

문제 정의

기존의 1번에서 만들었던 코드에서 복사 생성자를 제거하고 이를 대체할 무언가의 코드가 첨가되거나 삭제되어야한다.

문제 해결 방법

바로 복사 생성자를 제거해버린다면 countPass 함수에서 Dept 객체 복사가 얇은 복사로 간주되어 이렇게 되면 같은 scores 메모리에 접근한다. 이렇게 되면 countPass 함수가 끝나고 소멸이 될때 scores도 같이 소멸이 되면서 더이상 존재하지 않는 메모리에 접근하여 오류가 생긴다. 이를 해결하기 위해서 참조를 이용한다.

아이디어 평가

헤더파일과 cpp 파일에서의 int countPass(Dept& dept) 처럼 DEpt 객체에 &을 붙여 참조를 하면 객체를 복사하지 않고 원본 객체에 접근을 하게 된다. 이렇게 되면 복사 생성자를 제거해도 오류를 발생시키지 않게 된다.

알고리즘 설명

앞서 풀이했던 1번 문제와 비슷하지만 countPass 함수는 Dept 객체를 참조 받아 원본 객체와 동일한 메모리를 공유한다. Dept 객체가 소멸이 되기 전까지 참조는 유효하여 정상적으로 코드가 수행이 된다.