

# 프로그래밍 언어 (2)

## 실습

복습하기

# swap 함수 예제(&)

- 예제1)

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

void swap(int &a, int &b) {
    int c = a;
    a = b;
    b = c;
}

int main()
{
    int a = 150;
    int b = 300;

    cout<<"a = "<<a<<endl;
    cout<<"b = "<<b<<endl;

    swap(a, b);
    cout<<"a = "<<a<<endl;
    cout<<"b = "<<b<<endl;

    return 0;
}
```

# 함수 오버로딩 예제

- 예제2)

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

void function()
{
    cout<<"첫 번째 함수 호출"<<endl;
}

void function(char c)
{
    cout<<"두 번째 함수 호출"<<endl;
}

void function(int a, int b)
{
    cout<<"세 번째 함수 호출"<<endl;
}

int main()
{
    function();
    function('a');
    function(12, 13);

    return 0;
}
```

# swap 함수 예제(\*)

- 예제3)

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

void swap(int *a, int *b) {
    int c = *a;
    *a = *b;
    *b = c;
}

int main()
{
    int a = 50;
    int b = 30;
    cout<<"a = "<<a<<endl;
    cout<<"b = "<<b<<endl;

    swap(&a, &b);
    cout<<"a = "<<a<<endl;
    cout<<"b = "<<b<<endl;

    return 0;
}
```

# 디폴트

- 예제4)

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

int function(int a = 0) {
    return a+1;
}

int main()
{
    cout<<function(19)<<endl;
    cout<<function()<<endl;

    return 0;
}
```

# 클래스 기초

- 예제5)

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

struct Account {
    char accID[20];    // 계좌 번호
    char secID[20];   // 비밀번호
    char name[20];    // 이름
    int balance;      // 잔액
};

void Deposit(Account &acc, int money) // 입금
{
    acc.balance+=money;
}

void Withdraw(Account &acc, int money) // 출금
{
    acc.balance-=money;
}
```

```
int main(void)
{
    Account yoon={"1234", "2321", "yoon", 1000};

    Deposit(yoon, 100);
    cout<<"잔액 : "<<yoon.balance<<endl;

    Withdraw(yoon, 200);
    cout<<"잔액 : "<<yoon.balance<<endl;

    return 0;
}
```

# 클래스 응용

- 예제6)

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

const int OPEN=1;
const int CLOSE=2;

class Door{
private:
    int state;
public:
    void Open(){
        state=OPEN;
    }
    void Close(){
        state=CLOSE;
    }
    void ShowState(){
        cout<<"현재 문의 상태 : ";
        cout<<((state==OPEN)? "OPEN" : "CLOSE")<<endl;
    }
};
```

```
int main( )
{
    Door d;
    //d.state=OPEN;

    d.Open( );
    d.ShowState( );

    return 0;
}
```

# 클래스의 완성 - 생성자

- 예제7)

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

const int SIZE=20;

class AAA
{
    int i, j;
public:
    AAA() //생성자.
    {
        cout<<"생성자 호출"<<endl;
        i=10, j=20;
    }
    void ShowData()
    {
        cout<<i<<' '<<j<<endl;
    }
};
```

```
int main()
{
    AAA aaa;
    aaa.ShowData();

    return 0;
}
```

# 클래스의 완성 - 정보은닉

## • 예제8)

```
#include<iostream>
using std::cout;
using std::endl;
using std::cin;

class Point
{
    int x;    // x좌표의 범위 : 0 ~ 100
    int y;    // y좌표의 범위 : 0 ~ 100
public:
    int GetX(){ return x; }
    int GetY(){ return y; }

    void SetX(int _x);
    void SetY(int _y);
};

void Point::SetX(int _x)
{
    if(_x<0 || _x>100)
    {
        cout<<"X좌표 입력 오류, 확인 요망"<<endl;
        return;
    }
    x=_x;
}
```

```
void Point::SetY(int _y)
{
    if(_y<0 || _y>100)
    {
        cout<<"Y좌표 입력 오류, 확인 요망"<<endl;
        return;
    }
    y=_y;
}

int main()
{
    int x, y;
    cout<<"좌표입력 : ";
    cin>>x>>y;

    Point p;
    p.SetX(x);
    p.SetY(y);

    cout<<"입력 된 데이터를 이용해서 그림을 그림"<<endl;
    return 0;
}
```

# 클래스의 완성 - 캡슐화

- 예제9)

```
#include<iostream>
using std::cout;
using std::endl;
using std::cin;

class Point
{
    int x; // x좌표의 범위 : 0 ~ 100
    int y; // y좌표의 범위 : 0 ~ 100
public:
    int GetX(){ return x; }
    int GetY(){ return y; }

    void SetX(int _x);
    void SetY(int _y);
};

void Point::SetX(int _x)
{
    if(_x<0 || _x>100) {
        cout<<"X좌표 입력 오류, 확인 요망"<<endl;
        return;
    }
    x=_x;
}

void Point::SetY(int _y)
{
    if(_y<0 || _y>100)
    {
        cout<<"Y좌표 입력 오류, 확인 요망"<<endl;
        return;
    }
    y=_y;
}
```

```
class PointShow
{
public:
    void ShowData(Point p)
    {
        cout<<"x좌표 : "<<p.GetX()<<endl;
        cout<<"y좌표 : "<<p.GetY()<<endl;
    }
};

int main()
{
    int x, y;
    cout<<"좌표입력 : ";
    cin>>x>>y;

    Point p;
    p.SetX(x);
    p.SetY(y);

    PointShow show;
    show.ShowData(p);

    return 0;
}
```

# 클래스 접근 지정자

## • 예제10)

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

const int OPEN = 1;
const int CLOSE = 0;

class Door{
private:
    int state;
public:
    void Open(){
        state = OPEN;
    }
    void Close(){
        state = CLOSE;
    }
    void ShowState(){
        cout<<"현재 문은 ";
        cout<<((state==OPEN)? "OPEN" : "CLOSE")<<endl;
    }
};
```

```
int main()
{
    Door door;

    door.Close();
    door.ShowState();

    door.Open();
    door.ShowState();

    return 0;
}
```

# 클래스의 완성 - 소멸자(2)

- 예제11-1)

```
#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)

- 예제11-2)

```
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

- 예제12)

```
#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;
}
```

# 상속 기본 예제

- ex13)

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

class Human {
    string name;
    int age;
public:
    Human(string n, int a) : name(n), age(a) { }
    string getName(){
        return name;
    }
    int getAge() {
        return age;
    }
    void setName(string n){
        name = n;
    }
    void setAge(int a){
        age = a;
    }
    void print(){
        cout << "이름: " << name << endl;
        cout << "나이: " << age << endl;
    }
    ~Human() {
        cout << "Human 소멸자호출" << endl;
    }
};
```

# 상속 기본 예제

- ex14)

```
class Student : public Human {
    string major;
public:
    Student(string name, int age, string major) : Human(name, age) {
        this->major = major;
    }
    string getMajor(){
        return major;
    }
    void setMajor(string m){
        major = m;
    }

    void print(){
        Human::print(); // 부모클래스에있는함수를호출할수있다.
        //cout << "이름: " << name << endl;
        //cout << "나이: " << age << endl;
        cout << "전공: " << major << endl;
    }

    ~Student() {
        cout << "Student 소멸자호출" << endl;
    }
};
```

```
void main(){
    Student s1("명진", 21, "컴퓨터"), s2("미현", 22, "경영"), s3("용준", 24, "전자");

    s1.print();
    s2.print();
    s3.print();
}
```

# C++ 상속의 이해

- ex15)

>> 다음 클래스는 은행 계좌 정보를 담을 수 있도록 정의되어 있는 Account 클래스이다.

```
class Account
{
private:
    char acc_num[10]; // 계좌 번호
    int balance;     // 계좌 잔액
public:
    Account (char* num, int bal)
    {
        strcpy(acc_num, num);
        balance = bal;
    }
};
```

# C++ 상속의 이해

- ex15) 계속

>> 앞 슬라이드의 클래스를 public 상속하는 KBAccount 클래스를 정의하라.

>> KBAccount 클래스는 Account 클래스가 지니고 있는 멤버 변수 이외에 고객별 이체 한도 정보를 담고 있는 멤버 변수 (trans\_limit)를 지녀야 한다.

>> 다음 슬라이드에 제시하고 있는 main 함수의 예와 실행 결과를 참조해서 정의하라.

# C++ 상속의 이해

- ex15) 계속

- main 함수의 예

```
int main(void)
{
    //계좌 번호: 1234-5678
    // 초기 입금액: 5000원
    // 이체 한도: 100000원
    KBAccount acc("1234-5678", 5000, 100000);
    acc.ShowData();
    return 0;
};
```

# 참고문헌

---

- <https://modoocode.com/210>
- <https://wikidocs.net/22468>
- <http://algamza.blogspot.com/2016/03/c-operator-overloading.html>
- [http://tcpschool.com/cpp/cpp\\_operatorOverloading\\_intro](http://tcpschool.com/cpp/cpp_operatorOverloading_intro)

Q & A