프로그래밍 언어 (2)
실습

참고 문헌 : IT강좌 프론티어 (www.freelec.com)

클래스의 완성 – 소멸자(1)

- 예제1)

```
#include <iostream>
using namespace std;

class ExConstructor
{
public:
    ExConstructor()
    {
        cout << "ExConstructor() called!" << endl;
    }

    ~ExConstructor()    //소멸자 정의
    {
        cout << "~ExConstructor() called!" << endl;
    }
};

int main()
{
    ExConstructor ec;
    return 0;
}
```

클래스의 완성 – 소멸자(2)

- 예제2)

```
#include <iostream>
using namespace std;

class Circle {
public:
    int radius;
    Circle();
    Circle(int r);
    ~Circle();      //소멸자 선언
    double getArea();
};
Circle::Circle() {
    radius = 1;
    cout << "반지름 " << radius << "원 생성" << endl;
}

Circle::Circle(int r) {
    radius = r;
    cout << "반지름 " << radius << "원 생성" << endl;
}

Circle::~Circle() { //소멸자 구현
    cout << "반지름 " << radius << "원 소멸" << endl;
}

double Circle::getArea() {
    return 3.14*radius*radius;
}
```

```
int main() {
    Circle donut;
    Circle pizza(30);
    return 0;
}
```

클래스의 완성 – 소멸자(2)

- 예제3-1)

```
#include <iostream>
#include <string>
using std::cout;

class Cat
{
private:
    int Age;
    double We;
    char* Name;
public:
    Cat(int age, double we, char* name)
    {
        Age=age;
        We=we;
        Name=name;
        cout<<"객체생성!₩n";
    }

    ~Cat( );
    int GetAge( );
    void SetAge(int age);
    double GetWe( );
    void SetWe(double we);
    char* GetName( );
    void SetName(char* we);
    void Meow( );
};
```

```
int Cat::GetAge( )
{
    return Age;
}
void Cat::SetAge(int age)
{
    Age = age;
}

double Cat::GetWe( )
{
    return We;
}
void Cat::SetWe(double we)
{
    We = we;
}

char* Cat::GetName( )
{
    return Name;
}
void Cat::SetName(char* name)
{
    //strcpy(Name,name);
    Name=name;
}
```

클래스의 완성 – 소멸자(3)

- 예제3-2)

```
Cat::~Cat( )
{
    cout<<"객체 소멸 !₩n";
}

void Cat::Meow( )
{
    cout<<"야옹~₩n";
}

int main( )
{
    Cat nabi(2,2.5,"메리");
    Cat nabi1(2,2.5,"메리");
    nabi.Meow( );
    cout<<nabi.GetName( )<<"는₩n";
    cout<<nabi.GetAge( )<<"살₩n";
    cout<<nabi.GetWe( )<<"kg₩n₩n";
    nabi.SetAge(3);
    nabi.SetWe(3.5);
    nabi.SetName("나비");
    nabi.Meow( );
    cout<<nabi.GetName( )<<"는₩n";
    cout<<nabi.GetAge( )<<"살₩n";
    cout<<nabi.GetWe( )<<"kg₩n";
    return 0;
}
```

클래스의 완성 - 복사생성자

- 예제4)

```
#include <iostream>
using namespace std;
class MyClass
{
private:
    int num1;
    int num2;
public:
    MyClass(int a, int b) {
        num1 = a;
        num2 = b;
    }
    void ShowData() {
        cout << "num1: " << num1 << " num2: " << num2 << endl;
    }
};
int main() {
    MyClass mc1(50, 40);
    MyClass mc2 = mc1;
    mc2.ShowData();
    return 0;
}
```

클래스의 완성 - 복사생성자

- 예제5-1)

```
#include<iostream>
#include<cstring>
using namespace std;

class data{
private:
    char *name;
    char *phone;
    int age;
public:
    data(char *_name, char* _phone, int _age){
        name = new char[strlen(_name)+1];
        strcpy(name,_name);
        phone = new char[strlen(_phone)+1];
        strcpy(phone,_phone);
        age = _age;
    }
    data(const data& object){
        name = new char[strlen(object.name)+1];
        strcpy(name,object.name);
        // 이경우, private 멤버변수도 접근이 가능하므로, 되도록 매개변수를 const로 설정
        phone = new char[strlen(object.phone)+1];
        strcpy(phone,object.phone);
```

클래스의 완성 - 복사생성자

- 예제5-2)

```
        age = object.age;
    }
    void output(){
        cout<<name<<endl;
        cout<<phone<<endl;
        cout<<age<<endl;
    }
    void modify(char *val){
        strcpy(name,val);
    }
    ~data(){
        delete []name;
        delete []phone;
    }
};

int main(){
    data *a = new data("sosal","010-6450-7939",21);
      data b(*a);
    a->output();
    delete a;
    b.output();
}
```

클래스의 완성 - 복사생성자

- 예제6-1)

```
#include<iostream>
#include<cstring>
using namespace std;

class Person {
    char* name;
    int id;
public :
    Person(int id, char* name);
    Person(Person& person);
    ~Person();
    void changeName(char* name);
    void show() { cout << id << ',' << name << endl; }
};
Person::Person(int id, char* name) {//보통 생성자
    this->id = id;
    int len = strlen(name);
    this->name = new char[len + 1];
    strcpy(this->name, name);
}
```

클래스의 완성 - 복사생성자

- 예제6-2)

```
Person::Person(Person& person) {//복사 생성자, 깊은 복사를 한다.
    this->id = person.id;//id 값 복사
    int len = strlen(person.name);//name의 문자 갯수 할당
    this->name = new char[len + 1];//name 공간 할당.
    strcpy(this->name, person.name);//name 복사
    cout << "복사 생성자 실행. 원본 객체의 이름" << this->name << endl;
}
Person :: ~Person() {
    if (name)
        delete[] name;
}
void Person::changeName(char* name) {
    if (strlen(name) > strlen(this->name))//현재 이름보다 긴 이름으로 변경 불가
        return;
    strcpy(this->name, name);
}
```

클래스의 완성 - 복사생성자

- 예제6-2)

```
int main() {
    Person father(1, "kitae");//보통 생성자로 객체 생성
    Person dauther(father);//복사 생성자로 father 객체를 복사 생성

    cout << "dauther 객체 생성 직후 ----" << endl;
    father.show();
    dauther.show();

    dauther.changeName("Grace");//dauther 객체의 이름을 grace로 변경.
    cout << "dauther 이름을 Grace라고 변경 한 후 -----" << endl;
    father.show();
    dauther.show();

    return 0;
}
```

Q & A

참고 문헌 : IT강좌 프론티어 (www.freelec.com)