

Homework #4

1. 간단하게 나만의 MyString class를 작성해보자. 멤버 변수로는 문자열을 저장할 char* 타입의 변수와 그 길이를 저장할 int 타입의 변수를 가진다. 문자열 초기화와 출력을 string class와 동일하게 하기 위해 = 연산자와 << 연산자를 오버로딩하여 구현하자.

2. 교과목 점수를 저장하는 Score class를 구현해보자. 과목으로는 수학, 언어, 외국어, 사회가 있다.

수학	언어	외국어	사회
41	85	72	38
80	69	65	68
96	22	49	67
51	61	63	87
66	24	80	83

(1) 구현한 Score class를 가지고 객체 배열을 동적 할당 받은 뒤 위의 정보를 출력하자.

(출력시에는 Score class에 << 연산자 오버로딩을 사용한다.)

출력값:

수학	언어	외국어	사회
41	85	72	38
80	69	65	68
96	22	49	67
51	61	63	87
66	24	80	83

(2) 위 정보를 가지고 각각의 과목의 평균을 구하는 함수를 구현하자.

- 조건 -

1) Score class에 평균을 구하기 위해 필요한 연산자들 (+, /, =)을 오버로딩하여 평균을 구하는 함수를 구현하자.

2) 이항연산자 오버로딩을 사용하여 구하는 함수(getAverageScore_1)와 단일연산자 오버로딩을 사용하여 구하는 함수(getAverageScore_2)를 각각 만들어라.

3) 소수점 이하 숫자는 1개까지만 출력

출력값:

평균				
66.8	52.2	65.8	68.6	// getAverageScore_1() 함수를 이용한 결과
66.8	52.2	65.8	68.6	// getAverageScore_2() 함수를 이용한 결과

3. 위의 Score 클래스를 멤버 변수로 사용하여 Student Class를 구현해보자.

(1) 멤버 변수로 이름은 string class 타입, 점수는 Score class 타입, 추가로 생년월일을 저장할 Birth 구조체를 추가하여 Student Class를 구현하자.

이름	수학	언어	외국어	사회	출생년도	출생월	출생일
김진우	41	85	72	38	2003	02	04
나문희	80	69	65	68	2002	04	05
박명수	96	22	49	67	2002	12	31
장산범	51	61	63	87	2003	01	11
차재우	66	24	80	83	2002	05	25

Student 객체 배열을 동적으로 생성한 뒤에 위 표의 값들을 넣어준 뒤 출력하시오. (<< 연산자 오버로딩 이용)

출력값:

이름	수학	국어	외국어	사회	생년월일
김진우	41	85	72	38	2003-2-4
나문희	80	69	65	68	2002-4-5
박명수	96	22	49	67	2002-12-31
장산범	51	61	63	87	2003-1-11
차재우	66	24	80	83	2002-5-25

(2) 위의 정보를 받은 객체 배열을 생년월일 순으로 재 정렬한 뒤 새로이 정렬된 배열을 다시 출력하자(sorting 함수를 구현하기)

- 조건 -

1) sorting 시에 = 연산자 오버로딩 이용

출력값:

소팅 후 리스트:

이름	수학	국어	외국어	사회	생년월일
나문희	80	69	65	68	2002-4-5
차재우	66	24	80	83	2002-5-25
박명수	96	22	49	67	2002-12-31
장산범	51	61	63	87	2003-1-11
김진우	41	85	72	38	2003-2-4