

Dept.h

```
6주차 과제 (전역 범위)
1  #ifndef Dept_H
2  #define Dept_H
3  class Dept {
4      int size;
5      int* scores;
6  public:
7      Dept(int size) {
8          this->size = size;
9          scores = new int[size];
10     }
11     Dept(Department& dept);
12     ~Dept();
13     int getSize() { return size; }
14     void read();
15     bool isOver60(int index);
16 };
17
18 #endif
```

1, 2, 18번 라인의 `#ifndef`, `#define`, `#endif` 명령어를 통하여 Dept_H가 이미 정의 되어있다면 중복으로 정의되는것을 방지합니다.

3~16번 라인에 Dept 클래스를 정의하고, 점수 값을 읽어와서 해당 값들을 배열로 만드는 함수들을 정의합니다.

11~12번 라인에 Dept복사 생성자와 소멸자를 정의합니다.

Dept.cpp

```
6주차 과제
Dept
isOver

1  #include<iostream>
2  using namespace std;
3
4  #include "Dept.h"
5
6  Dept::Dept(Dept& dept) {
7      this->size = dept.size;
8      this->scores = new int[this->size];
9      for (int i = 0; i < this->size; i++) {
10         this->scores[i] = dept.scores[i];
11     }
12 }
13 Dept::~Dept() {
14     delete[] scores;
15 }
16
17 void Dept::read() {
18     cout << size << "개 점수 입력>> ";
19     for (int i = 0; i < size; ++i) {
20         cin >> scores[i];
21     }
22 }
23 bool Dept::isOver60(int index) {
24     if (scores[index] > 60)
25         return true;
26     else
27         return false;
28 }
```

Dept.cpp

4번 라인에서 Dept.h 헤더파일을 include합니다.

6~11번 라인 Dept::Dept(Dept& dept) 복사 생성자를 이용하여 다른 Dept 객체에서 size와 scores 데이터를 복사합니다.

12~15번 라인에서 Dept::~~Dept() 소멸자를 이용하여 할당된 scores 배열을 해제하여 메모리 누수를 방지합니다.

17~22번 라인에서 사용자로부터 점수 값을 입력받아 scores 배열을 생성합니다.

23~28번 라인에서 주어진 인덱스가 점수 60점 이상이면 true를 반환하고 그렇지 않을 경우 false를 반환합니다.

main.cpp

```
6주차 과제 (전역 범위)

1  #include<iostream>
2  using namespace std;
3
4  #include "Dept.h"
5
6  int countPass(Dept dept) {
7      int count = 0;
8      for (int i = 0; i < dept.getSize(); i++) {
9          if (dept.isOver60(i)) count++;
10     }
11     return count;
12 }
13
14 int main() {
15     Dept com(10);
16     com.read();
17     int n = countPass(com);
18     cout << "60점 이상은 " << n << "명";
19 }
```

4번 라인에서 Dept.h 헤더파일을 include합니다.

6~12번 라인에 countPass에서 복사 생성자를 이용하여 만들 dept를 이용하여 60점 이상인 학생 수를 리턴받습니다.

14~19번 라인 메인함수에서 10개의 점수를 배열로 입력받아 60점 이상인 학생 수를 출력하도록 한다.

3번 문제(복사생성자 제거)

Dept.h

```
6주차 과제
1 #ifndef Dept_H
2 #define Dept_H
3 class Dept {
4     int size;
5     int* scores;
6 public:
7     Dept(int size) {
8         this->size = size;
9         scores = new int[size];
10    }
11    Dept(Department& dept);
12    ~Dept();
13    int getSize() { return size; }
14    void read();
15    bool isOver60(int index);
16 };
17
18 #endif
```



```
6주차 과제 (전역 범위)
1 #ifndef Dept_H
2 #define Dept_H
3 class Dept {
4     int size;
5     int* scores;
6 public:
7     Dept(int size) {
8         this->size = size;
9         scores = new int[size];
10    }
11    ~Dept();
12    int getSize() { return size; }
13    void read();
14    bool isOver60(int index);
15 };
16
17 #endif
```

기존 11번 라인에 복사생성자 Dept(Department& dept);를 삭제합니다.

3번 문제(복사생성자 제거)

Dept.cpp

```
6주차 과제
1  #include<iostream>
2  using namespace std;
3
4  #include "Dept.h"
5
6  Dept::Dept(Dept& dept) {
7      this->size = dept.size;
8      this->scores = new int[this->size];
9      for (int i = 0; i < this->size; i++) {
10         this->scores[i] = dept.scores[i];
11     }
12 }
13 Dept::~Dept() {
14     delete[] scores;
15 }
16
17 void Dept::read() {
18     cout << size << "개 점수 입력>> ";
19     for (int i = 0; i < size; ++i) {
20         cin >> scores[i];
21     }
22 }
23 bool Dept::isOver60(int index) {
24     if (scores[index] > 60)
25         return true;
26     else
27         return false;
28 }
```



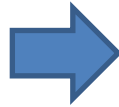
```
6주차 과제
1  #include<iostream>
2  using namespace std;
3
4  #include "Dept.h"
5
6  Dept::~Dept() {
7      delete[] scores;
8  }
9
10 void Dept::read() {
11     cout << size << "개 점수 입력>> ";
12     for (int i = 0; i < size; ++i) {
13         cin >> scores[i];
14     }
15 }
16 bool Dept::isOver60(int index) {
17     if (scores[index] > 60)
18         return true;
19     else
20         return false;
21 }
```

기존 6~12번 라인에 복사생성자를 사용하여 만든 객체를 삭제합니다.

3번 문제(복사생성자 제거)

main.cpp

```
6주차 과제 (전역)
1  #include<iostream>
2  using namespace std;
3
4  #include "Dept.h"
5
6  int countPass(Dept dept) {
7      int count = 0;
8      for (int i = 0; i < dept.getSize(); i++) {
9          if (dept.isOver60(i)) count++;
10     }
11     return count;
12 }
13
14 int main() {
15     Dept com(10);
16     com.read();
17     int n = countPass(com);
18     cout << "60점 이상은 " << n << "명";
19 }
```



```
6주차 과제 (전역)
1  #include<iostream>
2  using namespace std;
3
4  #include "Dept.h"
5
6  int countPass(Dept& dept) {
7      int count = 0;
8      for (int i = 0; i < dept.getSize(); i++) {
9          if (dept.isOver60(i)) count++;
10     }
11     return count;
12 }
13
14 int main() {
15     Dept com(10);
16     com.read();
17     int n = countPass(com);
18     cout << "60점 이상은 " << n << "명";
19 }
```

기존 6번 라인에 복사생성자로 만든 객체를 호출하던 것에서
int countPass(Dept& dept)로 변경하여 참조에 의한 호출로 변경합니다.