

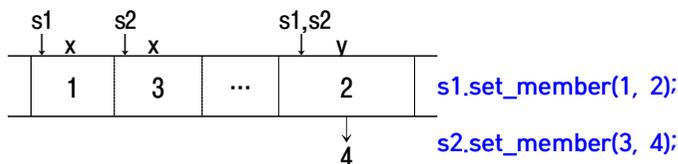
3. 정적멤버(static member)

정적멤버(static member) 변수는 하나의 클래스에 속한 모든 객체들이 공유한다.

```
#include <iostream.h>
class StaticMember{
private:
    int x;           //일반멤버변수 : 일반멤버함수로만 접근할 수 있다.
    static int y;   //정적멤버변수 : 일반 또는 정적멤버함수로 접근할 수 있다.
public :
    void set(int a, int b){ x = a; y = b; } //일반멤버함수
    int get(){ return x; }                //일반멤버함수
    static int sget(){ return y; }        //정적멤버함수 : 정적멤버변수만 접근 가능
};

int StaticMember::y; //정적멤버변수는 미리 이런 형식으로 선언 또는 초기화해야 한다.
//정적멤버변수는 선언시 '클래스명::'을 변수 앞에 붙인다.

void main()
{
    StaticMember s1, s2;
    s1.set(1, 2);
    s2.set(3, 4);
    cout<<s1.get()<<" "<<s1.sget()<<endl; //출력 : 1 4
    cout<<s2.get()<<" "<<s2.sget()<<endl; //출력 : 3 4
}
```



- 정적멤버 변수는 그 클래스의 객체가 하나도 생성되지 않아도 사용 가능하다.
- 정적멤버 함수는 정적멤버 변수만 접근할 수 있다.
- 일반멤버 함수는 모든 멤버 변수를 접근할 수 있다.

기출문제 분석

1. 다음 C++ 프로그램의 실행 결과로 옳은 것은? [2021년 지방 9급]

```
#include <iostream>
using namespace std;
class Student {
public:
    Student():Student(0) {};
    Student(int id):_id(id) { if (_id > 0) _cnt++; };
    static void print() { cout << _cnt; };
    void printID() { cout << ++_id; };
private:
    int _id;
    static int _cnt;
};
int Student::_cnt = 0;
int main() {
    Student A(2);
    Student B;
    Student C(4);
    Student D(-5);
    Student E;
    Student::print();           // 출력 : 2
    E.printID();               // 출력 : 1
    return 0;
}
```

- ① 21 ② 22 ③ 30 ④ 31

☞ C++ 프로그램 - 정적멤버변수가 있는 C++ 프로그램

- 먼저, 주어진 프로그램에서 헛갈리는 코드 분석

```
Student(int id):_id(id) { if (_id > 0) _cnt++; };
```

↓아래 코드와 같다.

```
Student(int id) {
    _id = id;           //속성 _id 값을 매개변수 id 값으로 설정
    if (_id > 0) _cnt++; //속성 _id>0 이면, 정적 속성 _cnt를 1 증가
}
```

```
// 프로그램 분석 - 속성 _id가 0보다 크면, 정적 속성 _cnt를 1 증가
#include <iostream>
using namespace std; //std라는 이름 공간에 정의된 모든 것을 사용하겠다는 의미
class Student { //클래스 Student
public:
    Student():Student(0) {}; //기본생성자 Student()에서 아래 정의된 Student(0)을 호출
    Student(int id): _id(id) { if (_id > 0) _cnt++; }; //속성 _id>0 이면, 정적 속성 _cnt를 1 증가
        //속성 _id 값을 매개변수 id 값으로 설정
    static void print() { cout << _cnt; }; // _cnt 값을 출력
    void printID() { cout << ++_id; }; // _id 값을 1 증가시켜서 출력
private:
    int _id; //속성 _id
    static int _cnt; //정적 속성 _cnt, 정적 속성은 모든 객체가 공유함
};
```

```
int Student::_cnt = 0; //클래스 Student의 정적 속성 _cnt = 0으로 초기화
//정적 속성은 반드시 이런 형식으로 초기화해야 한다.(C++에서는 중요한 것!)
```

```
int main() {
    Student A(2); //속성 _id = 2가 되므로, _cnt++이 수행되어 cnt = 1
    Student B; //속성 _id = 0이 되므로, _cnt++이 수행되지 않음
    Student C(4); //속성 _id = 4가 되므로, _cnt++이 수행되어 cnt = 2
    Student D(-5); //속성 _id = -5가 되므로, _cnt++이 수행되지 않음
    Student E; //속성 _id = 0이 되므로, _cnt++이 수행되지 않음
    Student::print(); // _cnt 값을 출력 : 2
    E.printID(); // Student E의 _id 값을 1 증가시켜서 출력 : 1
    return 0;
}
```

// 메모리 구조

A(2) id	B id	C(4) id	D(-5) id	E id	...	cnt
2	0	4	-5	0	...	0
				↓ 1		↓ 1
						↓ 2