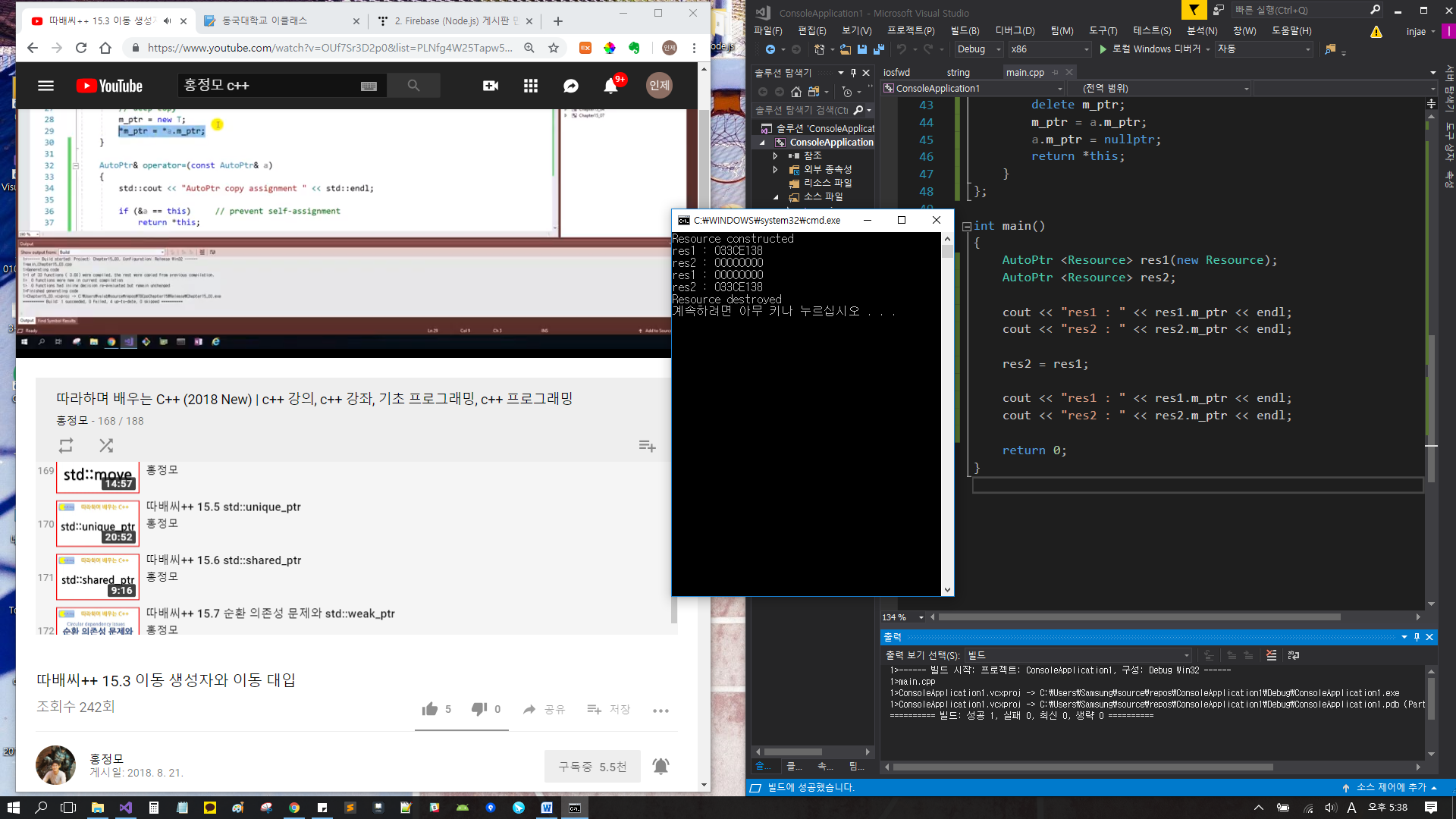
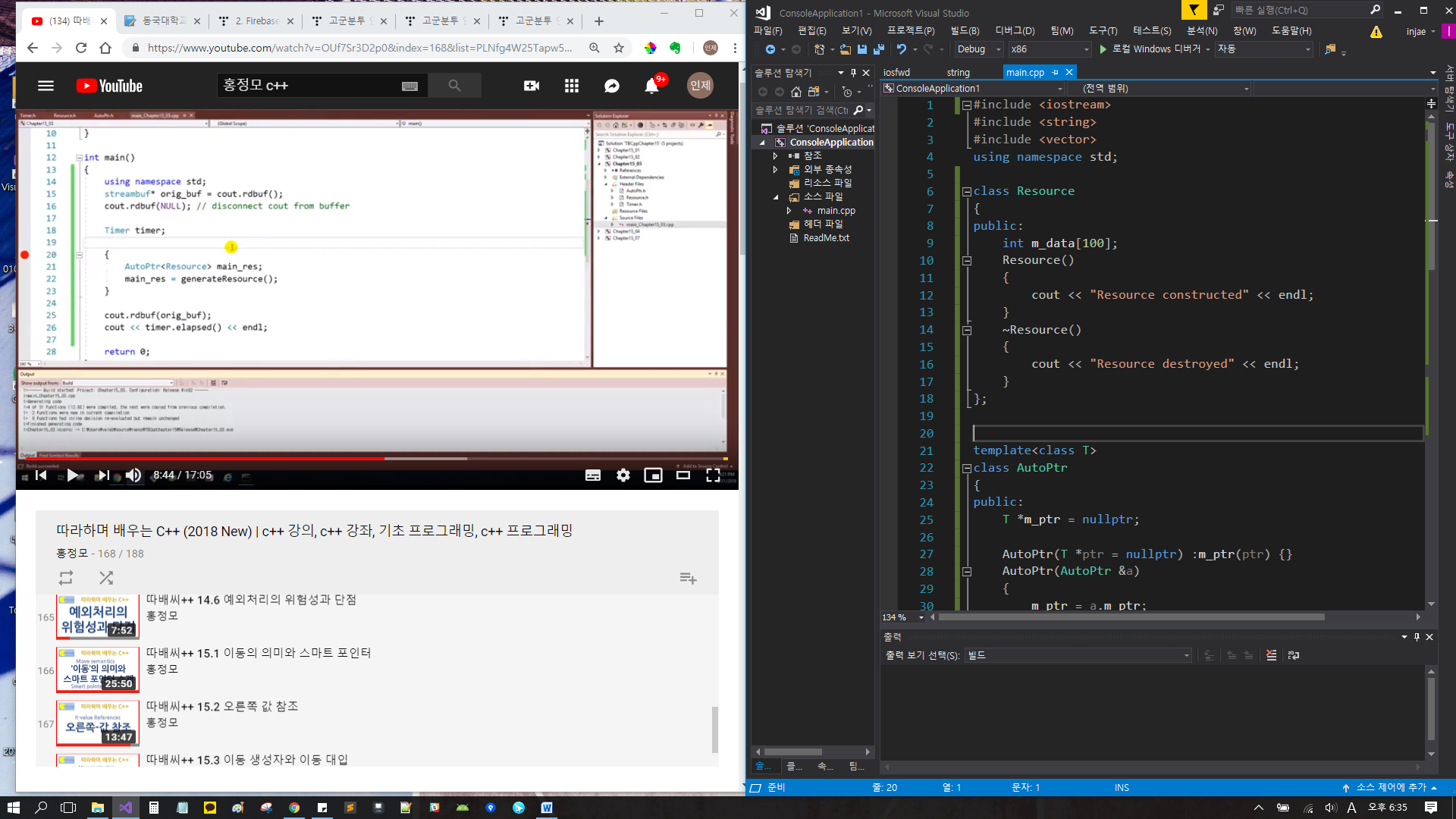
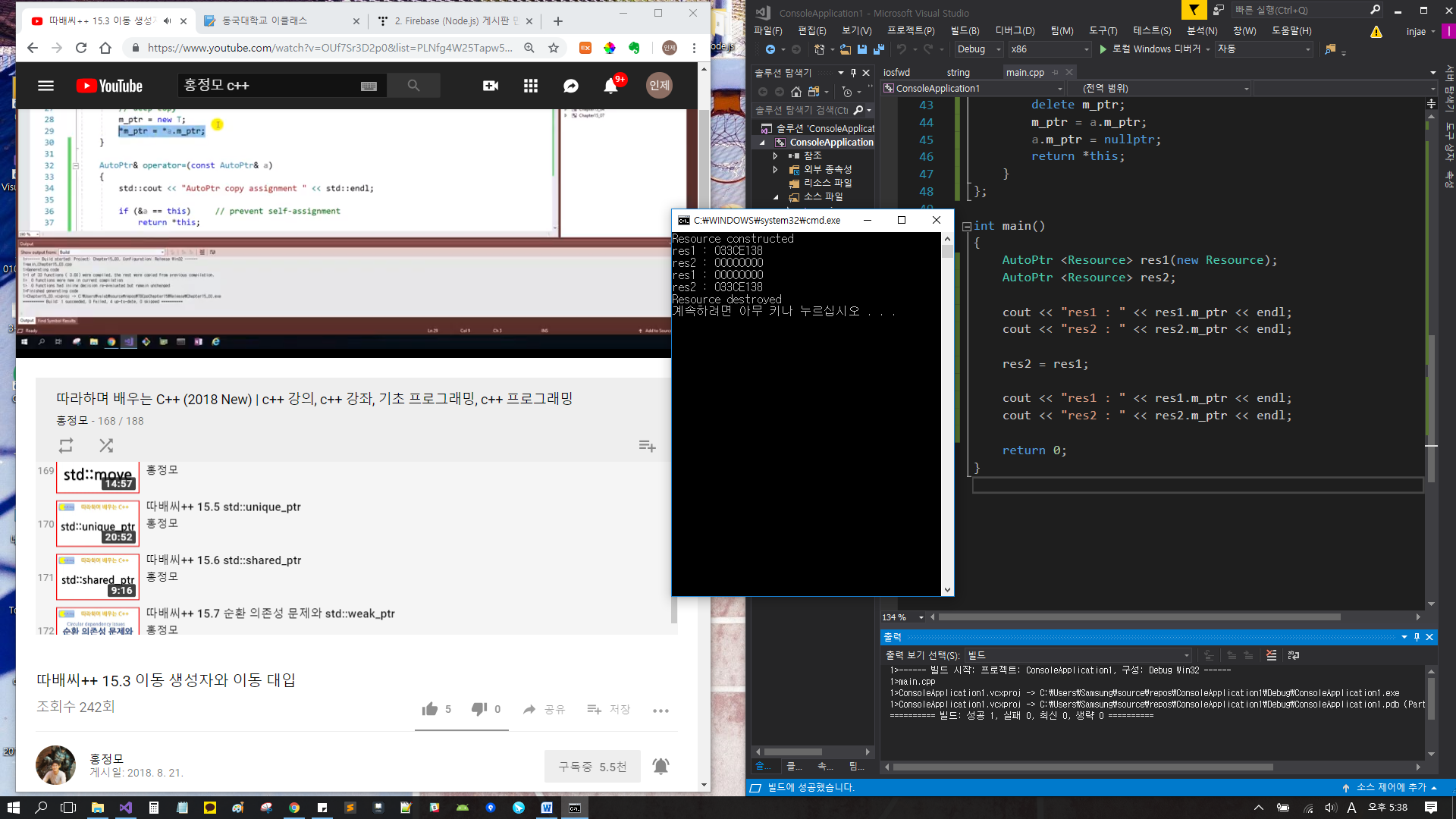
**2018 Autumn 심화프로그래밍 Week #14.**

**학번: 이름:**

1. 메모리 관리를 자동으로 해주는 AutoPtr 클래스를 구현하려고 한다. 밑의 사진과 조건을 참고하여 AutoPtr 클래스를 완성하시오. (30 pt)

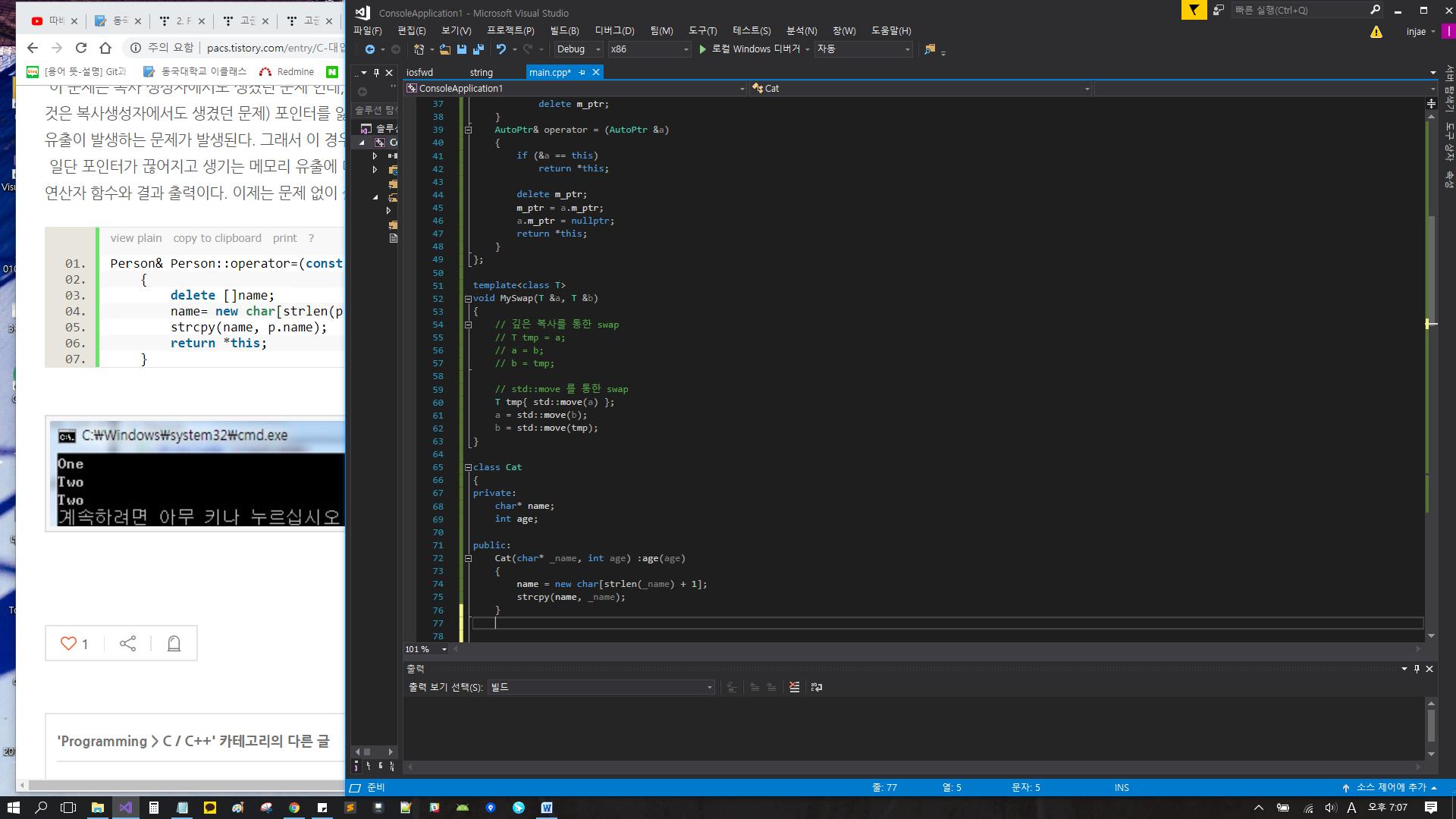
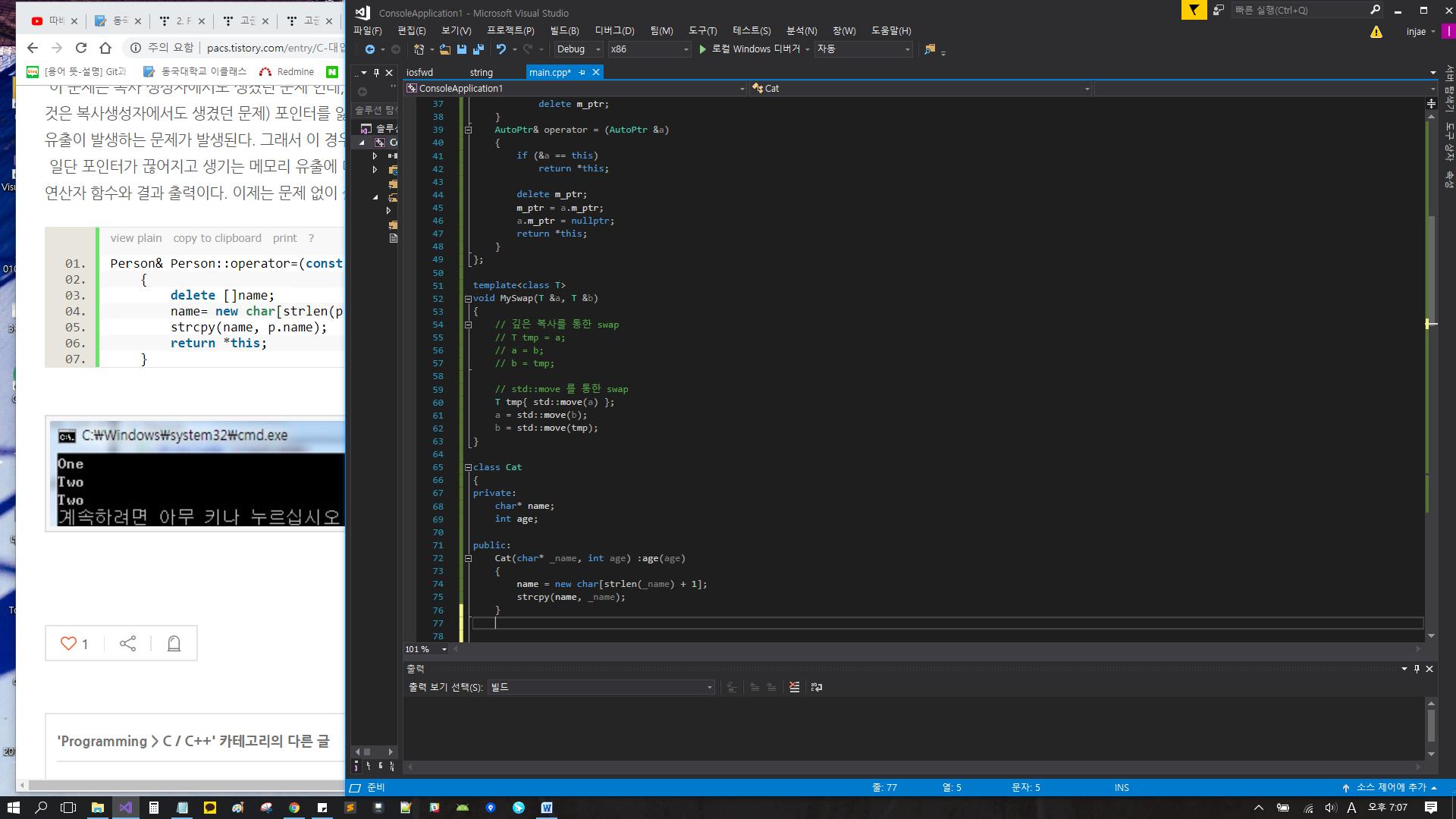


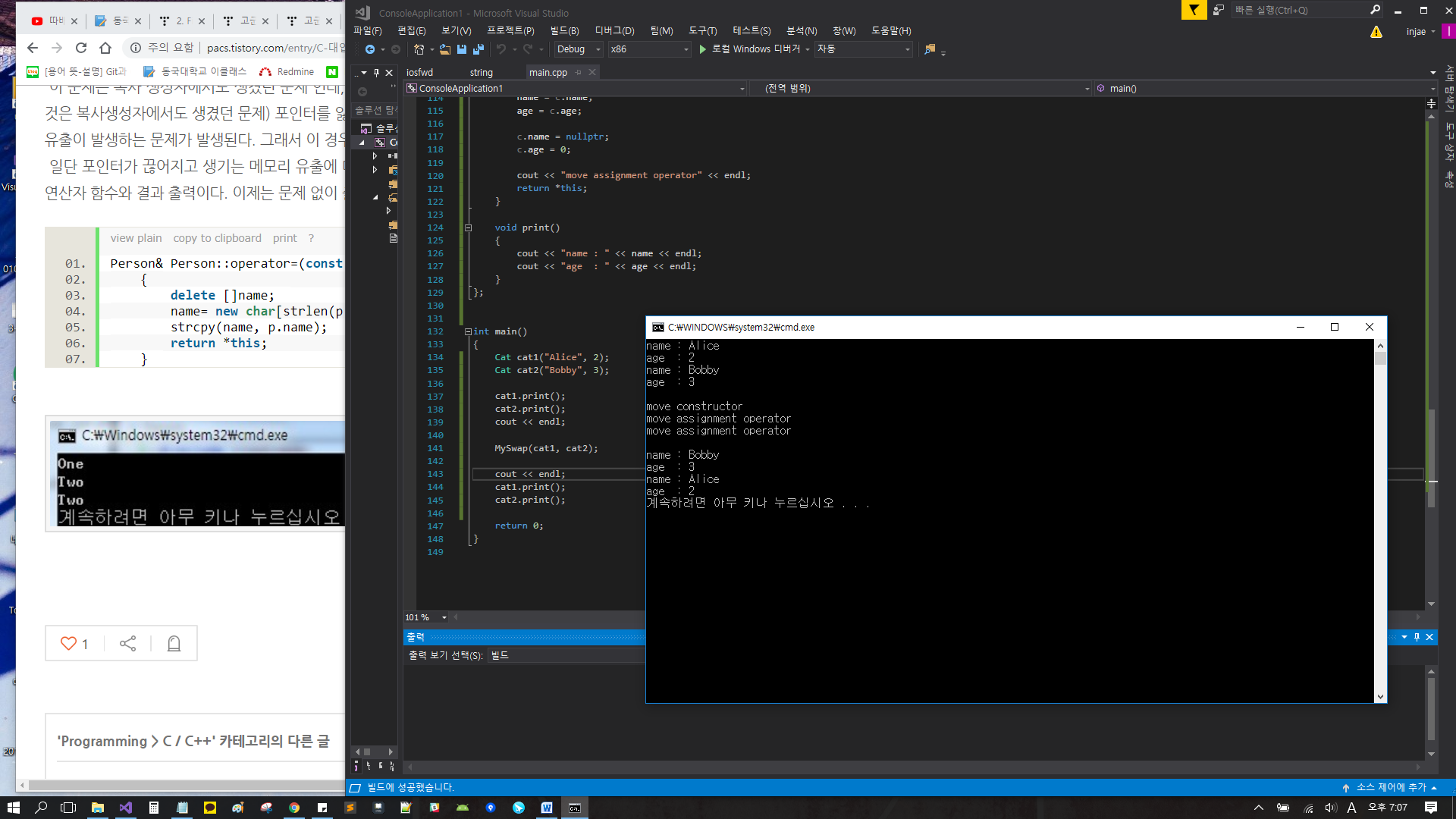
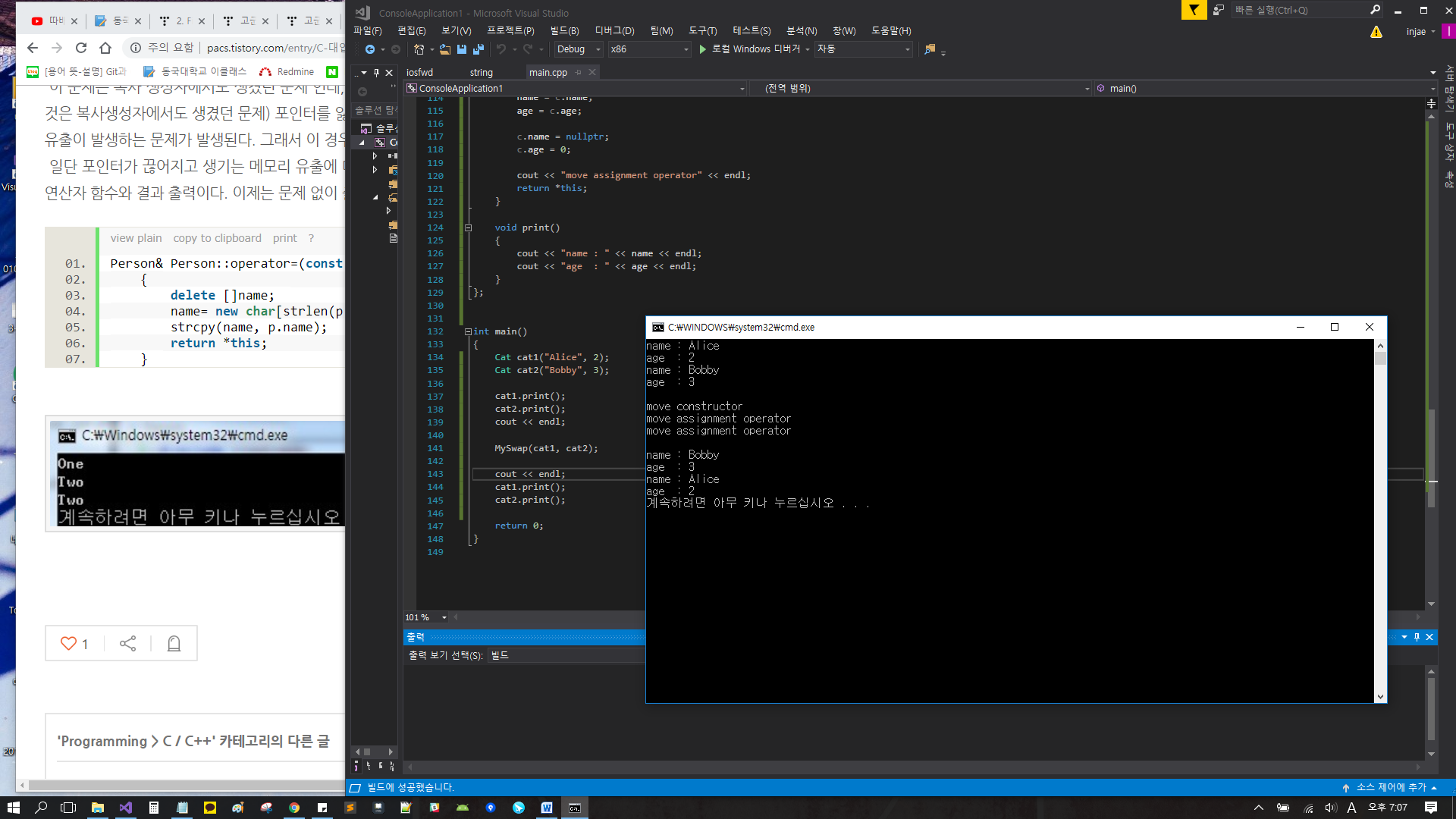
**조건 1)** Resource 클래스의 소멸을 자동적으로 담당해주는 AutoPtr 클래스를 구현하라.

**조건 2)** AutoPtr 의 대입 연산자 오버로딩을 move semantic으로 구현하여 위의 결과가 나오도록 AutoPtr 클래스를 완성하라.

1. 클래스에서 깊은 복사를 수행하는 복사 생성자 나 대입 연산자는 깊은 복사로 인한 과도한 연산을 수행할 가능성이 있다. 이를 해결하기 위하여 이동 생성자와 이동 대입을 구현하고 밑의 프로그램이 정상적으로 작동하도록 수정하라. (40 pt)

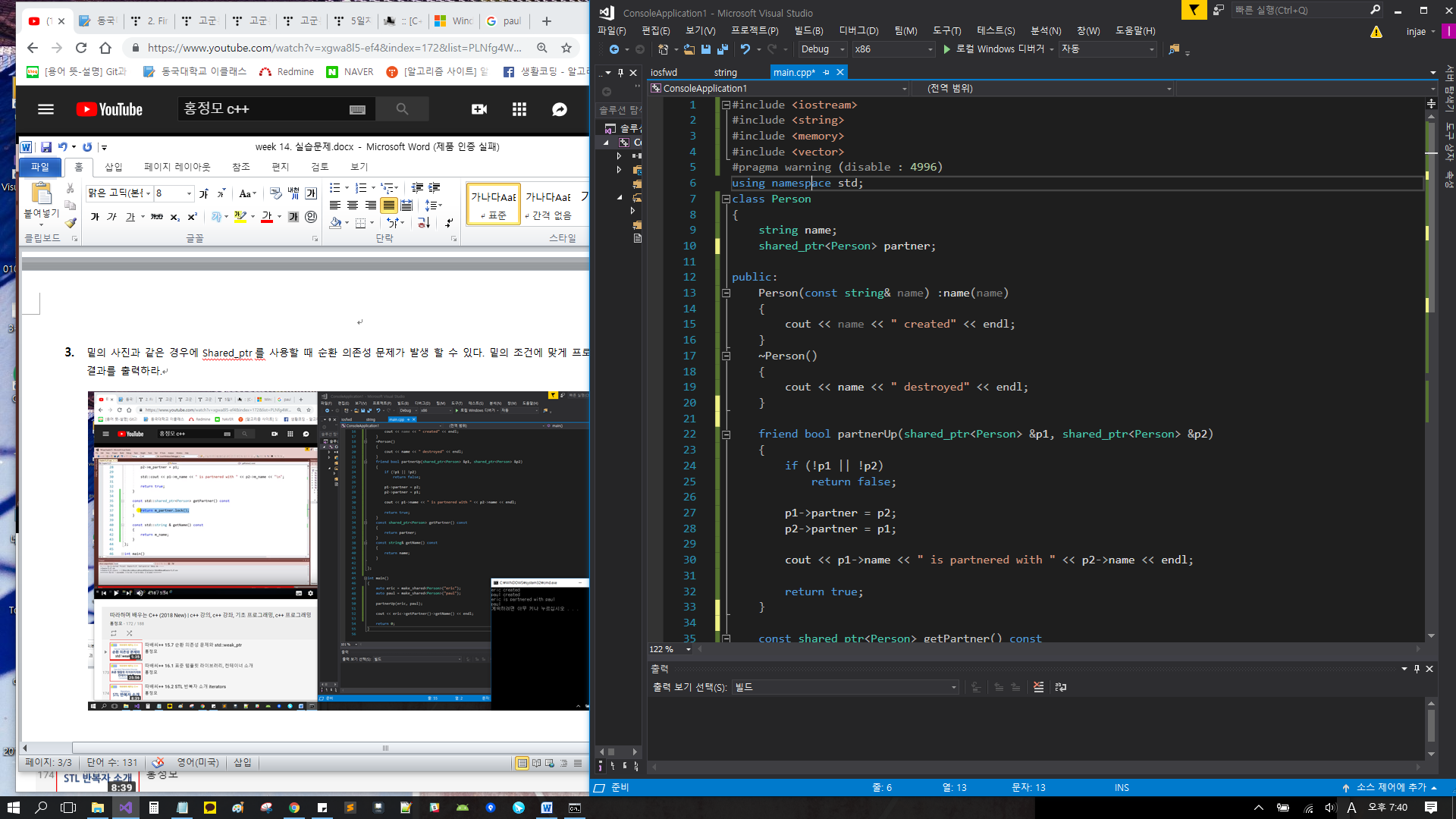
**Cat 클래스 와 MySwap() 함수**

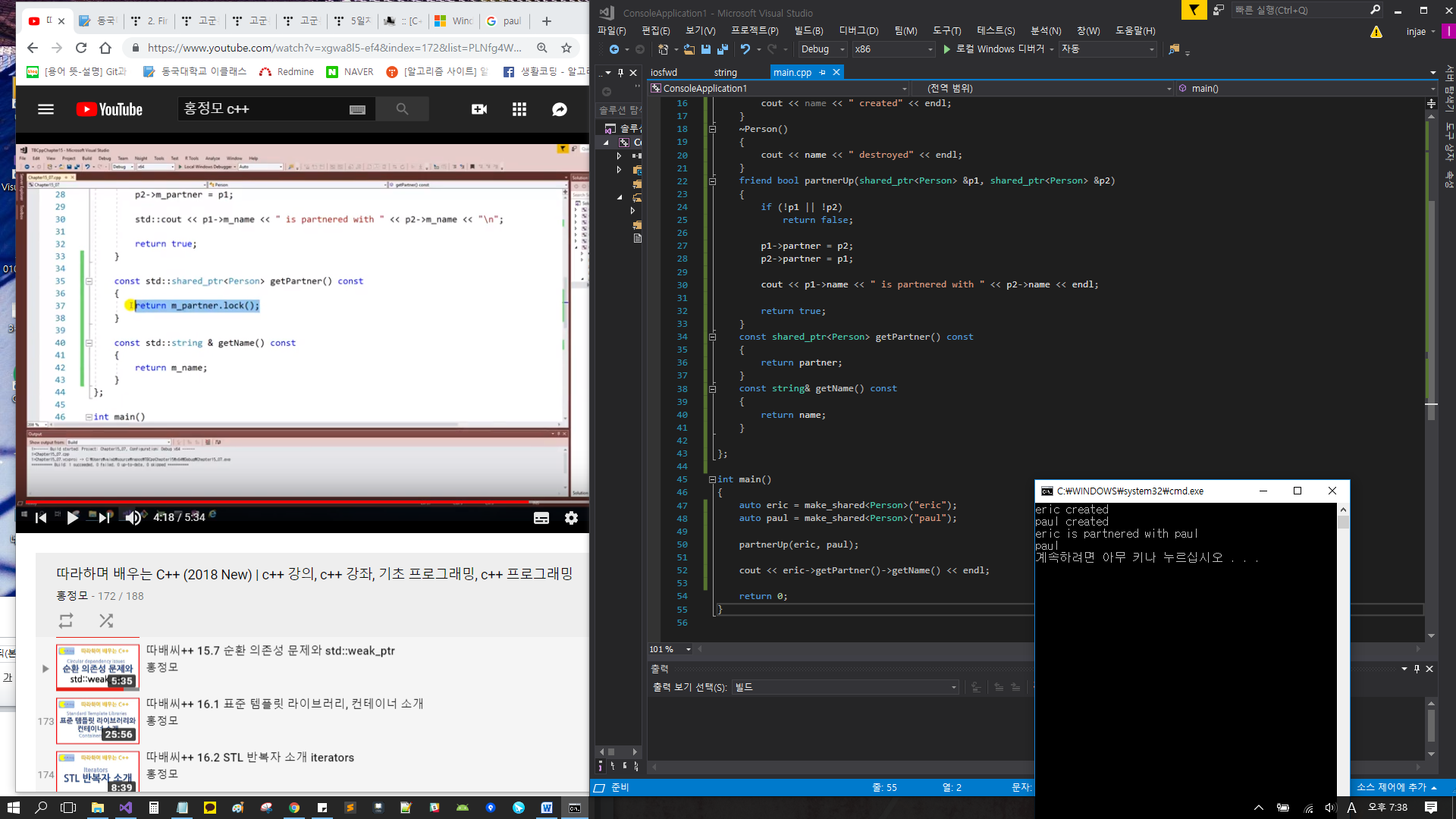
 

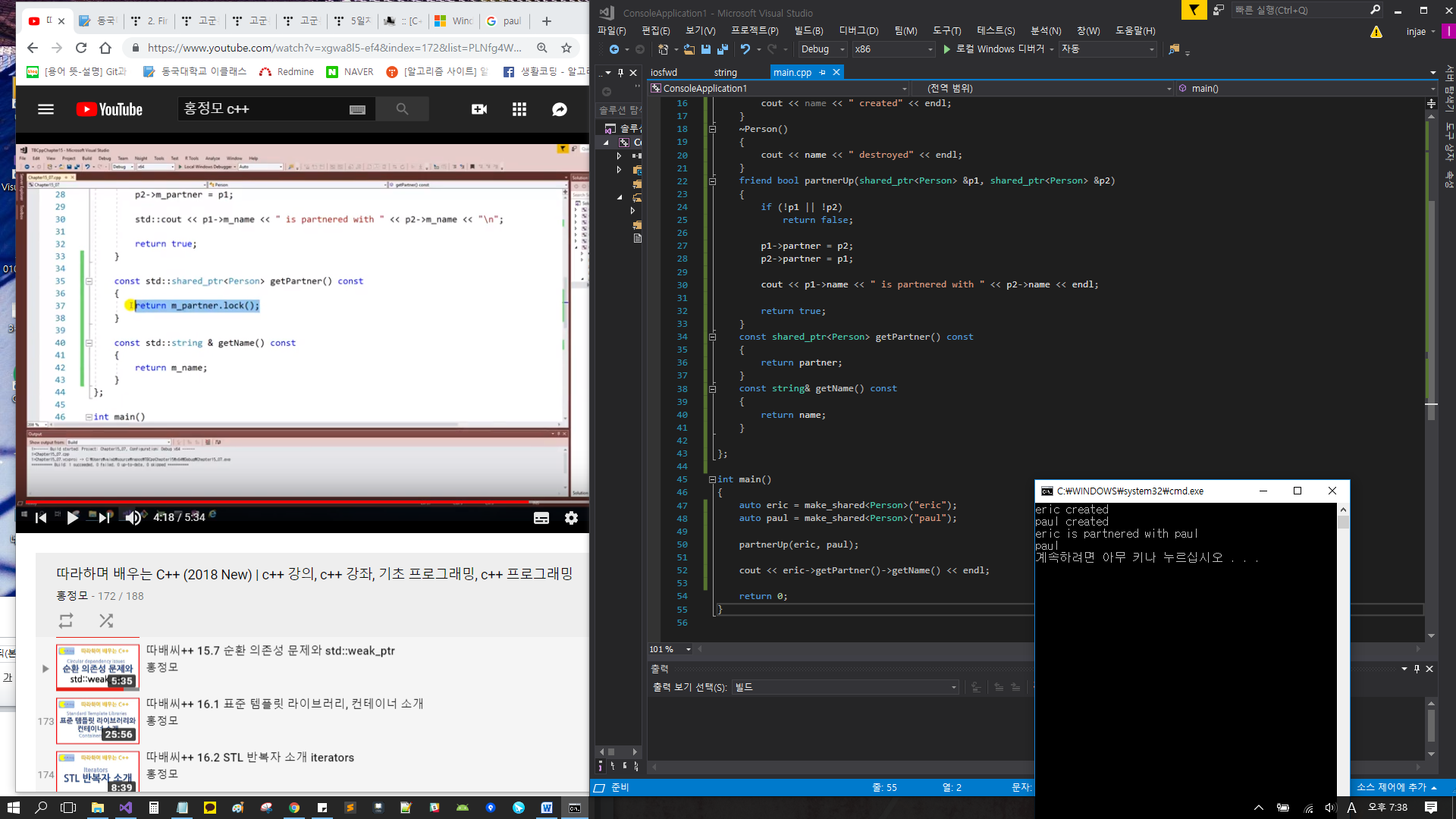
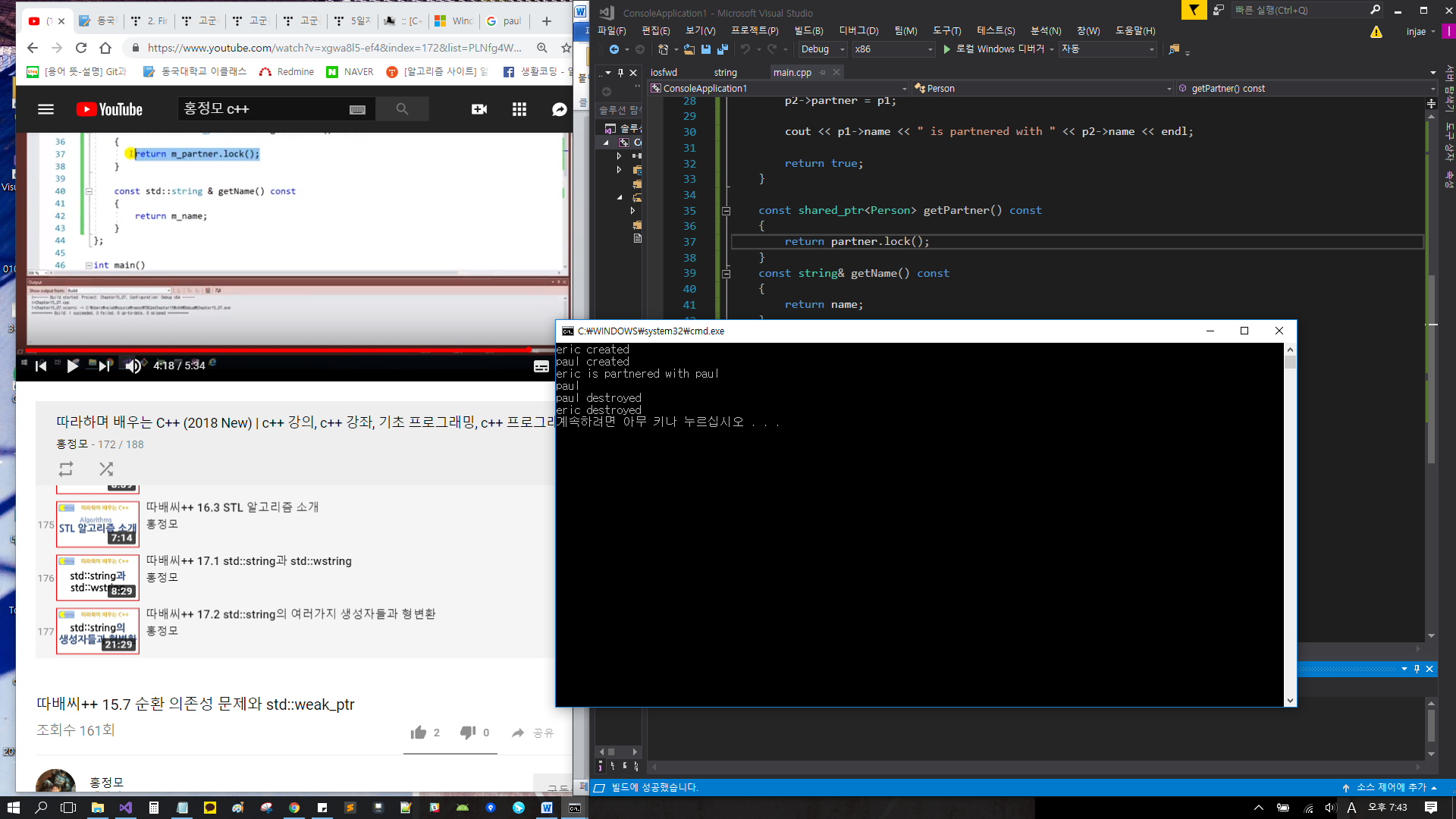
**Main() 과 실행 결과**

**조건 1)** Cat 클래스의 이동 생성자, 이동 대입을 구현하고 MySwap() 의 실행결과가 위와 같도록 구현하라.

1. 밑의 사진과 같은 경우에 shared\_ptr 를 사용할 때 순환 의존성 문제가 발생 할 수 있다. 밑의 조건에 맞게 프로그램을 수정하고 결과를 출력하라. (30 pt)





**좌)** 순환 의존성이 발생 했을 때 의 출력결과, **우)** 순환 의존성을 해결한 후 정상적인 출력 결과.

**조건 1)** shared\_ptr를 사용한 위의 상황에서 순환 의존성 문제가 발생한 이유를 간단히 설명하시오.

**조건 2)** 위의 문제에서 순환 의존성을 해결하고 위의 올바른 결과가 출력되도록 프로그램을 수정하시오.