상 18개

C++ Understanding 동영상 36개

3D Graphics 동영상 6개

C++ Programming 동영상 34개

왜 언어의 이름이 C++조?

처음 몇 년 동안은 저는 저의 언어를 "클래스를 가진 C(C with Classes)"라고 불렀습니다. 그러나 사람들은 "클래스를 가진 C"를 "새로운 C(new C)"로 불렀습니다. 그것은 원래의 C를 "평범한 C", "그냥 C(straight C)" 그리고 "옛날 C"로 불리도록 했습니다. 특별히 마지막 이름 C는 모욕적인 것으로 간주되었으므로(역주: Bjarne은 자신이 만든 언어가 C보다 훨씬 강력함을 나타내고자 했습니다), 예의가 필요했으며, 혼돈을 피하기 위해 저는 새로운 이름을 찾게 되었습니다. 저는 C++를 선택했습니다. 왜냐하면, 짧고, 훌륭한 해석을 가지며, "형용사 C"형태가 아니었기 때문입니다. C언어에서 ++는 문맥에 따라 "다음(next)", "계승자(successor)" 혹은 "증가(increment)"로 읽힙니다. 물론 실제 발음은 "plusplus(plus plus)"입니다(역주: 한국 사람들은 대부분 pluss라고 읽습니다. 플러스 플러스를 편하게 발음한 것입니다). C++과 ++C는 농담과 재미를 위한 비욕한 소스입니다. 물론 이러한 농담은 이름이 선택되기 전에 대부분 알려졌습니다. C++이란 이름은 Rick Mascitti에 의해 처음 제안되었습니다.

C++와 C, 무엇이 다른가?

「C++은 C에 기초한 범용 언어이다. C가 제공하는 기능 외에 C++는 **추가적인 데이터 타입, 클래스, 템플릿, 예외처리, 이름공간, 인라인 함수, 연산자 오버로딩, 함수이름 오버로딩, 참조, 메모리관리 연산자와 추가적인 라이브러리 기능을 제공한다**」

환경

- 1998년 초판을 작성할 당시, 필자의 시스템 사양
 - ☐ Pentium[™] 90, 32MegaByte Main Memory, 3GigaByte Hard Disk
 - ☐ 17inch SAMTRON 컬러 모니터
 - ☐ HWP97[™] 워드 프로세서
 - ☐ Visio[™] 드로잉 툴(drawing tool), PaintShopPro[™] 그래픽 툴
- 프로그램 컴파일에 사용한 툴(tool)은 다음과 같음.
 - ☐ Borland C++ 3.1, 4.5, 5.02
 - ☐ C++ Builder 3.0, 4.0, 5.0
 - ☐ Visual C++ 5.0, 6.0

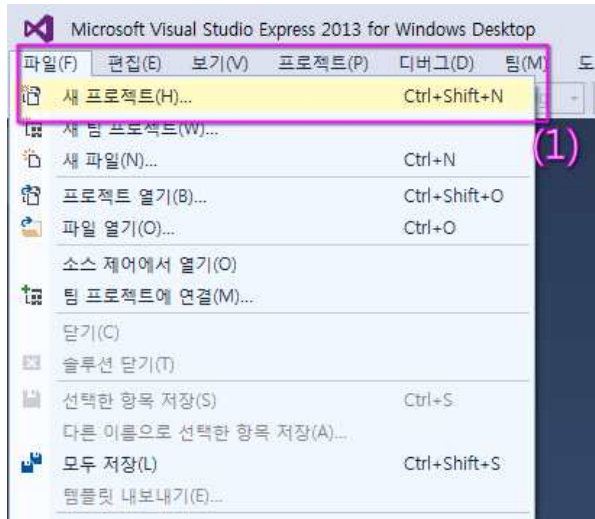
-
- 재판revised edition에 사용한 소스를 컴파일하기 위해 사용한 툴

- Microsoft Visual Studio 2013

- 더 상위 버전의 Visual Studio가 발표되었지만, C++14의 기능을 설명하는데는 Visual Studio 2013으로도 충분하다고 판단

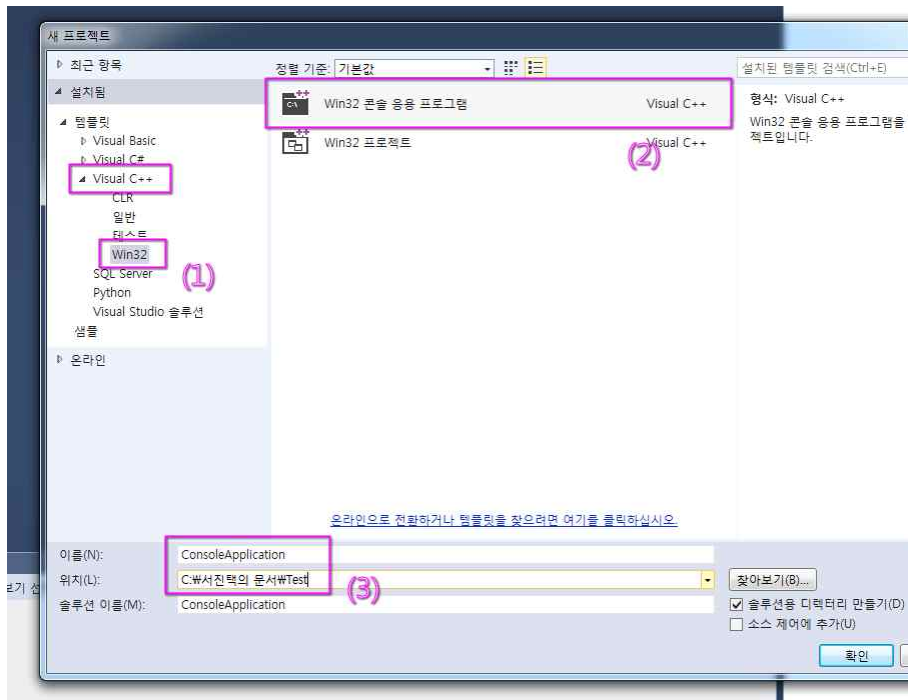
Visual Studio 2013 프로젝트project 설정

- Microsoft Visual Studio 2013을 실행합니다. [파일] 메뉴의 [새 프로젝트(H)...] 항목을 선택합니다. 프로젝트를 만드는 과정을 시작합니다.

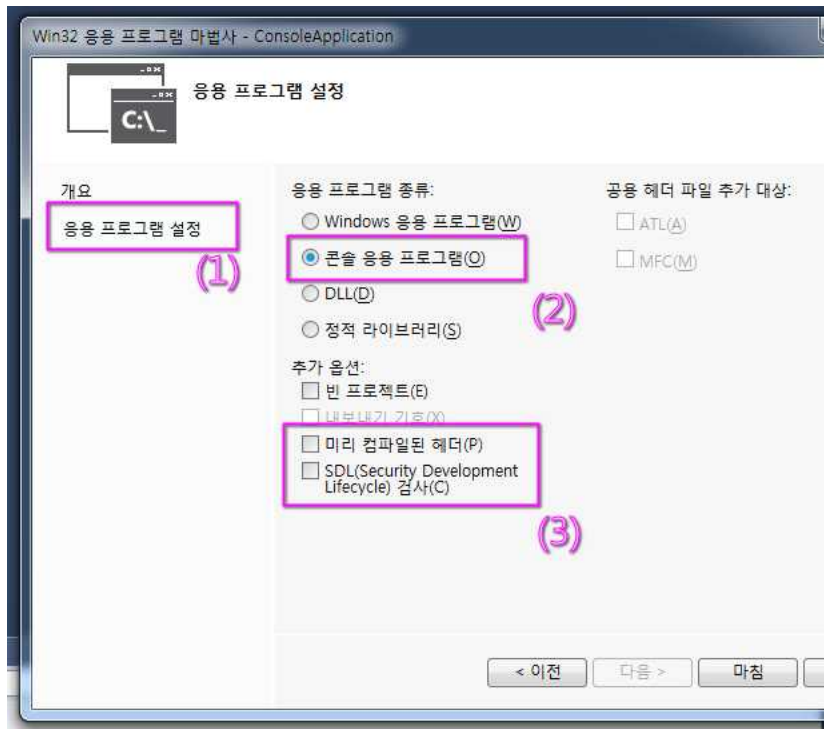


파일→새 프로젝트 메뉴를 선택합니다.

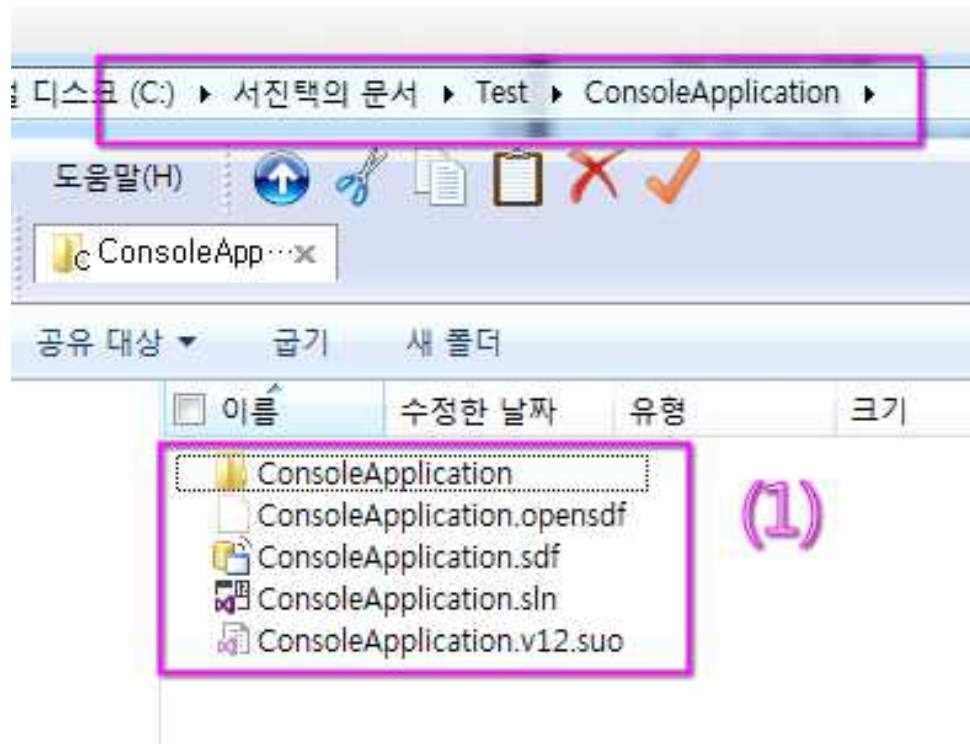
- (1) [Win32] 플랫폼에서 동작하는 간단한 (2) [Win32 콘솔 응용프로그램]을 선택합니다.
(3) 그리고 적절한 폴더의 위치를 지정하고 프로젝트 이름을 ConsoleApplication으로 설정합니다.



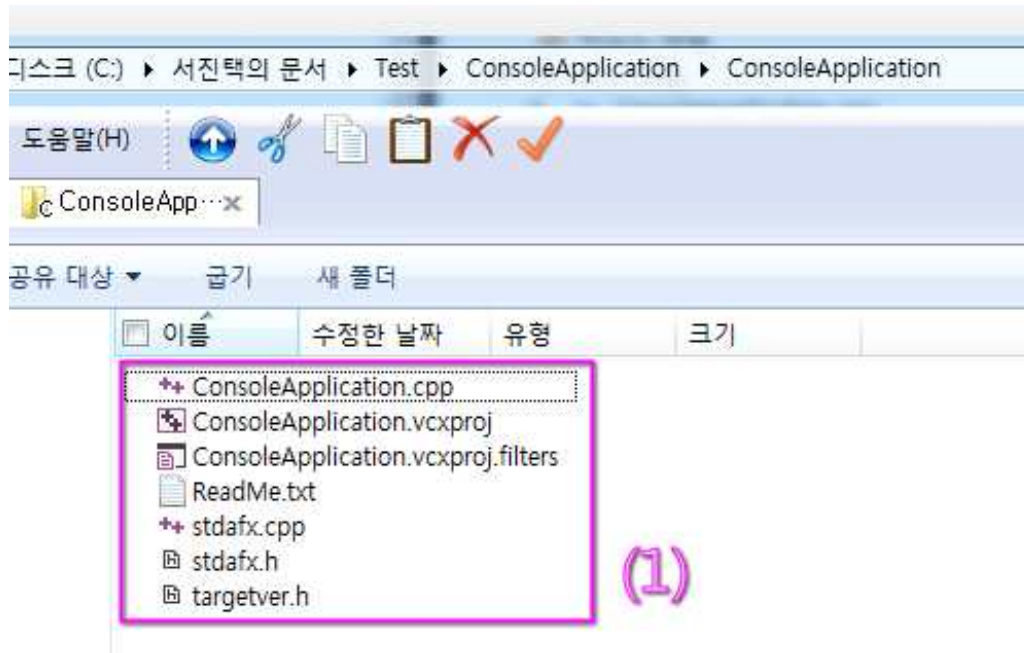
- 생성되는 코드를 간단하게 하기 위해서, (1) [응용 프로그램 설정]에서, (3) [미리 컴파일된 헤더(P)]와 [SDL 검사(C)]의 체크 박스check box를 해제합니다. 그리고 [마침] 버튼을 선택해서 프로젝트를 생성합니다.



- 프로젝트 생성이 완료되면 대상 폴더로 가서 생성된 파일들의 목록을 확인합니다. 파일 확장자가 .sln인 파일이 솔루션solution 파일인데, 솔루션 파일은 여러 개의 프로젝트를 포함하는 파일입니다.



- ConsoleApplication 폴더로 이동해서 생성된 프로젝트 파일들을 확인



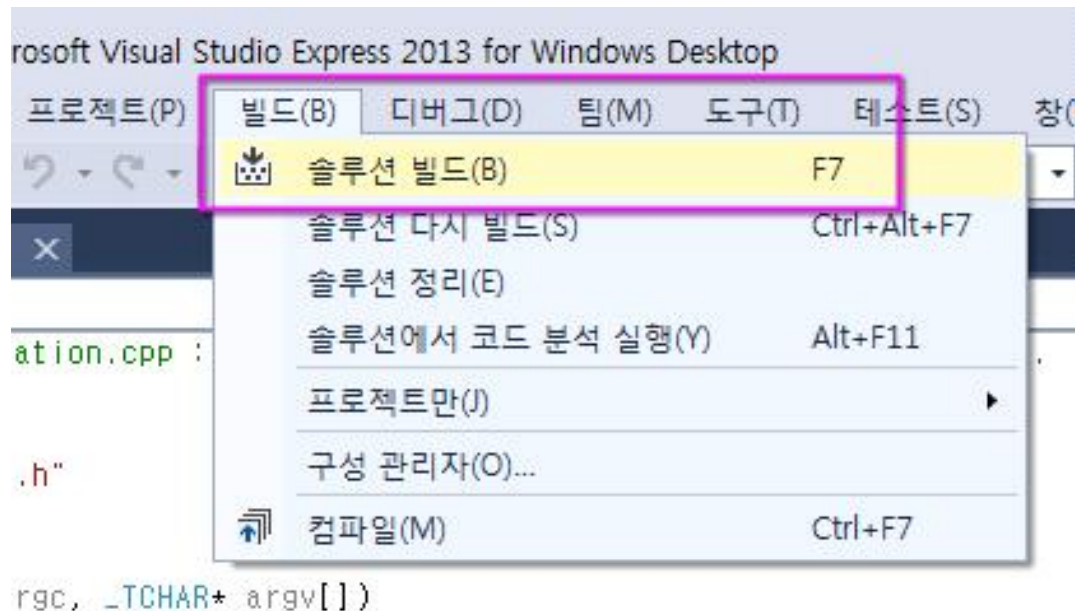
ConsoleApplication폴더로 이동하면, C++소스 파일인 ConsoleApplication.cpp를 확인할 수 있습니다. 이 파일에 책의 소스를 입력해서 테스트합니다.

-
- ConsoleApplication.cpp에 자동으로 생성된 소스 코드의 내용은 아래와 같습니다.

```
#include "stdafx.h"
```

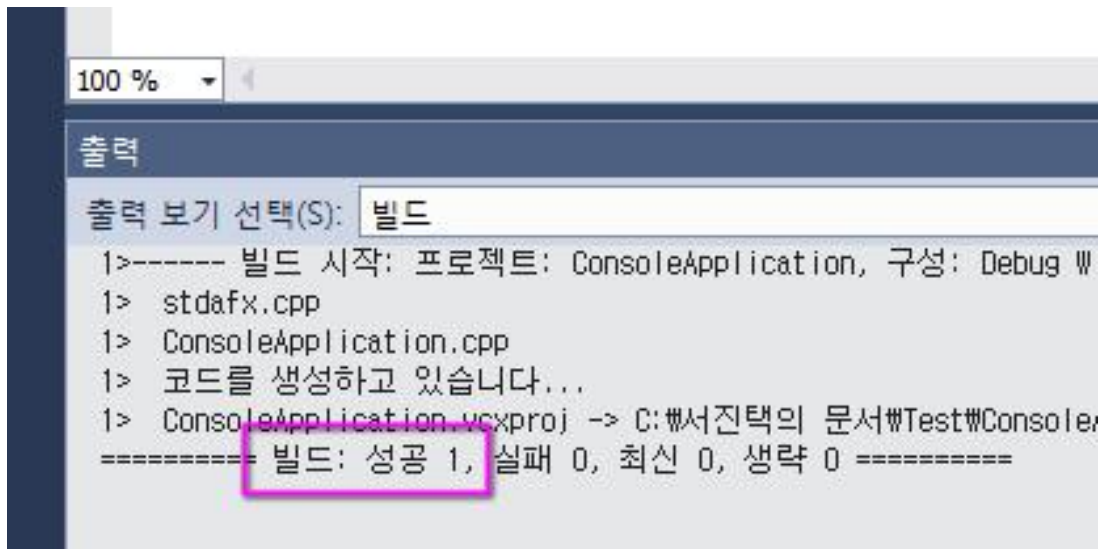
```
int _tmain(int argc, _TCHAR* argv[])  
// int main(void) // 이렇게 작성해도 됩니다.  
{  
    return 0;  
}
```

- 이제 [빌드]메뉴의 [솔루션 빌드(B)]를 선택합니다. 그러면, Visual Studio는 프로젝트에 있는 소스 파일들을 컴파일compile하고 링크link한 후에, 실행 파일을 생성합니다.



[빌드→솔루션 빌드]를 선택합니다.

- [출력] 창의 출력 문자열을 확인해서, 빌드가 성공한 것을 확인합니다.



```
100 %
출력
출력 보기 선택(S): 빌드
1>----- 빌드 시작: 프로젝트: ConsoleApplication, 구성: Debug W
1> stdafx.cpp
1> ConsoleApplication.cpp
1> 코드를 생성하고 있습니다...
1> ConsoleApplication.vcxproj -> C:\서진택의 문서\Test\ConsoleApplication.exe
===== 빌드: 성공 1, 실패 0, 최신 0, 생략 0 =====
```



출력창에 빌드 성공 메시지가 출력됩니다.

- 책의 소스들은 방금 생성한 프로젝트의 ConsoleApplication.cpp에 작성한 후 빌드하고 테스트할 수 있습니다.

int가 차지하는 메모리의 크기

- MS-DOS에서 사용하던 컴파일러에서 int는 16비트 이었습니다. 후에 Win16에서도 int는 16비트 이었지만, Win32에서는 int는 32비트입니다.
- 16비트 운영체제에서 4바이트의 정수를 선언하기 위해서는 long이라고 선언해야 했습니다. 그런데 Win32환경에서는 int와 long 모두 4바이트를 차지하는 정수형 변수가 되었습니다.
- Win64가 나오게 되면 int는 64비트가 될 작정이었습니다. 왜냐하면 C 표준에서 int의 크기는 컴퓨터의 워드 크기와 일치한다는 규정 때문이었습니다. 하지만 소스 코드 호환성 문제 때문에 Win64 환경에서 int와 long은 모두 4바이트를 차지합니다. 그리고 C++14표준에서 long long이라는 8바이트 정수가 추가되었습니다.