프로그래밍입문(2) 실습

9주차

복사 생성자

- 예제**1)**

```
1   #include <iostream>
2   using namespace std;
3
4 ▾ class MyClass{
5       private:
6           int num1;
7           int num2;
8       public:
9 ▾       MyClass(int a, int b){
10              num1=a;
11              num2=b;
12          }
13 ▾      void ShowData(){
14              cout<<"num1: "<<num1<<"num2: "<<num2<<endl;
15          }
16  };
17 ▾ int main(){
18      MyClass mc1(50, 30);
19      MyClass mc2=mc1;
20      mc2.ShowData();
21      return 0;
22  }
```

복사 생성자

- ## 예제2)

```cpp
1   #include <iostream>
2   using namespace std;
3
4   class Circle{
5       int radius;
6       public:
7           Circle(){ radius = 1; }
8           Circle(int radius){   this->radius = radius; }
9           void setRadius(int radius){ this->radius = radius; }
10          double getArea(){ return 3.14*radius*radius; }
11  };
12
13  int main(){
14      Circle donut;
15      readRadius(donut);
16      cout<<"donut의 면적 ="<<donut.getArea()<<endl;
17  }
```

```
정수 값으로 반지름을 입력하세요>>3
donut의 면적 =28.26
```

3

복사 생성자 - 깊은복사

- 예제3)

```cpp
1   #include <iostream>
2   #include <cstring>
3   using namespace std;
4
5 - class data{
6       private:
7           char *name;
8           char *phone;
9           int age;
10
11      public:
12 -         data(const char* _name, const char* _phone, int _age){
13              name = new char[strlen(_name)+1];
14              strcpy(name, _name);
15              phone = new char[strlen(_phone)+1];
16              strcpy(phone, _phone);
17              age = _age;
18          }
19 -         data(const data& object){
20              name = new char[strlen(object.name)+1];
21              strcpy(name, object.name);
22              phone = new char[strlen(object.phone)+1];
23              strcpy(phone, object.phone);
24              age = object.age;
25          }
```

복사 생성자 - 깊은복사

- 예제**3)**

```cpp
26         void output(){
27             cout << name << endl;
28             cout << phone << endl;
29             cout << age << endl;
30         }
31         void modify(const char *val){
32             strcpy(name, val);
33         }
34         ~data(){
35             delete[] name;
36             delete[] phone;
37         }
38 };
39
40 int main(){
41     data *a = new data("sosal", "010-2340-0432", 21);
42     data b(*a);
43     a->output();
44     b.modify("sosal2");
45     delete a;
46     b.output();
47 }
```

복사 생성자

- 예제**4)**

```cpp
1  #include <iostream>
2  using namespace std;
3
4  class Class319{
5      int size;
6      int *element;
7      public:
8          Class319(int size){
9              this->size=size;
10             element=new int[size];
11             for(int i=0;i<size;i++){
12                 element[i]=0;
13             }
14         }
15  };
```

복사 생성자

- 예제**4)**

1. 적절한 소멸자를 작성

2. 컴파일러가 삽입하는 디폴트 복사 생성자 작성

3. **Class319**에 깊은 복사를 실행하는 복사 생성자 코드를 작성

const

- 예제5)

```cpp
1   #include <iostream>
2
3   using namespace std;
4
5   class Line{
6       int *ptr;
7       public:
8           int show_Length() const{
9               cout << *ptr << endl;
10          }
11          Line(int len){
12              cout<<"일반 생성자"<<endl;
13              ptr=new int;
14              *ptr=len;
15          }
16          Line(const Line &obj){
17              cout << "복사 생성자" << endl;
18              ptr = new int;
19              *ptr = *obj.ptr;
20          }
21          void setLength(int n){
22              *ptr=n;
23          }
24
25  };
```

```cpp
27  int main(){
28
29      Line line1(10);
30      line1.show_Length();
31
32      const Line line2(20);
33      //line2.setLength(30);
34      line2.show_Length();
35
36  }
```

8

friend

- 예제**6)**

```cpp
1   #include<iostream>
2   using namespace std;
3   class CT
4   {
5       int number;
6
7       public:
8           friend void setNumber(CT&, int);
9           int getNumber(){ return number; }
10
11  };
12
13  void setNumber(CT &c, int n){
14      c.number=n;
15  }
16
17  int main(){
18      CT ct1;
19      setNumber(ct1, 3);
20  }
21
```

static

- 예제**7)**

```cpp
1   #include<iostream>
2   using namespace std;
3
4   class Game{
5       static int players;
6       public:
7           Game(){
8               players++;
9           }
10          static void reset()
11          {
12          players = 0;
13          }
14          static void inc()
15          {
16          players++;
17          }
18          static void show()
19          {
20          cout << "현재 경기자수: " << players << endl;
21          }
22  };
23
24  int Game::players = 0;
25
26  int main(){
27      Game player1;
28      Game::reset();
29      Game::show();
30      Game player2;
31      Game::show();
32  }
```

Q & A