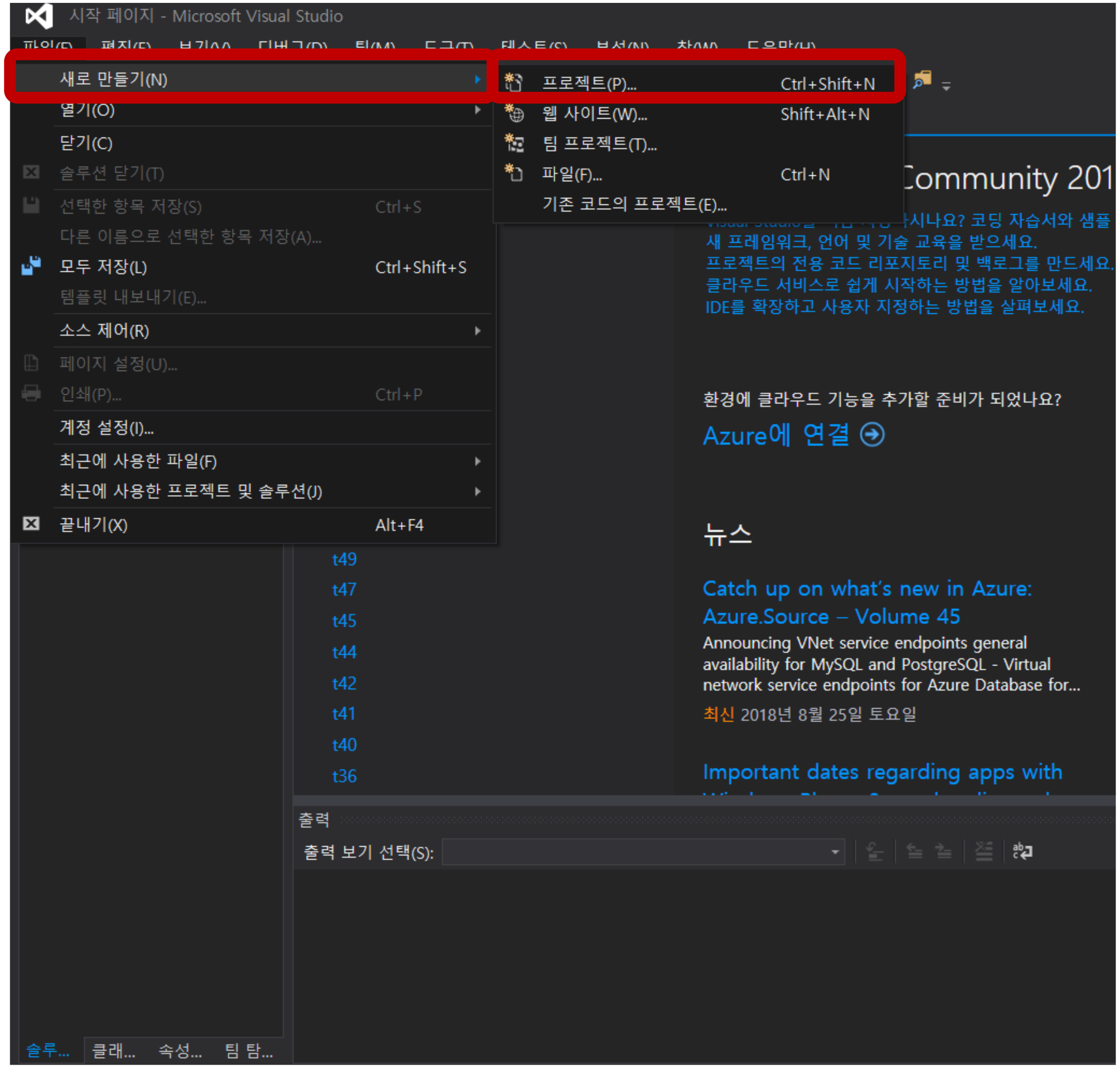
C++ 프로그래밍 실습

Visual Studio 2015
Basic Concepts

Contents

- Understanding basic concepts of OO(Object-Oriented) programming with simple C++ codes
  - Understanding classes
    - Data members and methods
    - Class objects
  - Understanding inheritance
    - Inherited data members
    - Inherited methods
  - Understanding polymorphism
    - Polymorphic method definition
    - Polymorphic method invocation

Create Project



시작 페이지 - Microsoft Visual Studio
파일(F) 편집(E) 보기(V) 디버그(D) 팀(M) 도구(T) 테스트(S) 분석(N) 창(W) 도움말(H)
새로 만들기(N)
열기(O)
닫기(C)
솔루션 닫기(T)
선택한 항목 저장(S) Ctrl+S
다른 이름으로 선택한 항목 저장(A)...
모두 저장(L) Ctrl+Shift+S
템플릿 내보내기(E)...
소스 제어(R)
페이지 설정(U)...
인쇄(P)... Ctrl+P
계정 설정(I)...
최근에 사용한 파일(F)
최근에 사용한 프로젝트 및 솔루션(J)
끝내기(X) Alt+F4
프로젝트(P)... Ctrl+Shift+N
웹 사이트(W)... Shift+Alt+N
팀 프로젝트(T)...
파일(F)... Ctrl+N
기존 코드의 프로젝트(E)...
t49
t47
t45
t44
t42
t41
t40
t36
출력
출력 보기 선택(S):
솔루... 클래... 속성... 팀 탐...

Community 201

시나요? 코딩 자습서와 샘플
새 프레임워크, 언어 및 기술 교육을 받으세요.
프로젝트의 전용 코드 리포지토리 및 백로그를 만드세요.
클라우드 서비스로 쉽게 시작하는 방법을 알아보세요.
IDE를 확장하고 사용자 지정하는 방법을 살펴보세요.

환경에 클라우드 기능을 추가할 준비가 되었나요?

Azure에 연결

뉴스

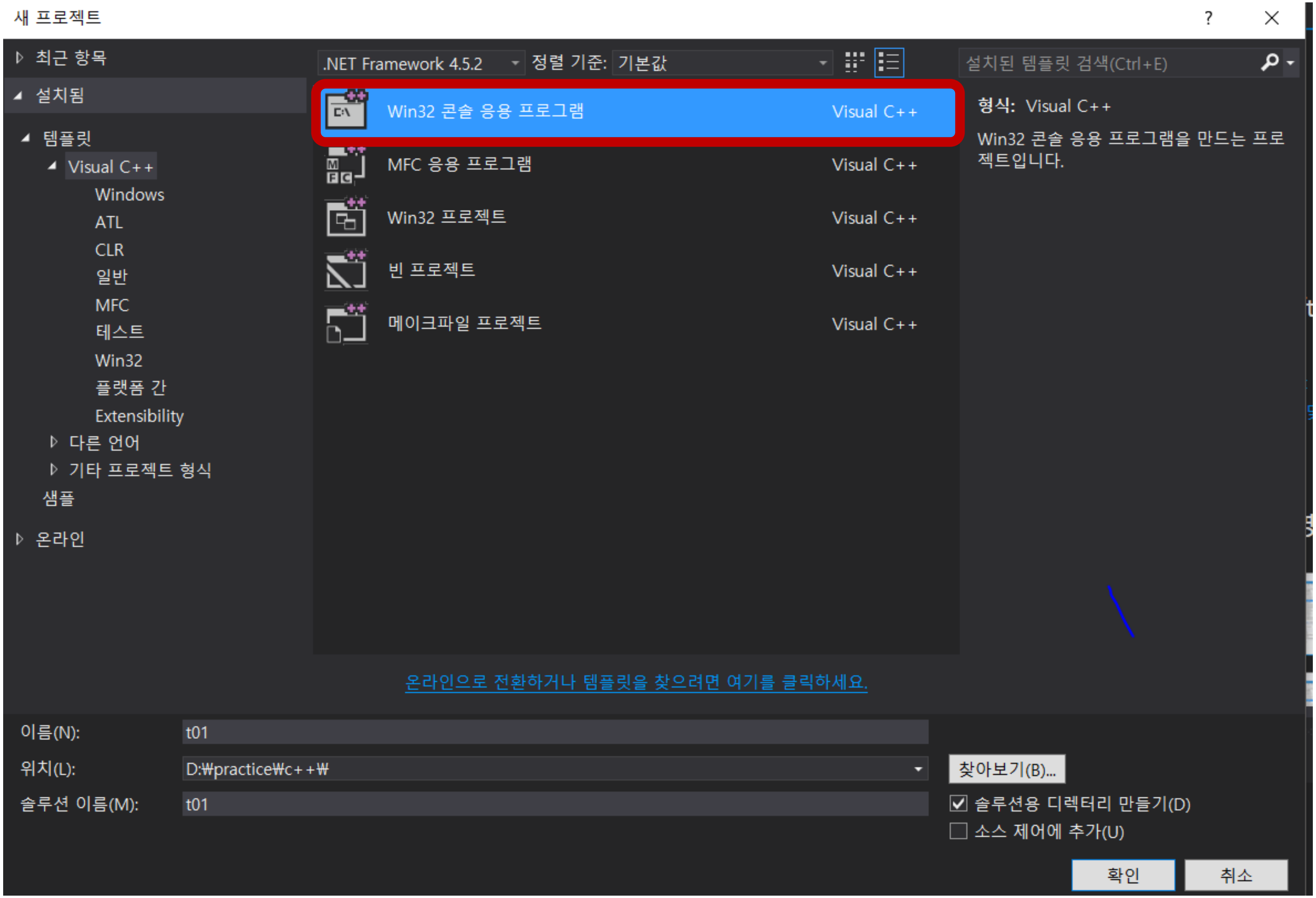
Catch up on what's new in Azure: Azure.Source – Volume 45

Announcing VNet service endpoints general availability for MySQL and PostgreSQL - Virtual network service endpoints for Azure Database for...

최신 2018년 8월 25일 토요일

Important dates regarding apps with

Create Project



새 프로젝트

- 최근 항목
- 설치됨
  - 템플릿
    - Visual C++
      - Windows
      - ATL
      - CLR
      - 일반
      - MFC
      - 테스트
      - Win32
      - 플랫폼 간
      - Extensibility
    - 다른 언어
    - 기타 프로젝트 형식
  - 샘플
- 온라인

.NET Framework 4.5.2 　정렬 기준: 기본값

설치된 템플릿 검색(Ctrl+E)

Win32 콘솔 응용 프로그램	Visual C++
MFC 응용 프로그램	Visual C++
Win32 프로젝트	Visual C++
빈 프로젝트	Visual C++
메이크파일 프로젝트	Visual C++

형식: Visual C++

Win32 콘솔 응용 프로그램을 만드는 프로젝트입니다.

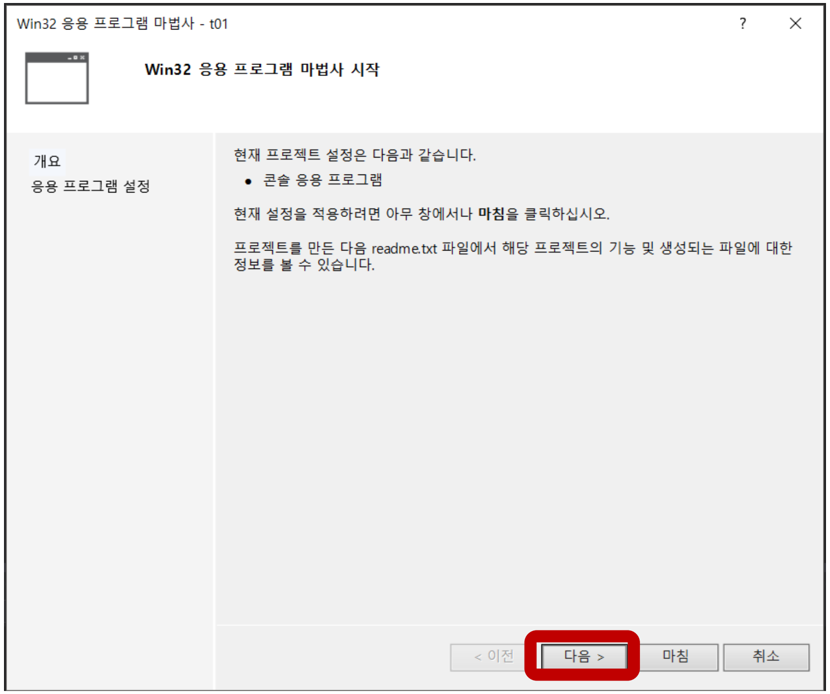
온라인으로 전환하거나 템플릿을 찾으려면 여기를 클릭하세요.

이름(N):	t01
위치(L):	D:₩practice₩c++₩
솔루션 이름(M):	t01

☑ 솔루션용 디렉터리 만들기(D)

□ 소스 제어에 추가(U)

Create Project



Win32 응용 프로그램 마법사 - t01

Win32 응용 프로그램 마법사 시작

개요

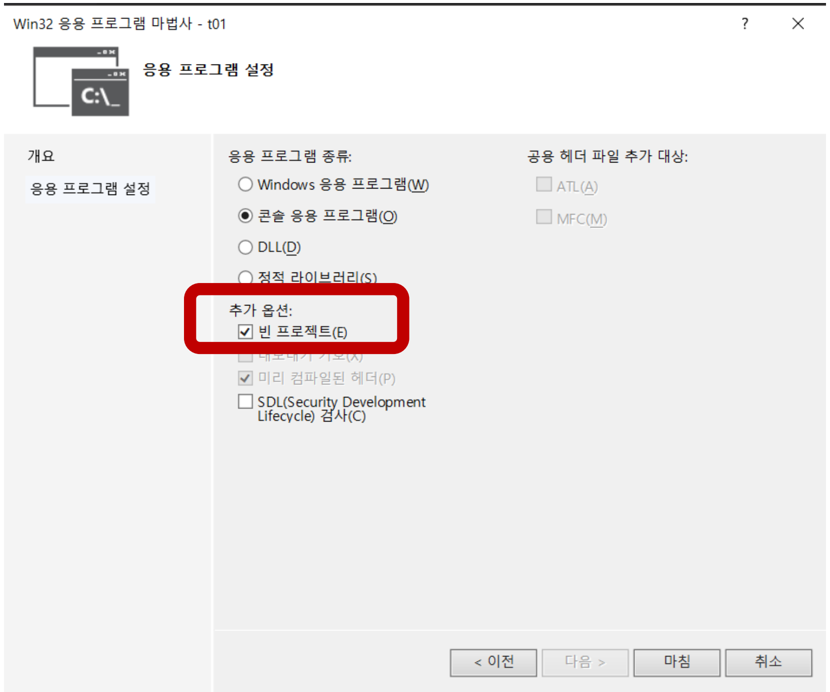
응용 프로그램 설정

현재 프로젝트 설정은 다음과 같습니다.

- 콘솔 응용 프로그램

현재 설정을 적용하려면 아무 창에서나 **마침**을 클릭하십시오.

프로젝트를 만든 다음 readme.txt 파일에서 해당 프로젝트의 기능 및 생성되는 파일에 대한 정보를 볼 수 있습니다.



Win32 응용 프로그램 마법사 - t01

응용 프로그램 설정

개요

응용 프로그램 설정

응용 프로그램 종류:

○ Windows 응용 프로그램(W)
● 콘솔 응용 프로그램(O)
○ DLL(D)
○ 정적 라이브러리(S)

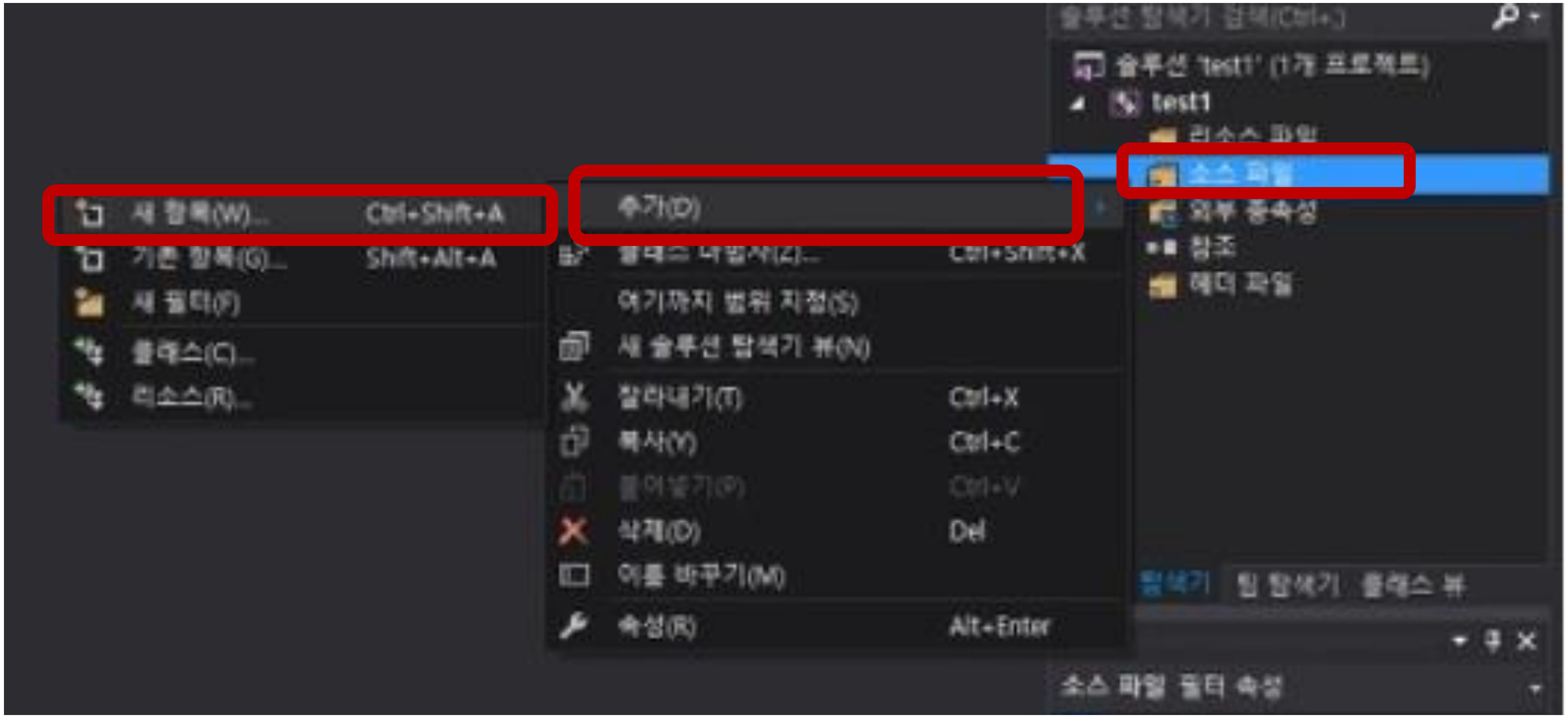
추가 옵션:

☑ 빈 프로젝트(E)
□ 내보내기 기호(X)
☑ 미리 컴파일된 헤더(P)
□ SDL(Security Development Lifecycle) 검사(C)

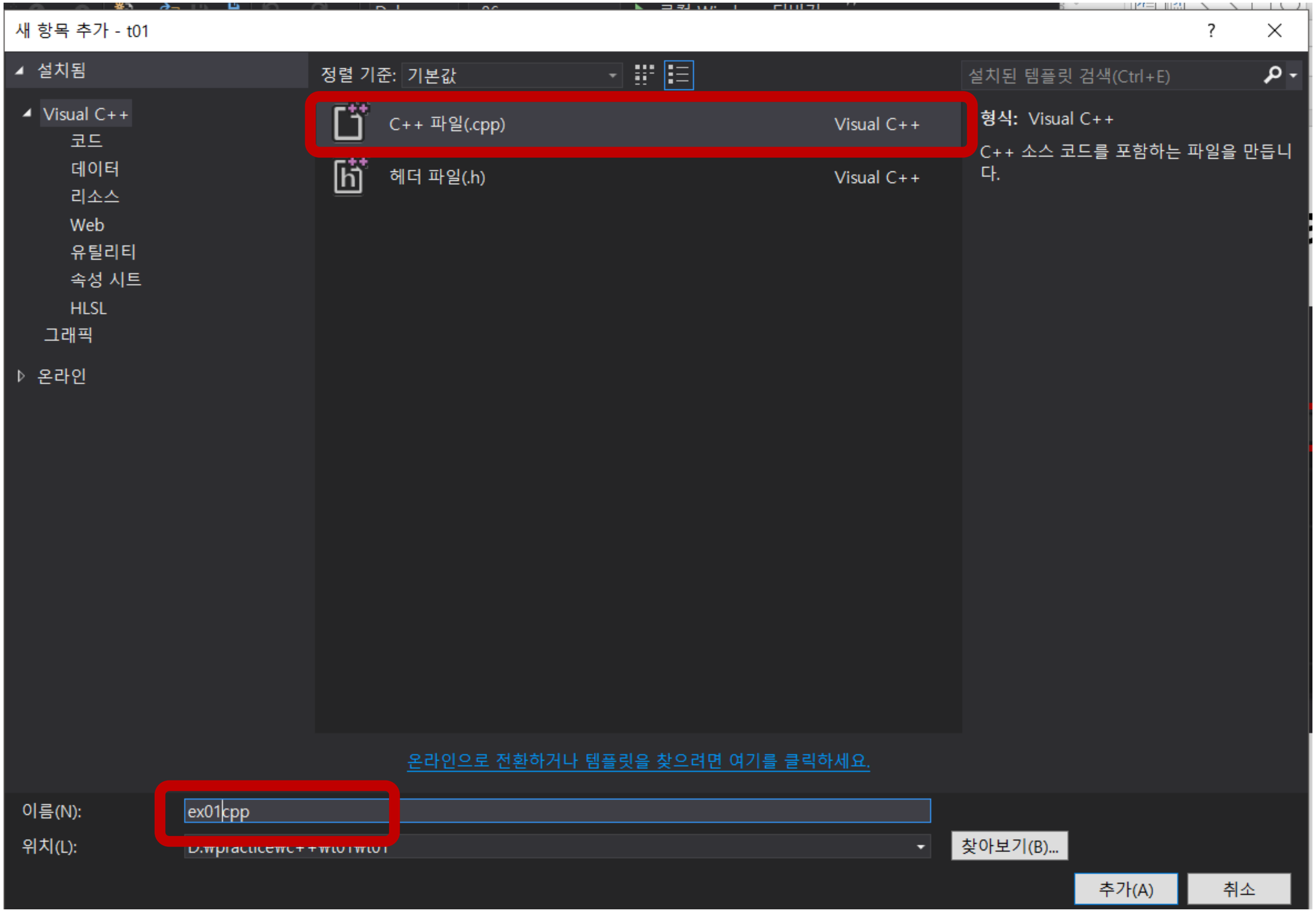
공용 헤더 파일 추가 대상:

□ ATL(A)
□ MFC(M)

Create Project



Create Project



정렬 기준: 기본값

C++ 파일(.cpp) Visual C++

형식: Visual C++

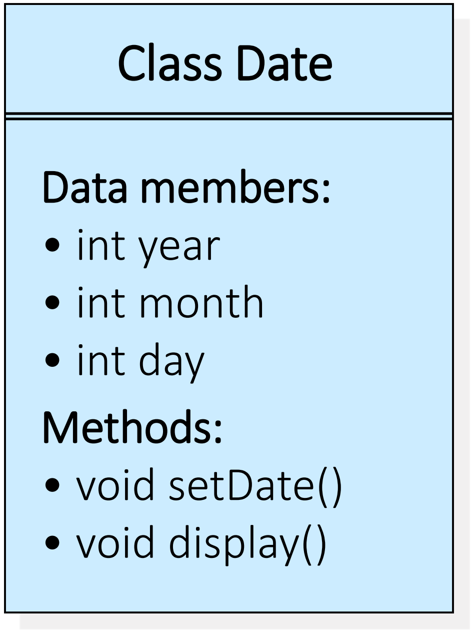
C++ 소스 코드를 포함하는 파일을 만듭니다.

이름(N): ex01cpp

위치(L): D:₩practice₩C++₩t01₩t01

Practice 1 : Class

- Introduce how to create and use classes
  - Creating class “Date”
    - Defining data members (private)
      - Integer variables year, month, and day
    - Defining methods (public)
      - “setDate”: sets values of the data members
      - “display”: displays values of the data members
  - Using class “Date” in the main() function
    - Creating an object “birthday” of Class “Date”
    - Invoking “setDate” and “display” methods of the object “birthday”



Class Date

Data members:

- int year
- int month
- int day

Methods:

- void setDate()
- void display()

```
#include<iostream>

class Date {
private:
    int year;
    int month;
    int day;

public:
    void setDate(int yy, int mm, int dd) {
        year = yy;
        month = mm;
        day = dd;
    }

    void display() {
        std::cout << year << "." << month << "." << day << std::endl;
    }
};
void main(void) {
    Date birthday;

    birthday.setDate(1999, 11, 22);
    birthday.display();
}
```

Class definition

Member variables

Methods

Main function

Creating Date object

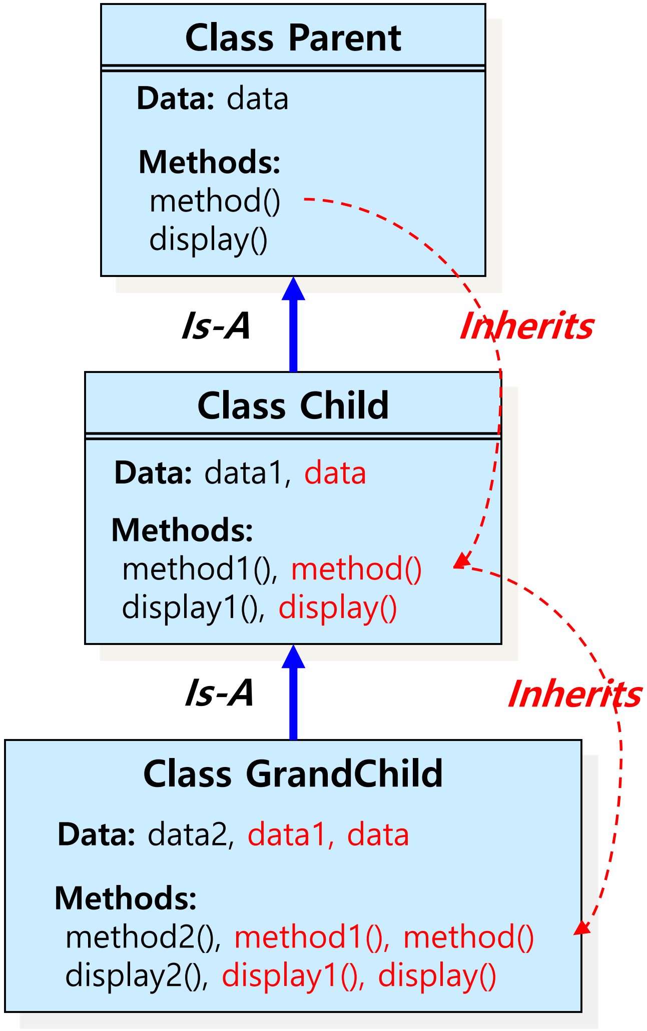
Method invocations

Execution Result:

```
1999.11.12
```

Practice 2 : Inheritance(Single)

- Introduce how to inherit from classes
  - Creating three classes
    - Class “Parent,” “Child,” and “GrandChild”
  - Subclasses Inherit from super classes
    - Class “Child” inherits from “Parent”
    - Class “GrandChild” inherits from “Child” and “Parent”
      - Protected data members
      - Public methods
  - Using Class “GrandChild”
    - Public methods are used in the main function
    - Each inherited public method can be used as if it is a member method of the Class “GrandChild”



```cpp
#include<iostream>

class Parent {
protected:
   int data;

public:
   void method(int param) {
      data = param;
   }

   void display() {
      std::cout << data << std::endl;
   }
};

class Child : public Parent {
protected:
   int data1;

public:
   void method1(int param1) {
      data1 = param1;
   }

   void display1() {
      std::cout << data << "," << data1 << std::endl;
   }
};
```

Class Parent

Class Child

Inherits from Class "Parent"

Inherited data member

Class GrandChild

```
class GrandChild : public Child {
private:
    int data2;

public:
    void method2(int param2) {
        data2 = param2;
    }

void display2() {
    std::cout << data << "," << data1 << "," << data2 << std::endl;
    }
};
```

Inherits from Class "Child"

Inherited data member

Main function

```
void main() {
    GrandChild obj;

    obj.method(10);
    obj.display();

    obj.method1(20);
    obj.display1();

    obj.method2(30);
    obj.display2();
}
```

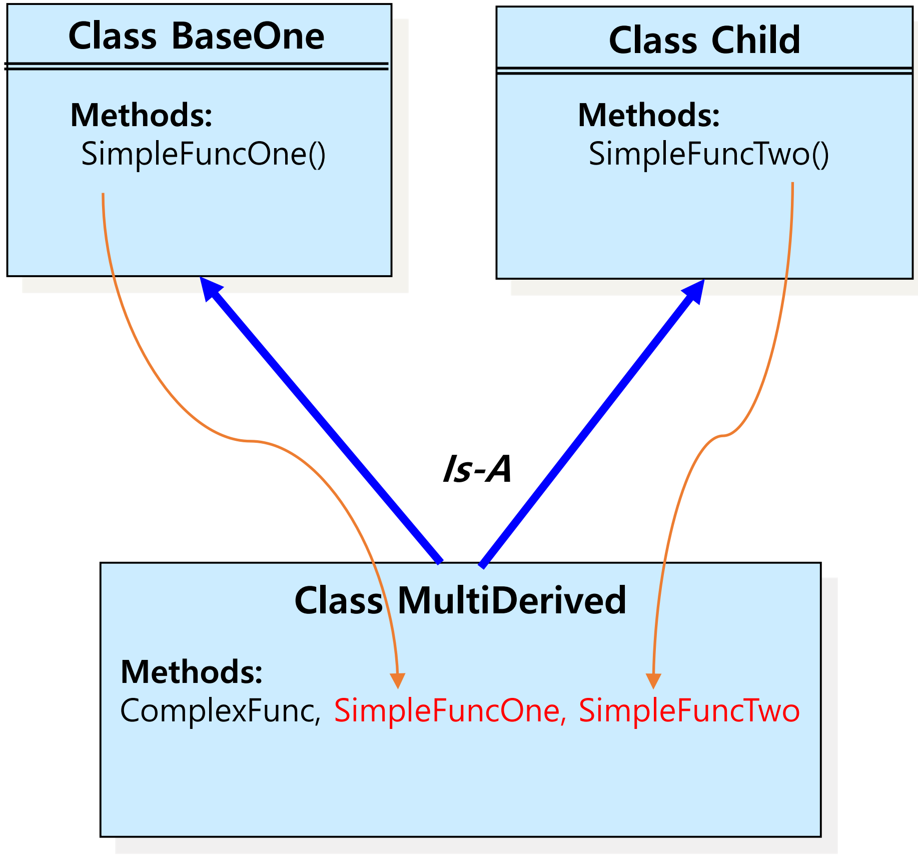
Inherited method invocations

Execution Result:

```
10
10,20
10,20,30
```

Practice 3 : Inheritance(Multiple)

- Introduce how to inherit from classes
  - Creating three classes
    - Class “BaseOne,” “BaseTwo,” and “MultiDerived”
  - Using Class “MultiDerived”
    - Public methods are used in the main function
    - Each inherited public method can be used as if it is a member method of the Class “BaseOne”, “Basetwo”



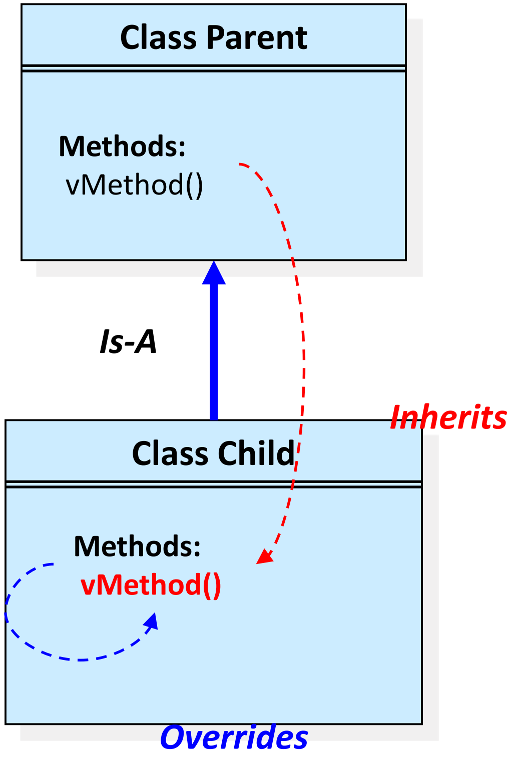
```cpp
#include<iostream>

class BaseOne {
public:
    void SimpleFuncOne() {
        std::cout << "BaseOne" << std::endl;;
    }
};

class BaseTwo {
public:
    void SimpleFuncTwo() {
        std::cout << "BaseTwo" << std::endl;
    }
};

class MultiDerived : public BaseOne, public BaseTwo {
public:
    void ComplexFunc() {
        SimpleFuncOne();
        SimpleFuncTwo();
    }
};

int main(void) {
    MultiDerived mdr;
    mdr.ComplexFunc();
    return 0;
}
```

Execution Result:

```
BaseOne
BaseTwo
```

Practice 4 : Polymorphism

- Introduce how to use polymorphic methods
  - Creating two classes
    - Class “Parent” and “Child”
    - Each class has a method named “vMethod”
    - Class “Child” inherits from Class “Parent”
  - Overriding polymorphic method
    - Method “vMethod” is a polymorphic method
    - “vMethod” of “Child” is inherited from the “Parent” and overridden
    - In the main function, each “vMethod” is invoked once and displays different outputs



```
#include<iostream>
```

Class Parent

```
class Parent {
public:
   virtual void vMethod() {
      std::cout << "This is a method of the parent class." << std::endl;
   }
};
```

Class Child

```
class Child : public Parent {
public:
   void vMethod() {
      std::cout << "This is a method of the child class." << std::endl;
   }
};
```

Polymorphic method definitions

Main function

```
void main() {
   Parent *pp, p;
   Child c;

   pp = &p;
   pp->vMethod();

   pp = &c;
   pp->vMethod();
}
```

Parent's vMethod called

Child's vMethod called

Execution Result:

```
This is a method of parent class.
This is a method of child class.
```

Course Homepage

- How to access
  - URL: sclab.konkuk.ac.kr
- Downloading class material
  - Students can download syllabus and lecture notes in PDF format
- Class announcement
  - About homework and project
  - Exam schedule and result
  - And so on

Exercise

- 자신만의 개성 있는 Class를 설계하시오

  - 최소 4개의 변수와 3개의 Method를 포함한다.

  - Inheritance를 사용하시오.

  - polymorphism은 가능하다면 사용하시오.

- 주석은 필수

Submit

- Teaching assistant: 장성수
  Office: 신공학관 1216호 (대학원 SCLab 연구실)
  Email: pik1100@naver.com

- Title of the email : [2018][Practice#]_student# _ student _ name

- Ex) [2018][Practice01]_201700000_장성수

- 제출일: 매주 화요일 23:59분 까지

- Create zip file. (C++ project folder)

- 주의 : 메일 양식이 잘못될 경우 채점이 되지 않을 수 있음.

- 질문 메일 : pik1100@naver.com : 장성수

끝