

## 창의적 소프트웨어 설계 실습 문제 9 – hw9-1

### 제출 기한

11 월 13 일 화 23:59 PM

- I. hw9-1(mkdir hw9-1)라는 폴더를 만들고 GitLab 에 push
- II. hw9-1 디렉토리에 Makefile 과 각 문제에서 요구하는 파일들을 작성
- III. **'make'명령을 수행하여 숙제가 모두 빌드**
- IV. 최종 버전을 GitLab 에 commit
- V. **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 test() 함수를 작성한다. A::test() 함수는 "A::test()"라는 문자열을 출력해야 한다.
- A를 상속받는 B라는 클래스 역시 "B::test()"라는 문자열을 출력하는 test() 함수를 가지고 있으며,
- B를 상속받는 C라는 클래스 역시 "C::test()"라는 문자열을 출력하는 test() 함수를 가지고 있어야 한다.
- std::vector<A\*> 타입의 벡터 objects에 A, B, C 클래스의 객체를 각각 1개씩 new로 생성하여 넣고, objects의 각 요소의 test() 함수를 호출하여 아래와 같이 실행 결과를 보이는 프로그램을 작성하시오.
- 사용이 끝난 objects의 각 요소는 할당 받은 메모리를 해제해야 하며, 프로그램 코드 전체에 걸쳐 type casting 연산자는 사용하면 안 된다.

파일명 : classes (classes.h classes.cc classes\_main.cc)

입력 : 없음

출력 : 각 객체의 정보 출력

```
$ ./classes
A::test()
B::test()
C::test()
$
```

## 과제 2.

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

- C 클래스는 B 클래스를 상속받고, B 클래스는 A 클래스를 상속받는다.
- A, B, C 클래스 각각에 `getTypeInfo()` 멤버 함수를 추가한다.
- `A::getTypeInfo()`은 "This is an instance of class A",  
`B:: getTypeInfo()`은 "This is an instance of class B",  
`C:: getTypeInfo()`은 "This is an instance of class C"라는  
"string 객체를 반환"하도록 한다 (화면에 바로 출력하는 것이 아님).
- global scope에  
`void printObjectTypeInfo1(A* object),`  
`void printObjectTypeInfo2(A& object)`  
라는 함수들을 각각 선언하고,  
두 함수 모두 인자로 전달받은 객체의 `getTypeInfo()`를 호출하여 얻은 string 객체를 출력  
하도록 구현한다.
- `std::vector<A*>` 타입의 벡터 `objects`에 A, B, C 클래스의 객체를 각각 1개씩 new로 생성  
하여 넣고, `objects`의 각 요소를 인자로 하여 `printObjectTypeInfo1()`와  
`printObjectTypeInfo2()` 함수를 순서대로 호출한다.
- 사용이 끝난 `objects`의 각 요소는 할당 받은 메모리를 해제해야 하며, 프로그램 코드 전체  
에 걸쳐 type casting 연산자는 사용하면 안 된다.

파일명 : `print_info` (`print_info.h` `print_info.cc` `print_info_main.cc`)

입력: 없음

출력: 과제에서 요구하는 문자열 출력

```
$ ./print_info
This is an instance of class A
This is an instance of class A
This is an instance of class B
This is an instance of class B
This is an instance of class C
This is an instance of class C
$
```

### 과제 3.

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

- C 클래스는 B 클래스를 상속받고, B 클래스는 A 클래스를 상속받는다.
- A 클래스에 int\*형 private member variable인 memberA를 추가한다
- B 클래스에 double\*형 private member variable인 memberB를 추가한다.
- C 클래스에 std::string\*형 private member variable인 memberC를 추가한다.
- A 클래스의 생성자(constructor)는 int형 인자를 하나 받으며,  
new를 이용해 memberA를 할당하고 할당된 공간에 인자로 받은 정수를 저장한 후  
"new memberA"라는 문자열을 출력한다.
- B 클래스의 생성자는 double형 인자를 하나 받으며,  
new를 이용해 memberB를 할당하고 할당된 공간에 인자로 받은 실수를 저장한 후  
"new memberB"라는 문자열을 출력한다.  
그리고 memberA를 초기화하기 위해 A 클래스의 생성자를 initialization list에서 호출해야  
하며, 인자로 1을 넘긴다.
- C 클래스의 생성자는 std::string형 인자를 하나 받으며,  
new를 이용해 memberC를 할당하고 할당된 공간에 인자로 받은 문자열을 저장한 후  
"new memberC"라는 문자열을 출력한다.  
그리고 memberA와 memberB를 초기화하기 위해 B 클래스의 생성자를 initialization list  
에서 호출해야 하며, 인자로 1.1을 넘긴다.
- A 클래스의 소멸자(destructor)는 delete를 이용해 memberA를 해제하고,  
"delete memberA"를 출력한다.
- B 클래스의 소멸자(destructor)는 delete를 이용해 memberB를 해제하고,  
"delete memberB"를 출력한다.
- C 클래스의 소멸자(destructor)는 delete를 이용해 memberC를 해제하고,  
"delete memberC"를 출력한다.

- A, B, C 모두 void print()라는 멤버 함수를 가지며,  
A::print()는 memberA가 가리키는 공간에 저장된 데이터를 출력하고,  
B::print()는 A::print()를 호출한 후 memberB가 가리키는 공간에 저장된 데이터를 출력하  
며,  
C::print()는 B::print()를 호출한 후 member가 가리키는 공간에 저장된 데이터를 출력한  
다. 출력되는 형식은 아래 실행 예를 참고하도록 한다.
- 사용자로부터 정수, 실수, 문자열을 하나씩 입력 받아, std::vector<A\*> 타입의 벡터  
objects에 A, B, C 클래스의 객체를 각각 1개씩 new로 생성하여 넣을 때의 인자로 넣는다.  
그리고 for문을 이용해 objects의 각 요소의 print() 함수를 호출한다.
- 사용이 끝난 objects의 각 요소는 할당 받은 메모리를 해제해야 하며, 프로그램 코드 전체  
에 걸쳐 type casting 연산자는 사용하면 안 된다.

파일명 : print\_member (print\_member.h print\_member.cc print\_member\_main.cc)

입력: 없음

출력: 과제에서 요구하는 문자열 출력

```
$ ./print_member
20 3.14 test
new memberA
new memberA
new memberB
new memberA
new memberB
new memberC
*memberA 20
*memberA 1
*memberB 3.14
*memberA 1
*memberB 1.1
*memberC test
delete memberA
delete memberB
delete memberA
delete memberC
delete memberB
delete memberA
$
```