



🏠 / C++프로그래밍과실습 (CB3500572-062) / 실습 078 - < 연산자 오버로딩

개요

제출

편집

코딩 결과

## 실습 078 - < 연산자 오버로딩

제출 마감일: 2023-05-04 23:59

업로드 가능한 파일 수: 3

제출 방식: 개인

### 목적

이 실습은 class 의 < 연산자 오버로딩을 연습합니다.

### 문제

std::map 컨테이너에 Student 객체를 저장하고 특정 속성 값으로 정렬하는 프로그램이 정상적으로 동작하도록 Student 클래스를 구현 하시오.

#### <참고>

- key는 학생 ID이고 값은 학생 객체입니다. (std::map은 내부적으로 key 값으로 정렬되어 있습니다)
- Student 클래스는 id, name 및 grade의 세 가지 속성이 있습니다.
- comparator function 인 bool compareByGrade(const Student& a, const Student& b) 전역 함수를 구현합니다.
- bool operator<(const Student& other) const; 멤버 함수를 구현합니다.

```
int main() {
    std::map<int, Student> student_grades;

    // Insert 5 student objects into the map
    student_grades[1] = {1, "Bob", 90};
    student_grades[2] = {2, "Alice", 85};
    student_grades[3] = {3, "David", 80};
    student_grades[4] = {4, "Eve", 95};
    student_grades[5] = {5, "Charlie", 88};

    std::cout << "***** In order of their id" << std::endl;
    for (const auto& [id, student] : student_grades) {
        std::cout << student << std::endl;
    }

    std::vector<Student> students;
```

```

// This is because std::sort requires random access iterators,
// but std::map iterators are bidirectional iterators, and this causes the error.

for (const auto& [id, student] : student_grades) {
    students.push_back(student);
}

std::sort(students.begin(), students.end(), compareByGrade);

std::cout << "***** In descending order of their grade" << std::endl;
for (const auto& student : students) {
    std::cout << student << std::endl;
}

std::sort(students.begin(), students.end());

std::cout << "***** In order of their name" << std::endl;
for (const auto& student : students) {
    std::cout << student << std::endl;
}
return 0;
}

```

## 입력

없음

## 출력

```

***** In order of their id

Id: 1 Name: Bob Grade: 90

Id: 2 Name: Alice Grade: 85

Id: 3 Name: David Grade: 80

Id: 4 Name: Eve Grade: 95

Id: 5 Name: Charlie Grade: 88

***** In descending order of their grade

Id: 4 Name: Eve Grade: 95

Id: 1 Name: Bob Grade: 90

Id: 5 Name: Charlie Grade: 88

Id: 2 Name: Alice Grade: 85

Id: 3 Name: David Grade: 80

***** In order of their name

Id: 2 Name: Alice Grade: 85

```

Id: 1 Name: Bob Grade: 90

Id: 5 Name: Charlie Grade: 88

Id: 3 Name: David Grade: 80

Id: 4 Name: Eve Grade: 95

## 제출파일

Student.cpp

Student.h

78.csv