

Class Student

Oct 13

Student class 실습1

- Student 클래스는 다음과 같은 private 데이터를 저장할 수 있다.
 - 성 (last name), string type
 - 이름 (first name), string type
 - 학년, integer type
 - 수강과목 리스트, vector(string) type
- Student 클래스는 다음과 같은 public 함수를 제공한다.
 - 생성자는 다음을 인자로 받는다.
 - 성(string), 이름(string)
 - 이 생성자가 불리면 학년을 1로 초기화 한다. 수강과목은 없음.
 - string get_name(): 성+빈칸+이름을 string으로 리턴한다. 예: Shin Minho
 - int get_year(): 학년을 리턴한다
 - void promote(): 학년을 하나 증가한다. 단, 현재 4학년이면 0으로 설정하고, 이미 0이면 증가하지 않는다.
 - bool graduated(): 학년이 0이면 참을 리턴한다. 그 외는 거짓.
 - void enroll(string classname): 주어진 이름의 수업을 수강한다.
 - string get_class(int i): i번째 수강과목을 리턴한다.

Student class 실습1

```
#define check( s, a, b ) { \
    if( a != b ) { \
        cout << "## FAIL (" << s << "): " << a << " != " << b << endl; \
        exit(1); \
    } \
    cout << "## PASS (" << s << "): " << a << endl; \
}
```

Student class 실습1

```
int main() {  
    Student s1( "Kim", "Soochul" );  
    check( s1.get_year(), 1 );  
    check( s1.get_name(), "Kim Soochul" );  
    s1.promote();  
    check( s1.get_year(), 2 );  
    check( s1.graduated(), false );  
    s1.enroll( "C++" );  
    s1.enroll( "Java" );  
    check( s1.get_class(0), "C++" );  
    check( s1.get_class(1), "Java" );  
    s1.promote(); s1.promote(); s1.promote();  
    check( s1.graduated(), true );  
}
```

Student class 실습 2

- Student 클래스는 다음 기능을 추가적으로 제공한다
 1. 학생ID를 저장할 수 있다.
 2. 생성자가 불리면 학생 ID를 랜덤하게 할당한다.
 3. `int get_sid()`: 학생ID를 리턴한다.

Student class 실습 3

- void drop(string classname)이 불리면
 - 수강 과목 리스트에서 해당 클래스를 삭제한다
- int classes_cnt(): 현재 수강중인 수업 갯수

vector에서 삭제

- `vector.erase(X)`
 - X: iterator type
- `vector<T> list;`
- `vector<T>::iterator it;`
- `it = find(list.begin(), list.end(), item);`
- `if(it == list.end()) cout << “not found”;`
- `else list.erase(it);`

Student class 실습 4

- operator ++ (prefix)를 정의하라
 - 학년을 1년 증가시킴
- Student& Student::operator++() {
- }
- operator ++ (postfix)를 정의하라
 - 학년을 1년 증가시킴
- Student Student::operator++(int) {
- }

Student class 실습 5

- operator <, >, ==, !=를 정의하라
 - 학년을 비교한다
- bool operator<(const Student s1, const Student s2)

Student class 실습 6

- operator<<를 정의하여라

```
ostream &operator<<(ostream &out, const  
Student &t) {  
    output<<...  
    return out;  
}
```

Student class 실습 7

- class Department 정의
 - name, professors, students
- class Subject를 정의
 - name, teacher, code, time, students