C++ 퀴즈 (2023-01-05)

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 다음 코드의 출력 값과 그 이유는?

Base 생성자 호출

Derived 생성자 호출

Base 소멸자 호출

이유: a는 Derived 클래스 안의 base 객체를 가리키는 포인터로 base 클래스의 정보만 가지고 있으므로 delete시 base의 있는 소멸자만 호출된다.

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 다음 코드의 출력 값과 그 이유는?

Base 생성자 호출

2

이유: 우선 Base 클래스를 동적할당하여 생성해줬으므로 base 생성자 호출이 나오고 static\_cast를 사용하여 다운 캐스팅이 일어났지만 pb는 base의 정보를 가지고 있으므로 a를 가리키면 2가 출력이 된다.

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 다음 코드의 출력 값과 그 이유는?

*(reinterpret\_cast는 관련 없는 포인터끼리의 강제적인 형 변환을 가능하게 한다. )*

*이상한 값*

*포인터 클래스 를 일반 클래스로 형변환 시켰기 때문에 형변환이 일어날 때 주소값이 넘어가는게 아니라 nullptr이 넘어가서 일반 클래스에서는 이 값을 인식하지 못하기 때문에 이상한 값이 들어간다?..*

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 다음 코드의 문제점에 대해 서술하시오.

a[0]가 가리키고 있는 것은 주소 값이므로 일반 클래스 변수 b를 대입하는 것은 아무 의미가 없다.

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 다시 한번 이 문제에 대해 생각해보자.

블로그에선 직원 데이터와 매니저 데이터를 관리하기 위해 이중 포인터를 선언하였고, 이를 new를 통해 동적 할당하여 Employee\*의 배열과 Manager\*의 배열을 생성하였다. 그리고 이 배열을 다시 동적 할당하여 동적 할당된 지역에 Employee와 Manager를 선언하였는데, 반드시 이럴 필요가 있을지 생각해 보자.

다시 말하자면, Employee\*만 선언하여 먼저 배열을 동적 할당하고, 여기에 Employee를 대입하는 방법으로 블로그에서 구현했던 기능을 완전히 같게 구현할 수 있지 않을까?

Employee 클래스의 멤버 함수인 print\_info 와 calculate\_pay 함수의 virtual를 붙여주고 Employee\*만 선언하여 배열로 Employee와 Manager를 각각 할당하면 main에서 호출시에 런타임 시 컴퓨터에서 동적 바인딩이 일어나 Employee와 Manager는 각각 자신의 클래스의 함수들을 호출할 수 있게 된다.