

# 프로그래밍 언어 (2)

## 실습

# Const 멤버 변수 - 생성자

```
#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;
}
```

# 클래스 접근 지정자 – Const

```
#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;
    }
};
```

# Const 실습1

```
#include <iostream>

using namespace std;

class Line {
public:
    int getLength(void);
    Line(int len);
    Line(const Line &obj);
    ~Line();

private:
    int *ptr;
};

Line::Line(int len) {
    cout << "일반 생성자" << endl;
    ptr = new int;
    *ptr = len;
}

Line::Line(const Line &obj) {
    cout << "복사 생성자" << endl;
    ptr = new int;
    *ptr = *obj.ptr; // copy the value
}
```

# Const 실습1

```
Line::~Line(void) {
    delete ptr;
}

int Line::getLength(void) {
    return *ptr;
}

void display(Line obj) {
    cout << "선의 길이: " << obj.getLength() << endl;
}

int main() {

    Line line1(10);
    Line line2 = line1;

    display(line1);
    display(line2);

    return 0;
}
```

# Const 실습2

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

class Complex
{
private:
    double real, imag;

public:
    Complex(double r = 0, double i = 0) : real{ r }, imag{ i } {}
    void operator =(double value) {
        real = value;
        imag = 0.0;
    }
    friend ostream &operator<<(ostream&, const Complex&);
};

ostream &operator<<(ostream&output, const Complex &complex)
{
    output << '(' << complex.real << "," << complex.imag << ')';
    return output;
}

int main()
{
    Complex a = 3;
    Complex b;
    cout << a;
    cout << b;
    return 0;
}
```

# Friend 선언 실습1

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

class Complex
{
private:
    double real, imag;

public:
    Complex(double r = 0, double i = 0) : real{ r }, imag{ i } { }
    void operator =(double value) {
        real = value;
        imag = 0.0;
    }
    friend ostream &operator<<(ostream&, const Complex&);
};

ostream &operator<<(ostream&output, const Complex &complex)
{
    output << '(' << complex.real << ", " << complex.imag << ')';
    return output;
}

int main()
{
    Complex a = 3;
    Complex b;
    cout << a;
    cout << b;
    return 0;
}
```

# Friend 선언 실습2

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

class Box {
    double length;
    double width;
    double height;
    bool isEmpty;

public:
    Box(double w, double l, double h) : length{ l }, width{ w }, height{ h } {
        isEmpty = true;
    }
    Box() : length{ 0 }, width{ 0 }, height{ 0 } {
        isEmpty = true;
    }
    double getVolume() {
        return length*width*height;
    }
    void print() {
        cout << "상자의 길이: " << length << endl;
        cout << "상자의 너비: " << width << endl;
        cout << "상자의 높이: " << height << endl;
        cout << "상자의 부피: " << getVolume() << endl << endl;
    }
}
```

# Friend 선언 실습2

```
double getHeight() const {  
    return height;  
}  
void setHeight(double height) {  
    this->height = height;  
}  
double getLength() const {  
    return length;  
}  
void setLength(double length) {  
    this->length = length;  
}  
double getWidth() const {  
    return width;  
}  
void setWidth(double width) {  
    this->width = width;  
}  
Box Box::operator+(const Box& v2);  
bool Box::operator==(const Box& v2);  
bool Box::operator<(Box& v2);  
friend void printBox(Box box);
```

```
};
```

# Friend 선언 실습2

```
Box Box::operator+(const Box& v2)
{
    Box v;
    v.setLength(this->length + v2.length);
    v.setWidth(this->width + v2.width);
    v.setHeight(this->height + v2.height);
    return v;
}
bool Box::operator==(const Box& v2)
{
    return (this->length == v2.length &&
            this->width == v2.width &&
            this->height == v2.height);
}
bool Box::operator<(Box& v2)
{
    return (this->getVolume() < v2.getVolume());
}
void printBox(Box box)
{
    cout << "상자의 길이: " << box.length << endl;
    cout << "상자의 너비: " << box.width << endl;
    cout << "상자의 높이: " << box.height << endl;
    cout << "상자의 부피: " << box.getVolume() << endl << endl;
}

int main() {
    cout << "상자 #1" << endl;
    Box box1{ 10.0, 10.0, 10.0 };
    printBox(box1);

    return 0;
}
```

# Static 실습

```
#include<iostream>
using namespace std;
class Game
{
    static int players;
public:
    Game()
    {
        players++;
    }
    static void reset()
    {
        players = 0;
    }
    static void inc()
    {
        players++;
    }
    static void show()
    {
        cout << "현재 경기자수: " << players << endl;
    }
};
int Game::players = 0;
int main()
{
    Game player1;
    Game::reset();
    Game::show();
    Game player2;
    Game::show();
}
```

Q & A