C++ 퀴즈 (2023-01-03)

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 다음 코드의 출력 값과 그 이유는?

10

A = a.operator<<(b);

Temp.m\_data = 5 + 5; -> temp.m\_data = 10;

Operator<< 내부의 덧셈은 상수끼리의 덧셈이므로 5+5 = 10으로 처리

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 다음 코드에서 ++a와 a++ 중 더 느린 코드와 그 이유를 서술하시오. (컴파일러의 최적화 등은 고려하지 않는다.)

또한 해당 코드에 잘못된 점이 있다면, 그 역시 서술하시오.

++a가 느림 내부에서 임시객체를 만들고 복사본을 리턴하기 때문

전위 연산과 후위 연산의 방식이 뒤바뀌어 있음

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 블로그에서 제시된 operator<<의 구현은 friend 키워드의 이해를 돕기 위해 friend를 사용하였지만, 이는 클래스 내부 변수에 직접 접근하기 때문에 객체 지향 프로그래밍의 캡슐화 원칙을 해친다는 문제도 있다.

friend 키워드 없이 그림의 코드가 실행되게 하는 operator<<를 구현하시오.

ostream& operator<<(ostream& os, const MyInt& a)

{

a.printData(os);

return os;

}

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 다음 코드가 컴파일되지 않는 이유와 해결 방법을 서술하시오.

[] 연산자는 값을 리턴함, 값에는 값을 할당할 수 없으므로 에러

리턴값을 int&로 변환하여 원본을 리턴하면 해결됨

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 다음과 같은 N차원 배열을 오버로딩 하기 위하여, 오늘 배운 내용만으로 해결하기 어려운 부분이 무엇일지, 어떻게 해야 operator[]를 이용해 N차원 배열을 구현할 수 있을지 생각해 보자.

(힌트: operator[]는 앞부터 계산된다. 즉 a[0]부터 계산된다.)

[] 연산자가 MyInts 객체를 반환해야 n차원 [] 연산자를 다시 호출할 수 있음

[]를 호출할 때 현재가 몇 번째 차원의 몇 번째 원소인지를 알 수 있어야 함

마지막 차원에선 int& 를 리턴해서 값을 바꿀 수 있어야 함