

friend

# friend란?

- ▶ 접근 지정자를 완벽히 무시할 수 있는 예외적인 기능을 가졌다
- ▶ 객체 지향의 정보 은닉 개념에 위배된다
- ▶ friend로 선언된 대상에게 **private**, **protected**는 **public**으로 작용한다
- ▶ **friend**는 사용하는 쪽이 아닌 사용되는 쪽에 선언된다

# friend 클래스

```
class assailant;
class victim //피해자
{
private:
    unsigned int money;
public:
    victim():money(10000) {}
    void confirm() { cout << "victim 소지금:" << this->money << endl; }
    friend assailant;
};

class assailant //가해자
{
private:
    unsigned int money;
public:
    assailant() :money(0) {}
    void extort(victim& person)
    {
        this->money += person.money;
        person.money = 0;
    }
    void confirm() { cout << "assailant 소지금:" << this->money << endl; }
};
```

# friend 멤버 함수

```
class victim;
class assailant //가해자
{
private:
    unsigned int money;
public:
    assailant() :money(0) {}
    void steal(victim& person); //훔친다
    void confirm() { cout << "assailant 소지금:" << this->money << endl; }
};

class victim //피해자
{
private:
    unsigned int money;
    unsigned int be_robbed() //도둑맞는다
    {
        unsigned int temp = this->money;
        this->money = 0;
        return temp;
    }
public:
    victim():money(10000) {}
    void confirm() { cout << "victim 소지금:" << this->money << endl; }
    friend void assailant::steal(victim& person);
};

void assailant::steal(victim& person) { this->money += person.be_robbed(); }
```

# friend 전역 함수

```
class victim
{
public:
    victim(int, int, int);
private:
    int _b, _w, _h;
    void My3Size() { cout << "B:" << this->_b << " W:" << this->_w << " H:" << this->_h; }
    friend void hacked(victim& person)
    {
        cout << "비밀이 털렸다!!" << endl;
        person.My3Size();
    }
};

victim::victim(int b, int w, int h) :_b(b), _w(w), _h(h) {}

int main()
{
    victim person(1, 2, 3);
    hacked(person);
}
```