항목 28: 내부에서 사용하는 객체에 대한 ‘핸들’을 반환하는 코드는 되도록 피하자

**핸들 = 다른 객체에 손을 댈 수 있게 하는 매개자**

private으로 선언된 변수가 있습니다. 그런데 만약 이 변수의 참조자를 반환하는 함수가 있을 때 어떤 문제가 발생할까요? 다음 예제를 보도록 하겠습니다.

*class Point{*

*private:*

*int x;*

*public:*

*int& getX( ){ return x; }*

*}*

**외부에서 private으로 선언된 x의 값을 수정 할 수 있을까요? 네 그렇습니다. 왜냐하면 getX함수가 참조자(핸들)을 반환 하기 때문입니다.** “getX( ) = 값” 이렇게 바꿀 수 있습니다. 굳이 private으로 선언한 이유가 없는 것이고 캡슐화 되어 있다고 할 수 없습니다.

해결방법으로 앞에 const를 붙이면 일단 값의 변경은 막을 수 있습니다.

**cons tint& getX( ){ return x; } 이렇습니다.**

그러나 아직 해결되지 않은 문제가 있습니다. 그것은 “무효참조 핸들”이 발생 할 수 있습니다.

**“무효참조 핸들” = 핸들이 있기는 하지만 그 핸들을 따라갔을 때 실제 객체의 데이터는 없는 것.**

예제를 통해 알아봅시다.

*class Point*

*{*

*public:*

*int& getX( ) { return x; }*

*private:*

*int x;*

*}*

*// non-member함수*

*int\* func( )*

*{*

*Point p; // Point 객체 생성*

*return &p.getX( ); // p객체의 private 맴버 x의 주소를 반환*

*}// 함수가 끝나고 p객체 소멸*

*int\* handleX = func( ); // Point 객체의 member변수 x의 핸들을 받음.*

getX 함수는 참조자를 리턴 하고 func 함수는 포인터를 리턴 하기 때문에 어느 곳에서도 복사가 일어 나지 않습니다. 복사가 없으니 빠르긴 **하겠지만 func함수 내에서 선언된 p객체는 함수 종료 후에 소멸 되기 때문에 handleX포인터 가리키는 객체는 소멸되고 없습니다.**

**핸들은 객체의 값을 변화시킬 수 있는 것이면 무엇이든(참조자, 포인터, 반복자) 상관 없습니다. 핸들을 public함수로 반환 하는 순간 그 핸들에 영향을 받는 모든 변수들의 접근권한은 public로 바뀐 것 입니다.**