# 창의적 소프트웨어 설계 실습 문제 10 – hw10-1

**제출 기한**

11월 20일 화 11:59 PM

1. hw10-1(mkdir hw10-1)라는 폴더를 만들고 GitLab에 push
2. hw10-1 디렉토리에 CMakeLists.txt과 각 문제에서 요구하는 파일들을 작성
3. **‘cmake’ 와 ‘make’명령을 수행하여 숙제가 모두 빌드**
4. 최종 버전을 GitLab에 commit
5. **시간과 파일명, 입력과 출력 방식 반드시 지키기**

## 과제 1.

아래의 요구 조건을 만족하고 아래 실행 예와 같이 동작하는 프로그램을 작성하시오.

* A라는 클래스를 작성하고, public member function으로 test1()과 test2()함수를 작성한다. A::test1()함수는 “A::test1()”라는 문자열을 출력해야 하며, A::test2()는 순수 가상 함수(pure virtual function)이다.
* A를 상속받는 B라는 클래스는 “B::test2()”라는 문자열을 출력하도록 test2() 함수를 구현한다.
* A를 상속받는 C라는 클래스는 “C::test1()”라는 문자열을 출력하는 test1()함수와 “C::test2()”를 출력하는 test2() 함수를 가진다.
* B를 상속받는 BB라는 클래스는 “BB:test1()”라는 문자열을 출력하는 test1() 함수를 가진다.
* 사용자로부터 정해지지 않은 개수의 “B”, “C”, “BB” 문자열을 임의의 순서로 받아, 각각의 문자열에 해당하는 B, C, BB 클래스의 객체를 하나씩 생성해서 std::vector<A\*> 타입의 벡터 objects에 저장한다.
* 사용자가 “0”을 입력하여 입력이 종료되면, objects의 각 요소마다 test1()과 test2()를 순서대로 실행하여 아래와 같이 실행 결과를 보이는 프로그램을 작성하시오
* 사용이 끝난 objects의 각 요소는 할당받은 메모리를 해제해야 하며, 프로그램 코드 전체에 걸쳐 type casting 연산자는 사용하면 안 된다.

파일명 : class\_function (class\_function.h class\_function.cc class\_function\_main.cc)

입력 : 각 객체를 의미하는 문자열

출력 : 입력받았던 각 객체에 알맞은 문자열 출력

|  |
| --- |
| **$ ./print\_shape**  **B**  **C**  **BB**  **0**  **A::test1()**  **B::test2()**  **C::test1()**  **C::test2()**  **BB::test1()**  **B::test2()**  **$** |
|  |

## 과제 2.

아래 프로그램은 2를 출력한다. 실제 나눗셈의 결과인 2.5가 출력되도록 C++의 type casting operator 중 하나를 단 한번만 사용하여 아래 프로그램을 수정하시오 (단 C의 casting operator 혹은 C++의 reinterpret\_cast는 사용하지 않도록 한다).

#include <iostream>

using namespace std;

int main()

{

int a = 10;

int b = 4;

cout << a / b << endl;

return 0;

}

파일명 : cast\_quiz (cast\_quiz.cc)

입력 : 없음

출력 : 2.5

|  |
| --- |
| **$ ./cast\_quiz**  **2.5**  **$** |

## 과제 3.

아래와 같은 클래스들이 있다. 아래 클래스 코드를 그대로 이용하여 아래에서 설명하는 프로그램을 작성하시오.

class B

{

public:

virtual ~B() {}

};

class C : public B

{

public:

void test\_C() { std::cout << "C::test\_C()" << std::endl; }

};

class D : public B

{

public:

void test\_D() { std::cout << "D::test\_D()" << std::endl; }

};

* 사용자로부터 정해지지 않은 개수의 “B”, “C”, “D” 문자열을 임의의 순서로 받아, 각각의 문자열에 해당하는 B, C, D 클래스의 객체를 하나씩 생성해서 std::vector<B\*> 타입의 벡터 objects에 저장한다.
* 사용자가 “0”을 입력하여 입력이 종료되면, for문에서 objects의 각 요소를 순회하며 objects의 각 요소가 C타입 객체인 경우에는 C::test\_C()를, D타입 객체인 경우에는 D::test\_D()를 호출하도록 한다 (dynamic\_cast를 이용). B타입 객체인 경우에는 아무 것도 출력하지 않는다.
* **단, 이 문제는 dynamic\_cast의 사용 방법을 연습하기 위한 것으로, 실제로 이러한 용도로 dynamic\_cast를 사용하는 코드는 바람직하지 않게 설계된 코드라는 것을 기억하기 바람.**
* 사용이 끝난 objects의 각 요소는 할당받은 메모리를 해제해야 한다

파일명 : dynamic\_cast (dynamic\_cast.cc dynamic\_cast.h dynamic\_cast\_main.cc)

입력 : 각 객체를 의미하는 문자

출력 : 저장된 각 객체에 대해 과제가 요구하는 문자열

|  |
| --- |
| **$ ./dynamic\_cast**  **B**  **C**  **D**  **B**  **C**  **0**  **C::test\_C()**  **D::test\_D()**  **C::test\_C()**  **$** |