



🏠 / C++프로그래밍과실습 (CB3500572-062) / 실습 098 - virtual & destructor / 개요

개요

제출

편집

코딩 결과

실습 098 - virtual & destructor

제출 마감일: 2023-06-01 23:59

업로드 가능한 파일 수: 5

제출 방식: 개인

목적

이 실습은 클래스의 virtual destructor 를 사용하는 연습을 합니다.

설명

base 클래스의 소멸자는 public 이면서 virtual 이여야 합니다.

이유는 다음과 같은 코드가 있을 때 문제가 발생할 수 있습니다.

```
{
    class Base { };

    class Derived : public Base { };

    Base* base = new Base();    ---- (나)
}    ---- (가)
```

(가) 블록이 종료되어 내부에 선언된 변수들이 정리될 때, (나) 기본(base) 클래스 타입의 포인터를 통해 파생(derived) 클래스의 인스턴스를 소멸시키려고 시도 합니다. 이때 기본 클래스의 소멸자가 가상 (virtual) 이 아니면 그 결과는 undefined behavior (UB) 입니다.

virtual 멤버 함수를 가진 클래스 역시 virtual 소멸자를 가져야 합니다.

위 예제 코드를 보면서 그 이유를 한번 생각해 보시기를 권합니다.

참고: C++ Core Guidelines C.35, C.127

문제

주어진 프로그램이 정상적으로 동작하도록 프로그램을 수정하시오.

단, 096에서 구현된 클래스를 수정하여 이용하시오.

<참고>

```
int main() {
    {
        Monitor monitor;
        //monitor.print();
    }
    std::cout << "-----" << std::endl;
    {
        Monitor* pMonitor = new LED_Monitor();
        if (pMonitor) delete pMonitor;
    }
    std::cout << "-----" << std::endl;
    {
        std::unique_ptr<OutputDevice> pMonitor = std::make_unique<LCD_Monitor>();
    }
}
```

입력

없음

출력

OutputDevice constructor

Monitor constructor

Monitor destructor

OutputDevice destructor

OutputDevice constructor

Monitor constructor

LED_Monitor constructor

LED_Monitor destructor

Monitor destructor

OutputDevice destructor

OutputDevice constructor

Monitor constructor

LCD_Monitor constructor

LCD_Monitor destructor

Monitor destructor

OutputDevice destructor

제출파일

LCD_Monitor.h
LED_Monitor.h
OutputDevice.h
Monitor.h

