

# 5장 상속으로 코드 재사용

변영철 교수  
([ycb@jejunu.ac.kr](mailto:ycb@jejunu.ac.kr))

# 1.Console3 프로젝트 생성

- CPoint 클래스 -> gildong 객체

```
#include <stdio.h>

class CPoint //클래스는 뭐 하라고 있는 것?
{
private:
    int m_iX;
    int m_iY;


public:
    void Assign(int x, int y);
    int Add();
};

void CPoint::Assign(int x, int y)
{
    m_iX = x;
    m_iY = y;
}

int CPoint::Add()
{
    return m_iX + m_iY;
}

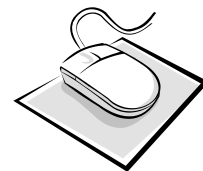
void main()
{
    CPoint gildong; //객체 만들라고 있는 것!
    gildong.Assign(2, 3);
    printf("두 개의 값을 더한 결과: %d\n", gildong.Add());
}
```

## 2.귀찮은 건 정말 싫은데..

- 여러분, **CCircle** 클래스 작성할 수 있나요?  
→ youngSu 객체
  - 원은 중심점(m\_iX, m\_iY)과 반지름(m\_iRadius)을 가짐
  - 원 객체가 할 줄 아는 일은? Assign, Add,  SetRadius, Area, ...
- gildong과 youngSu는 **완전히 별개**의 객체
- CPoint와 CCircle 코드가 중복
  - 코드를 다시 입력하는 것이 귀찮으면 재사용

### 3.관계는 없다

- 상속을 통한 코드 재사용
  - gildong과 youngSu는 여전히 완전히 별개 객체로서 아무런 관계도 없다.
  - 단지 동일한 코드를 다시 입력하는 것이 귀찮아서 재사용(상속)을 했을 뿐
  - 상속은 관계를 만들기 위한 것이 아님

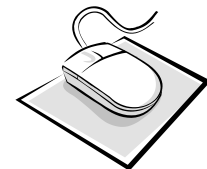


## 4.눈에 안 보이는 멤버

- CCircle 클래스를 보면
  - 눈에 보이는 멤버만이 멤버가 아니다.
  - 보이는 않는 멤버(변수, 함수)도 많이 있을 수 있다.

## 5.상속받은 멤버 중 private 멤버

- 원의 중심점을 옮기는 함수 Move 작성
- private 멤버는
  - 오직 해당 클래스 안에서나 접근 가능
  - CCircle 클래스에서 CPoint 클래스의 private 멤버 접근 불가
- protected 멤버 변수는
  - 외부 클래스에서는 접근할 수 없고 **코드를 재 사용하는 클래스에서는 접근할 수 있도록 보호된 멤버**



## 6.아마 상속받고 있을 꺼야

- 다음 코드에 오류가 없다면

```
void CCircle::Move(int iX, int iY)
{
    m_iX = m_iX + iX;
    m_iY = m_iX + iY;
}
```



- Visual C++ 프로그램 코드의 예
  - InitInstance 멤버 함수

# 7.함수 재정의

- CEllipse 클래스 작성
  - 원 클래스에 반지름 하나 더 추가
- 면적을 출력하려면
  - 재사용하는 코드 Area 멤버 함수
- 함수 재정의(overriding)
  - 상속 받은 함수가 쓸모 없을 때
  - 상속 받은 함수가 불충분할 때

