

◆ 소스 수행 결과 화면

* 문제 12-(1)

```
Microsoft Visual Studio 디버그 x + v
10개 점수 입력>> 10 20 30 40 50 60 70 80 90 100
60점 이상은 4명
C:\C++\HW\x64\Debug\HW#3.exe(프로세스 8448개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요 ...|
```

* 문제 12-(3)

```
Microsoft Visual Studio 디버그 x + v
10개 점수 입력>> 10 20 30 40 50 60 70 80 90 100
60점 이상은 4명
C:\C++\HW\x64\Debug\HW#3_3.exe(프로세스 2540개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요 ...|
```

◆ 소스 구현 설명

* 문제 12-(1)

- 문제 정의

이 문제는 Dept 클래스를 사용하여 학생의 성적을 관리하고, 60점 이상의 학생 수를 계산하여 출력하는 프로그램을 작성하는 것이다. 이 프로그램은 학생 수와 성적을 입력받고, 60점 이상인 학생의 수를 계산하여 출력해야 한다.

- 문제 해결 방법 / 아이디어 평가

프로그램을 완성시키기 위해서는 복사 생성자인 Dept(Dept& dept) 메서드, 소멸자인 ~Dept() 메서드, read() 메서드 그리고 isOver60(int index) 메서드를 설계해야 한다. 먼저, Dept(Dept& dept) 복사 생성자는 다른 Dept 객체의 데이터를 복사하여 새로운 객체를 생성할 때 사용되어야 한다.

~Dept() 소멸자는 동적 메모리에서 할당된 scores 배열을 해제하여 메모리 누수를 방지하기 위해 delete를 사용하였다.

read() 메서드는 키보드로부터 학생의 성적을 입력받아야 한다. for 반복문을 사용하여 입력된 성적을 scores 배열에 저장하게 만들었다.

isOver60(int index) 메서드는 특정 인덱스의 학생 성적이 60점 이상인지 확인하는 기능을 해야 한다. 이를 위해 if-else문을 사용하여 60점 이상이면 true를 반환하고, 그렇지 않으면 false를 반환하게 하였다.

- 문제를 해결한 키 아이디어

문제를 해결한 키 아이디어는 'new 연산자'이다. 프로그램 실행 중에 어떻게 유연하게 학생 수를 조정할 수 있는지 고민이었는데, new 연산자를 사용하여 메모리를 할당하고 필요할 때만 메모리를 사용할 수 있도록 하였다. 이를 통해, 효율적이고 유연한 성적 관리 프로그램을 구현할 수 있었다.

*** 문제 12-(1)**

- 문제 정의

복사 생성자가 없는 경우의 오류를 처리하는 방법에 대한 것이다.

- 문제 해결 방법 / 아이디어 평가

먼저, Dept클래스의 복사 생성자인 Dept(Dept& dept) 메서드를 제거해야 한다. 복사 생성이 없는 상황에서는 실행 오류가 발생할 수 있는데, 실행 오류가 발생하지 않게 하려면 countPass 함수를 이용하면 된다. int countPass(Dept& dept); 처럼 countPass(Dept dept) 함수에 참조 연산자만 추가하여 참조에 의한 호출을 사용하면 복사 생성자를 제거하여도 실행 오류가 발생하지 않게 된다.

- 문제를 해결한 키 아이디어

문제를 해결한 키 아이디어는 '참조 연산자 &'이다. 복사 생성자가 없는 상황에서도 어떻게 실행 오류가 발생하지 않게 하는지가 고민이었는데, 간단하게 참조 연산자만 추가하여 코드를 극히 일부분만 수정하고 간단하게 오류를 발생하지 않게 할 수 있었다.