

## 17-03-15-Inheritance(상속)-Point

```

/*ColorPoint.h*/
#include "Point.h"

#ifndef COLORPOINT_H
#define COLORPOINT_H

class ColorPoint : public Point{
private:
    string color;

public:
    void setColor(string color){
        this->color = color;
    }
    void showColor(){
        cout << this->color << " >>> ";
        showPoint();
    }
};

#endif
-----
/*Point.h*/
#include <iostream>
#include <string>

using namespace std;

#ifndef POINT_H
#define POINT_H

class Point{
protected:
    int x, y;

public:
    void setPoint(int x, int y){
        this->x = x;
        this->y = y;
    }
    void showPoint(){
        cout << "(" << this->x << "," << this->y << ")" << endl;
    }

    friend ostream& operator<<(ostream& os, const Point &p);
    friend istream& operator>>(istream& is, Point &p);
};

#endif

```

```

/*main.cpp*/
#include "ColorPoint.h"
#include "Point.h"

ostream& operator<<(ostream& os, const Point &p){
    os << "(" << p.x << "," << p.y << ")" << endl;
    return os;
}

istream& operator>>(istream& is, Point &p){
    is >> p.x >> p.y;
    return is;
}

int main(){
    Point p;
    ColorPoint cp;

    p.setPoint(7, 3);
    cout << p;

    cp.setPoint(5, 6);
    //cp.showPoint();

    cp.setColor("Red");
    cp.showColor();

    return 0;
}

```

## 17-03-15-Inheritance\_Casting-Point

```

/*ColorPoint.h*/
#include "Point.h"

#ifndef COLORPOINT_H
#define COLORPOINT_H

class ColorPoint : public Point{
private:
    string color;

public:
    void setColor(string color){
        this->color = color;
    }
    void showColor(){
        cout << this->color << " >>> ";
        showPoint();
    }
};

#endif
-----
/*Point.h*/
#include <iostream>
#include <string>

using namespace std;

#ifndef POINT_H
#define POINT_H

class Point{
protected:
    int x, y;

public:
    void setPoint(int x, int y){
        this->x = x;
        this->y = y;
    }
    void showPoint(){
        cout << "(" << this->x << "," << this->y << ")" << endl;
    }

    friend ostream& operator<<(ostream& os, const Point &p);
    friend istream& operator>>(istream& is, Point &p);
};

#endif

```

```

/*main.cpp*/
#include "ColorPoint.h"
#include "Point.h"

ostream& operator<<(ostream& os, const Point &p){
    os << "(" << p.x << "," << p.y << ")" << endl;
    return os;
}

istream& operator>>(istream& is, Point &p){
    is >> p.x >> p.y;
    return is;
}

int main(){
    Point p, *pBase;
    ColorPoint cp, *pDer;

    /*pDer = &cp;
    pBase = pDer; // 업캐스팅

    pDer->setPoint(10, 20);
    pDer->setColor("Red");
    pDer->showColor();*/

    pBase = &p;
    pBase->setPoint(33, 44);
    pDer = (ColorPoint*)pBase; // 다운캐스팅
    pDer->showPoint();

    return 0;
}

```

## 17-03-22-I\_Casting-ColorPoint

```

/*ColorPoint.h*/
#include <iostream>
#include <string>

using namespace std;

class Point {
protected:
    int x, y;
public:
    Point(int x=0, int y=0) {
        this->x = x; this->y = y;
    }

    int getX() {
        return this->x;
    }

    int getY() {
        return this->y;
    }

    void setPoint(int x, int y){
        this->x = x;
        this->y = y;
    }

protected:
    void move(int x, int y) {
        this->x = x; this->y = y;
    }
};

```

```

class ColorPoint : public Point{
    string color;
public:
    ColorPoint(int x=0, int y=0, string color="Black")
: Point(x, y){
        this->color = color;
    }

    void setColor(string color){
        this->color = color;
    }

    string getColor(){
        return this->color;
    }

    void show(){
        cout << getColor() << "색으로" << "(" <<
getX() << ", " << getY() << ")에 위치한 점입니다." << endl;
};
}

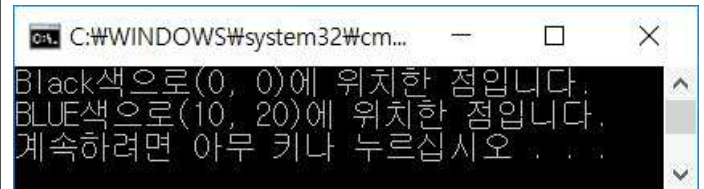
-----
/*main.cpp*/
#include "ColorPoint.h"

int main() {
    ColorPoint zeroPoint; // Black에 (0, 0) 위치의 점
    zeroPoint.show();

    ColorPoint cp(5, 5, "RED");
    cp.setPoint(10, 20);
    cp.setColor("BLUE");
    cp.show();

    return 0;
}

```



## 17-03-22-I\_Casting-Multiple\_TV

```

/*TV.h*/
#include <iostream>
#include <string>
using namespace std;

class TV {
private:
    int size;
public:
    TV();
    TV(int s);
    int getSize();
};

class WideTV : public TV {
private:
    bool videoIn;
public:
    WideTV(int s, bool in);
    bool getVideoIn();
};

class SmartTV : public WideTV {
private:
    string ipAddress;
public:
    SmartTV(string ipAdd, int s);
    string getIpAddress();
};

```

```

/*TV.cpp*/
#include "TV.h"

TV::TV() {
    this->size = 0;
}

TV::TV(int s){
    this->size = s;
}

int TV::getSize(){ return this->size; }

WideTV::WideTV(int s, bool in) : TV(s){
    this->videoIn = in;
}

bool WideTV::getVideoIn(){ return videoIn; }

SmartTV::SmartTV(string ipAdd, int s) : WideTV(s, false){
    this->ipAddress = ipAdd;
}

string SmartTV::getIpAddress(){ return this->ipAddress; }

-----
/*main.cpp*/
#include "TV.h"

int main(void){
    SmartTV mytv("120. 11. 22. 1", 64);

    cout << "크기 >>> " << mytv.getSize() << endl;
    cout << "비디오 유무 >>> " << mytv.getVideoIn()
<< endl;
    cout << "ip주소 >>> " << mytv.getIpAddress() <<
endl;

    return 0;
}

```

## 17-03-22-I\_Casting-MyQueue

```

/*MyQueue.h*/
#include <iostream>
using namespace std;

class BaseArray{
protected:
    int capacity;
    int *mem;
public:
    BaseArray(int c = 100){
        this->capacity = c;
        this->mem = new int[c];
    }

    ~BaseArray() {
        delete[] this->mem;
    }

    void put(int index, int val){
        this->mem[index] = val;
    }

    int get(int index) {
        return mem[index];
    }

    int getCapacity() {
        return capacity;
    }
};

class MyQueue : public BaseArray{
    int front, rear;
    int size;
public:
    MyQueue(int c = 100) : BaseArray(c){
        front = -1;
        rear = -1;
        this->size = 0;
    }

    // 큐에 삽입
    void enqueue(int val);

    // 큐 삭제
    int dequeue();

    // 큐의 크기
    int length();
};

-----
/*MyQueue.cpp*/
#include "MyQueue.h"

// 큐에 삽입
void MyQueue::enqueue(int val){
    rear++;
    if (rear == this->capacity){
        cout << "큐가 포화상태, " << val << "추가 못함\n";
        return;
    }

    put(rear, val);
    this->size++;
}

// 큐 삭제
int MyQueue::dequeue() { // 큐에서 제거하여 출력
    if (rear == -1){
        cout << "큐가 공백상태\n";
        return -1;
    }

    front++;
    this->size--;
    cout << "<size 감소>" << endl;
    get(front);
}

// 큐의 크기
int MyQueue::length(){
    return this->size;
}

```

```

/*main.cpp*/
#include "MyQueue.h"

int main() {
    MyQueue mQ(5);
    int n;

    for (int i = 0; i<6; i++) {
        cout << "값 입력 >> ";
        cin >> n;
        mQ.enqueue(n); // 큐에 삽입
    }
    cout << endl;
    cout << "큐 삭제 >> " << mQ.dequeue() << endl;
    cout << "큐 삭제 >> " << mQ.dequeue() << endl;

    cout << "\n큐의 크기 >> " << mQ.length() <<
endl;

    while (mQ.length() != 0){
        cout << "큐 삭제 >> " << mQ.dequeue()
<< endl;
    }

    /*
    cout << "큐의 용량:" << mQ.capacity() << ", 큐의
크기:" << mQ.length() << endl;
    cout << "큐의 원소를 순서대로 제거하여 출력한다>>
";
    while(mQ.length() != 0) {
        cout << mQ.dequeue() << ' '; // 큐에서
제거하여 출력
    }
    cout << endl << "큐의 현재 크기 : " <<
mQ.length() << endl;
    */
    return 0;
}

```

## 17-03-22-I\_Casting-NamedCircle

```

/*NamedCircle.h*/
#include <iostream>
#include <string>

using namespace std;

class Circle{
protected:
    int radius;
public:
    Circle(int radius = 0){
        this->radius = radius;
    }

    void setRadius(int radius){
        this->radius = radius;
    }

    int getRadius(){
        return this->radius;
    }

    double getArea(){
        return 3.14*radius*radius;
    }
};

class NamedCircle : public Circle{
    string name;
public:
    NamedCircle(int radius=0, string name ="default
pizza") : Circle(radius){
        this->name = name;
    }

    void set(int size, string n){
        this->radius = size;
        this->name = n;
    }

    string getName(){
        return this->name;
    }

    void Show(){
        cout << "반지름이 " << this->radius << "
인 " << this->name << endl;
    }
};

```

```

/*main.cpp*/
#include "NamedCircle.h"

void main(){
    NamedCircle waffle(3, "waffle");
    waffle.Show();

    NamedCircle c[5];
    int size;
    string name;
    int max = 0;

    cout << "5개의 정수 반지름과 원의 이름을 입력하세
요" << endl;
    for (int i = 0; i < 5; i++){
        cout << i+1 << " >>> ";
        cin >> size;
        getline(cin, name, '\n');
        //cin >> name;
        c[i].set(size, name);
    }

    for (int i = 1; i < 5; i++){
        if (c[max].getRadius() < c[i].getRadius()){
            max = i;
        }
    }

    cout << "가장 면적이 큰 피자는 " <<
c[max].getName() << " 입니다." << endl;
}

```

```

C:\WINDOWS\system32\cmd.exe
반지름이 3인 waffle
5개의 정수 반지름과 원의 이름을 입력하세요
1 >>> 5 크림피자
2 >>> 8 치즈피자
3 >>> 25 대왕피자
4 >>> 30 블랙홀 피자
5 >>> 15 마늘피자
가장 면적이 큰 피자는 블랙홀 피자 입니다.
계속하려면 아무 키나 누르십시오...

```

## 17-03-29-I\_Casting-Converter(KmToMile)

```

/*Converter.h*/
#include <iostream>
#include <string>
using namespace std;

class Converter{
protected:
    double ratio;
    virtual double convert(double src) = 0; // src를
다른 단위로 변환
    virtual string getSourceString() = 0; // 소스 단위 명칭
    virtual string getDestString() = 0; // dest 단위 명칭
public:
    Converter(double ratio) {
        this->ratio = ratio;
    }

    void run() {
        double src;
        cout << getSourceString() << "을 " <<
getDestString() << "로 바꿉니다. ";
        cout << getSourceString() << "을 입력하세요>> ";
        cin >> src;
        cout << "변환 결과 : " << convert(src) <<
getDestString() << endl;
    }
};

class Km : public Converter{
protected:
    double ratio;

public:
    Km(double ratio) : Converter(ratio){};

    virtual double convert(double src){ // src를 다른 단위로 변환
        // 1km -> 0.621371mile
        this->ratio = src*0.621371;
        return this->ratio;
    }

    virtual string getSourceString(){ // 소스 단위 명칭
        return "(Mile)";
    }

    virtual string getDestString(){// dest 단위 명칭
        return "(Km)";
    }
};

```

```

class Mile : public Converter{
protected:
    double ratio;

public:
    Mile(double ratio) : Converter(ratio){};

    virtual double convert(double src){ // src를 다른
단위로 변환
        // 0.621371mile -> 1km
        this->ratio = src / 0.621371;
        return this->ratio;
    }

    virtual string getSourceString(){ // 소스 단위 명칭
        return "(Km)";
    }

    virtual string getDestString(){// dest 단위 명칭
        return "(Mile)";
    }
};

```

---

```
#include "Converter.h"
```

```

void main(){
    Km k(20000);
    k.run();
}

```



```

C:\WINDOWS\system32\cmd.exe
(Mile)을 (Km)로 바꿉니다. (Mile)을 입력하세요>> 500
변환 결과 : 310.685(Km)
계속하려면 아무 키나 누르십시오 . . .

```

## 17-03-29-I\_Casting-Converter(WonToDollor)

```

/*Converter.h*/
#include <iostream>
#include <string>
using namespace std;

class Converter{
protected:
    double ratio;
    virtual double convert(double src) = 0; // src를
다른 단위로 변환
    virtual string getSourceString() = 0; // 소스 단위 명칭
    virtual string getDestString()=0; // dest 단위 명칭
public:
    Converter(double ratio) {
        this->ratio = ratio;
    }

    void run() {
        double src;
        cout << getSourceString() << "을 " <<
getDestString() << "로 바꿉니다. ";
        cout << getSourceString() << "을 입력하세요>> ";
        cin >> src;
        cout << "변환 결과 : " << convert(src) <<
getDestString() << endl;
    }
};

class Won : public Converter{
protected:
    double ratio;

public:
    Won(double ratio) : Converter(ratio){};

    virtual double convert(double src){ // src를 다른
단위로 변환
        // 1달러 -> 1,113.50원
        this->ratio = src*1113.50;
        return this->ratio;
    }

    virtual string getSourceString(){ // 소스 단위 명칭
        return "[달러]";
    }

    virtual string getDestString(){// dest 단위 명칭
        return "[원]";
    }
};

```

```

class Dollor : public Converter{
protected:
    double ratio;

public:
    Dollor(double ratio) : Converter(ratio){};

    virtual double convert(double src){ // src를 다른
단위로 변환
        // 1,113.50원 -> 1달러
        this->ratio = src / 1113.50;
        return this->ratio;
    }

    virtual string getSourceString(){ // 소스 단위 명칭
        return "[원]";
    }

    virtual string getDestString(){// dest 단위 명칭
        return "[달러]";
    }
};

```

```

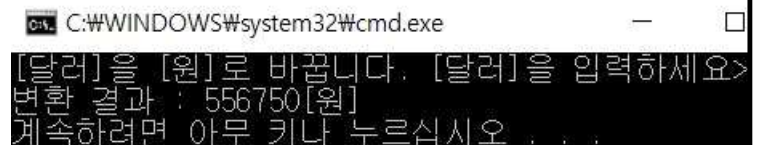
/*main.cpp*/
#include "Converter.h"

```

```

void main(){
    Won w(20000);
    w.run();
}

```



```

C:\WINDOWS\system32\cmd.exe
[달러]을 [원]로 바꿉니다. [달러]을 입력하세요>
변환 결과 : 556750[원]
계속하려면 아무 키나 누르십시오 . . .

```

## 17-03-29-I\_Casting-Overriding\_Connection

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

class Shape{
private:
    Shape* next;

public:
    Shape() { next = NULL; }

    void paint(){
        draw();
    }

    Shape* add(Shape * p){
        this->next = p; // 새로 생성한 주소를 넣는다.
        return p; // 마지막으로 받은 주소가 p이므로 p를 리턴
    }

    Shape* getNext() {
        cout << "[다음 주소 불러오기]" << endl;
        return this->next;
    }

    ~Shape(){}

protected:
    virtual void draw(){
        cout << "Shape Draw" << endl;
    }
};

class Circle : public Shape{
public:
    void paint(){
        draw(); // 범위지정 연산자
    }

    virtual void draw(){
        cout << "Circle Draw" << endl;
    }
};

class Line : public Shape{
public:
    void paint(){
        draw();
    }

    virtual void draw(){
        cout << "Line Draw" << endl;
    }
};
```

```
class Rectangle : public Shape{
public:
    virtual void paint(){
        draw();
    }

    virtual void draw(){
        cout << "Rectangle Draw" << endl;
    }
};

/*main.cpp*/
#include "Shape.h"

void main(){
    Shape *pStart, *pLast;
    pStart = new Circle();
    pLast = pStart;

    pLast = pLast->add(new Rectangle());
    pLast = pLast->add(new Line());
    pLast = pLast->add(new Circle());
    pLast = pLast->add(new Rectangle());
    pLast = pLast->add(new Line());

    Shape *p;
    p = pStart;
    while (p != NULL){ // 마지막 객체는 next가 null이
        기 때문
        p->paint();
        p = p->getNext(); // 다음 주소를 받아옴
    }
}
```

```
C:\WINDOWS\system32\cmd...
Circle Draw
[다음 주소 불러오기]
Rectangle Draw
[다음 주소 불러오기]
Line Draw
[다음 주소 불러오기]
Circle Draw
[다음 주소 불러오기]
Rectangle Draw
[다음 주소 불러오기]
Line Draw
[다음 주소 불러오기]
계속하려면 아무 키나 누르십시오 . . .
```



## 17-04-05-STL-Vector

```

/*main1.cpp*/
#include <iostream>
#include <vector>
#include <string>
using namespace std;

void main() {
    vector<string> name;
    string input;

    cout << "5개 입력" << endl;
    for (int i = 0; i < 5; i++) {
        getline(cin, input);
        name.push_back(input);
    }
    cout << "벡터 크기 >>> " << name.size() << endl;
    cout << "벡터 용량 >>> " << name.capacity() << endl;

    cout << "세번째 데이터 >>> " << name.at(2) << endl;
    cout << "네번째 데이터 >>> " << name[3] << endl;

    name.at(2) = "버그"; // 바꾼 데이터
    name[3] = "썩썩"; // 바꾼 데이터

    cout << "세번째 데이터 >>> " << name.at(2) << endl;
    cout << "네번째 데이터 >>> " << name[3] << endl;
}

```

```

C:\Windows\system32\cmd.exe
32
55
64
77
89
벡터 크기 >>> 5
벡터 용량 >>> 6
세번째 데이터 >>> 버그
네번째 데이터 >>> 썩썩
세번째 데이터 >>> 버그
네번째 데이터 >>> 썩썩
계속하려면 아무 키나 누르십시오 . . .

```

```

/*main2.cpp*/
#include <iostream>
#include <vector>
#include <string>
using namespace std;

void main() {
    vector<string> name;
    string input;

    cout << "5개 입력" << endl;
    for (int i = 0; i < 5; i++) {
        getline(cin, input);
        name.push_back(input);
    }
    string max = name[0];

    for (int i = 0; i < name.size(); i++) {
        if (max < name[i])
            max = name[i];
    }

    cout << "사전적으로 맨 나중에 나오는 이름 >>> "
    << max << endl;
}

```

```

C:\Windows\system32\cmd.exe
5개 입력
키
니
티
이
시
사전적으로 맨 나중에 나오는 이름 >>> 티글
계속하려면 아무 키나 누르십시오 . . .

```

## 17-04-05-Template\_Class-MyStack

```

/*MyStack.h*/
#include <iostream>
#include <string>
using namespace std;

/* 제네릭 클래스 선언 */
template <class T>
class MyStack {
    int top;
    T data[100]; // T타입의 배열

public:
    MyStack() { top = -1; };
    void push(T element);
    T pop();
};

/* 제네릭 클래스 구현 */
template <class T>
void MyStack<T>::push(T value) {
    data[++top] = value;
}

template <class T>
T MyStack<T>::pop(){
    return data[top--];
    /*T tmp = data[top];
    top--;
    return tmp;*/
}

```

```

/*main.cpp*/
#include <iostream>
#include <string>
using namespace std;

template <typename T>
void myswap(T &a, T &b) {
    T tmp;
    tmp = a;
    a = b;
    b = tmp;
}

void main() {
    int a = 20, b = 200;
    myswap(a, b);
    cout << "a >>> " << a << "b >>> " << b << endl;

    string str1 = "콩길동", str2 = "콩콩길동";
    myswap(str1, str2);
    cout << "str1 >>> " << str1 << "str2 >>> " <<
    str2 << endl;

    char c1 = 'A', c2 = 'F';
    myswap(c1, c2);
    cout << "c1 >>> " << c1 << "c2 >>> " << c2 <<
    endl;
}

```

```

C:\Windows\system32\cmd.exe
999
666
3
정적변수
계속하려면 아무 키나 누르십시오 . . .

```



```

/*Cricle.h*/

#include <iostream>

using namespace std;

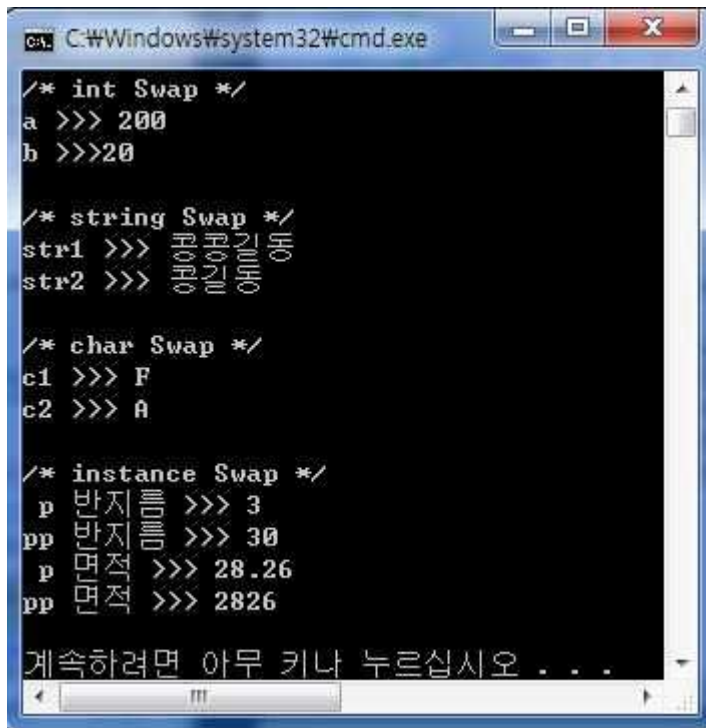
class Circle {
    double radius;

public:
    Circle(double r = 3) { this->radius = r; };

    double getArea() {
        return this->radius*this->radius*3.14;
    }

    int getRadius(){
        return this->radius;
    }
};

```



```

C:\Windows\system32\cmd.exe

/* int Swap */
a >>> 200
b >>>20

/* string Swap */
str1 >>> 콩길동
str2 >>> 콩콩길동

/* char Swap */
c1 >>> F
c2 >>> A

/* instance Swap */
p 반지름 >>> 3
pp 반지름 >>> 30
p 면적 >>> 28.26
pp 면적 >>> 2826

계속하려면 아무 키나 누르십시오 . . .

```

```

/*main.cpp*/

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

#include "Circle.h"

template <typename T>
void myswap(T &a, T &b) {
    T tmp;
    tmp = a;
    a = b;
    b = tmp;
}

void main() {
    /* int 스왑 */
    int a = 20, b = 200;
    myswap(a, b);
    cout << "/* int Swap */" << endl;
    cout << "a >>> " << a << endl;
    cout << "b >>>" << b << endl << endl;

    /* string 스왑 */
    string str1 = "콩길동", str2 = "콩콩길동";
    myswap(str1, str2);
    cout << "/* string Swap */" << endl;
    cout << "str1 >>> " << str1 << endl;
    cout << "str2 >>> " << str2 << endl << endl;

    /* char 스왑 */
    char c1 = 'A', c2 = 'F';
    myswap(c1, c2);
    cout << "/* char Swap */" << endl;
    cout << "c1 >>> " << c1 << endl;
    cout << "c2 >>> " << c2 << endl << endl;

    /* 객체 스왑 */
    Circle p(30), pp(3);
    myswap(p, pp);
    cout << "/* instance Swap */" << endl;
    cout << " p 반지름 >>> " << p.getRadius() << endl;
    cout << "pp 반지름 >>> " << pp.getRadius() << endl;
    cout << " p 면적 >>> " << p.getArea() << endl;
    cout << "pp 면적 >>> " << pp.getArea() << endl << endl;
}

```

## 17-04-12-STL\_iterator-Instance\_Sort

```

/*Circle.h*/
#include <iostream>
#include <string>
#include <vector>
#include <algorithm>
using namespace std;

class Circle{
    int radius;
    string name;
public:
    Circle(int radius = 3, string name="circle") {
        this->radius = radius;
        this->name = name;
    }
    double getArea() { return 3.14*radius*radius; }
    string getName() { return name; }

    bool operator <(const Circle &a) const{
        return this-> radius < a.radius;
    }
};

```

```

C:\Windows\system32\cmd.exe
반지름 입력 >>> 5
이름 입력 >>> z

반지름 입력 >>> 3
이름 입력 >>> r

반지름 입력 >>> 9
이름 입력 >>> o

반지름 입력 >>> 7
이름 입력 >>> a

반지름 입력 >>> -1
r:28.26
z:78.5
a:153.86
o:254.34
계속하려면 아무 키나 누르십시오 . . .

```

```

/*main.cpp*/
#include "Circle.h"

void main(){
    vector<Circle> data;
    int radius;
    string name;

    // vector에 Circle객체(반지름, 이름) 입력
    while(true){
        cout << "반지름 입력 >>> ";
        cin >> radius;
        if (radius == -1) break;
        getchar();
        cout << "이름 입력 >>> ";
        getline(cin, name, '\n');
        cout << endl;
        data.push_back(Circle(radius, name));
    }

    // 객체 정렬
    stable_sort( data.begin(), data.end() );
    vector<Circle>::iterator it; // it는 포인터

    for (it = data.begin(); it < data.end(); it++){
        cout << it->getName() << " : " <<
        it->getArea() << endl;
    }
}

```

## 17-04-12-STL\_iterator-Sort

```

#include <iostream>
#include <vector>
#include <algorithm>
#include <string>
using namespace std;

void main(){
    vector<string> data;
    string value;

    // vector에 정수 입력
    while(true){
        cout << "문자열 입력 (exit이면 stop) >>> ";
        getline(cin, value);
        if (value == "exit") break;
        data.push_back(value);
    }

    // #include <algorithm>의 함수
    sort(data.begin(), data.end());

    vector<string>::iterator it; // it는 포인터
    for (it = data.begin(); it < data.end(); it++){
        cout << *it << " ";
    }
}

```

```

C:\Windows\system32\cmd.exe
문자열 입력 (exit이면 stop) >>> id
문자열 입력 (exit이면 stop) >>> pw
문자열 입력 (exit이면 stop) >>> hi
문자열 입력 (exit이면 stop) >>> hello
문자열 입력 (exit이면 stop) >>> hell!
문자열 입력 (exit이면 stop) >>> exit
hell! hello hi id pw 계속하려면 아무 키나 누르십시오 . . .

```

```

#include "NationGame.h"

NationGame::NationGame(){
    data.push_back(NationGame("한국", "서울"));
    data.push_back(NationGame("일본", "도쿄"));
    data.push_back(NationGame("미국", "워싱턴"));
    data.push_back(NationGame("중국", "베이징"));
}

void NationGame::run(){
    int select = 0;

    cout << "***** 나라의 수도 맞추기 게임을 시작합니
다. *****" << endl;
    while (select != 3){
        cout << "정보 입력 : 1, 퀴즈 : 2, 종료 : 3 >>> ";
        cin >> select;

        switch (select){
            // 정보 입력
            case 1: input(); break;

            // 퀴즈
            case 2: quiz(); break;

            // 종료
            case 3: cout << "게임을 종료합니
다." << endl; break;
        }
    }

    void NationGame::input(){
        cout << "현재" << data.size() << "개의 나라가 있습
니다." << endl;
        vector<NationGame>::iterator it; // it는 포인터
        for (it = data.begin(); it < data.end(); it++){
            cout << "[" << it->getNation() << " / " <<
it->getName() << "]" << endl;
        }

        while(true){
            getchar();
            cout << "나라 입력 >>> ";
            getline(cin, nation, '\n');
            cout << "수도 입력 >>> ";
            getline(cin, name, '\n');
            cout << endl;
            if (nation == "no" && name == "no") break;
            data.push_back(NationGame(nation, name));
        }

        void NationGame::quiz(){
            int key = rand() % data.size();
            string q;

            while(true){
                key = rand() % data.size();
                getchar();
                cout << "<<" << data[key].getNation() <<
"의 수도는? >>> ";
                getline(cin, q, '\n');
                if (q == "exit")
                    break;
                else if (q == data[key].getName())
                    cout << "Correct !!" << endl;
                else
                    cout << "No !! 정답은 " <<
data[key].getName() << "입니다." << endl;
            }
        }
    }

```

```

/*NationGame.h*/
#include <iostream>
#include <string>
#include <vector>
#include <algorithm>
#include <stdlib.h>
#include <time.h>
using namespace std;

class NationGame{
    vector<NationGame> data;
    string nation;
    string name;

public:
    NationGame();
    NationGame(string nation, string name) {
        this->nation = nation;
        this->name = name;
    }

    void run();
    void input();
    void quiz();
    string getNation() { return this->nation; }
    string getName() { return this->name; }
};

```

```

/*main.cpp*/
#include "NationGame.h"

void main(){
    srand((unsigned)time(NULL));
    NationGame game;
    game.run();
}

```

```

C:\Windows\system32\cmd.exe
***** 나라의 수도 맞추기 게임을 시작합니다. *****
정보 입력 : 1, 퀴즈 : 2, 종료 : 3 >>> 1
현재4개의 나라가 있습니다.
[한국 / 서울]
[일본 / 도쿄]
[미국 / 워싱턴]
[중국 / 베이징]
나라 입력 >>> 룩셈부르크
수도 입력 >>> 룩셈부르크

나라 입력 >>> 브라질
수도 입력 >>> 브라질리아

나라 입력 >>> no
수도 입력 >>> no

정보 입력 : 1, 퀴즈 : 2, 종료 : 3 >>> 2
<<일본의 수도는? >>> 도쿄
Correct !!

<<브라질의 수도는? >>> 브라질
No !! 정답은 브라질리아입니다.

<<중국의 수도는? >>> 베이징
Correct !!

<<룩셈부르크의 수도는? >>> 룩셈부르크
Correct !!

<<룩셈부르크의 수도는? >>> exit
정보 입력 : 1, 퀴즈 : 2, 종료 : 3 >>> 3
게임을 종료합니다.
계속하려면 아무 키나 누르십시오 . . .

```

## 17-04-12-STL\_iterator-Sum

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

void main(){
    vector<int> data;
    int value;

    // vector에 정수 입력
    while(true){
        cout << "정수 입력 (-1이면 stop) >>> ";
        cin >> value;
        if (value == -1) break;
        data.push_back(value);
    }

    //vector<int>::const_iterator it;
    //const로 선언할 경우 값 변경 불가

    // 합계 구하기
    vector<int>::iterator it; // it는 포인터

    cout << "벡터의 크기 >>> " << data.size() << endl;
    //it = data.begin(); // 시작 지점
    for (it = data.begin(); it < data.end(); it++){
        *it += 10;
        cout << *it << " ";
    }
}
```

```
//for (int i=0; i< data.size(); i++){
    /*if (*it >= 100){ // it가 100이상이면 삭제
        it--;
        cout << *it << " : " <<
data.capacity() << endl;
        data.erase(it+1);
    }
    else if (*it < 100){
        cout << *it << " : " <<
data.capacity() << endl;
        sum += *it;
    }
}*/
```

## 17-04-12-Template-Biggest

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

template <typename T>
//template <class T> 를 적어주어도 된다.

T biggest(T data[], int n){
    T big = data[0];
    for (int i=0; i<n; i++)
        if (big < data[i])
            big = data[i];
    return big;
}

void main(){
    int x[] = {1, 10, 100, 5, 4};
    cout << biggest(x, 5) << endl; //5는 배열 x의 크기

    double y[] = {6.4, 3.3, -200.6, 7.7, 8.8, 20.3};
    cout << biggest(y, 6) << endl;

    char c[] = {'b', 'c', 'z', 's', 'q', 'u'};
    cout << biggest(c, 6) << endl;

    string s[] = {"홍길동", "박근혜", "심청", "손흥민"};
    cout << biggest(s, 4) << endl;
}
```

## 17-04-12-Template-Search

```

/*main.cpp*/

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

template <typename T>
//template <class T> 를 적어주어도 된다.

bool search (T search, T b[], int n){
    for (int i=0; i<n; i++){
        if (search == b[i])
            return true;

    }

    return false;
}

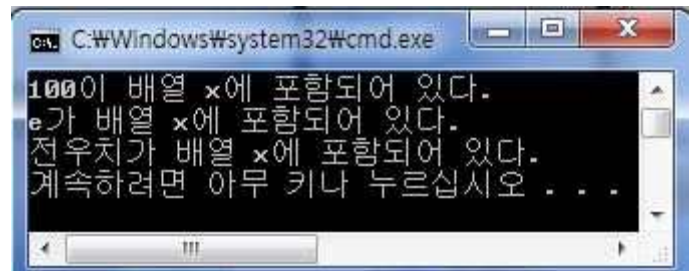
```

```

void main(){
    int x[] = {1, 10, 100, 5, 4};
    if (search(100, x, 5)) // 100이 배열 x에 포함되어 있는가?
        cout << "100이 배열 x에 포함되어 있다." << endl;
    else
        cout << "100이 배열 x에 포함되어있지 않다." << endl;

    char c[] = {'h', 'e', 'l', 'l', 'o'};
    if (search('e', c, 5)) // 100이 배열 x에 포함되어 있는가?
        cout << "e가 배열 x에 포함되어 있다." << endl;
    else
        cout << "e가 배열 x에 포함되어있지 않다." << endl;
}

```



## 17-04-19-File\_Binary

```

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

void main(){
    char* file = "system.ini";

    ofstream fout("system.ini", ios::binary);
    ifstream fin("c:\\windows\\system.ini", ios::binary);

    if (!fin)
        cout << file << "열기 오류" << endl;
    if (!fout)
        cout << file << "쓰기 오류" << endl;

    int count = 0;
    int c; // char를 써도 된다.

    //while (!fin.eof()) // 이걸 하게 되면 eof의 값까지 읽어 들인다
    while ((c = fin.get()) != EOF){
        // EOF : End Of File, 파일의 끝까지 문자 읽기
        fout.put(c);
        //fout << (char)c; // 이걸 써도 됨
        count++;
    }

    cout << "읽은 바이트 수는" << count << endl;

    fin.close(); // 파일 닫기
    fout.close(); // 파일 닫기

    count = 0;

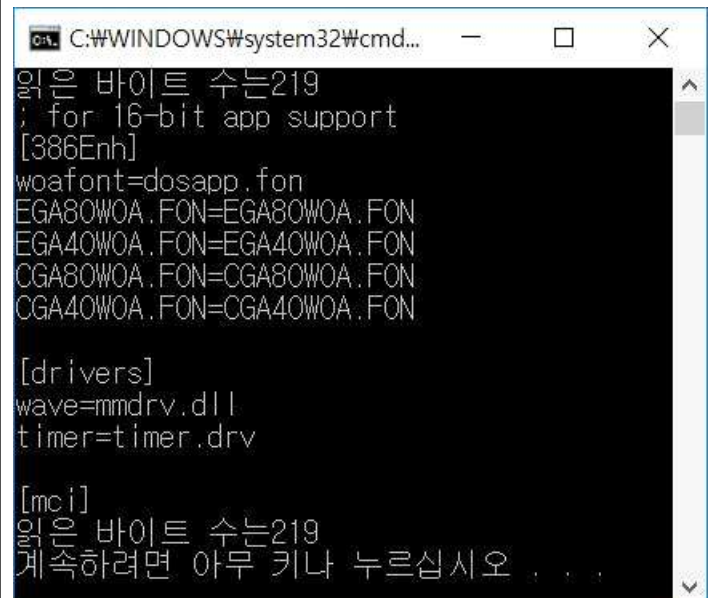
    ifstream fin2("system.ini", ios::binary);

    while ((c = fin2.get()) != EOF){
        // EOF : End Of File, 파일의 끝까지 문자 읽기
        cout << (char)c;
        count++;
    }

    cout << "읽은 바이트 수는" << count << endl;

    fin2.close();
}

```





## 17-04-19-File\_Get을 이용한 불러오기

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

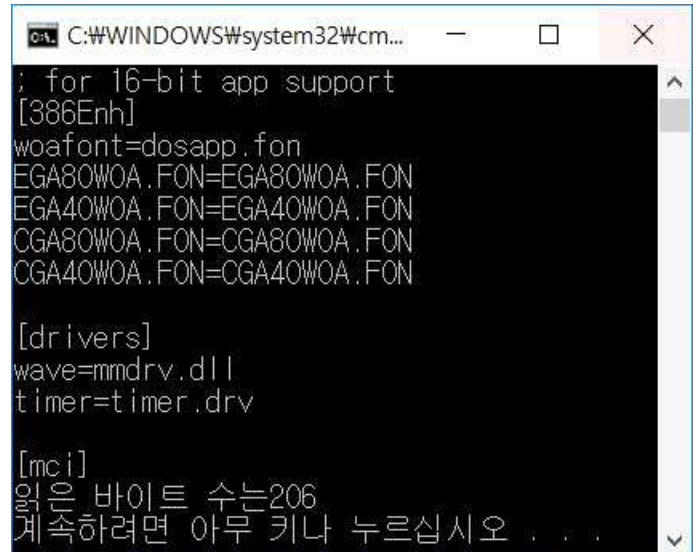
void main(){
    char* file = "c:\\windows\\system.ini";

    ifstream fin(file);
    if (!fin)
        cout << file << "열기 오류" << endl;

    int count = 0;
    int c; // char를 써도 된다.

    while ((c = fin.get()) != EOF){
        // EOF : End Of File, 파일의 끝까지 문자 읽기
        cout << (char)c;
        count++;
    }
    cout << "읽은 바이트 수는" << count << endl;

    fin.close(); // 파일 닫기
}
```



```
C:\WINDOWS\system32\cmd.exe
; for 16-bit app support
[386Enh]
woafont=dosapp.fon
EGA80W0A.FON=EGA80W0A.FON
EGA40W0A.FON=EGA40W0A.FON
CGA80W0A.FON=CGA80W0A.FON
CGA40W0A.FON=CGA40W0A.FON

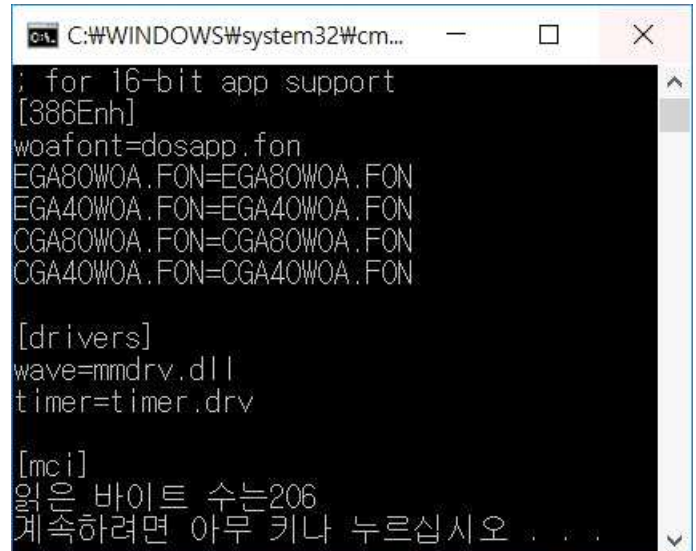
[drivers]
wave=mmdrv.dll
timer=timer.drv

[mci]
읽은 바이트 수는206
계속하려면 아무 키나 누르십시오 . . .
```

## 17-04-19-File\_Getline을 이용한 불러오기

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

int main() {
    ifstream fin("c:\\windows\\system.ini");
    if (!fin) { cout << "c:\\windows\\system.ini 열기
실패" << endl; return 0; }
    char buf[81]; // 한 라인이 최대 80개의 문자로 구
성된다고 가정
    while (true) {
        fin.getline(buf, 81); // 한 라인이 최대 80
개의문자로 구성
        if (fin.eof()) break; // 읽기 종료
        cout << buf << endl; // 라인 출력
    }
    fin.close();
}
```



```
C:\WINDOWS\system32\cmd.exe
; for 16-bit app support
[386Enh]
woafont=dosapp.fon
EGA80W0A.FON=EGA80W0A.FON
EGA40W0A.FON=EGA40W0A.FON
CGA80W0A.FON=CGA80W0A.FON
CGA40W0A.FON=CGA40W0A.FON

[drivers]
wave=mmdrv.dll
timer=timer.drv

[mci]
읽은 바이트 수는206
계속하려면 아무 키나 누르십시오 . . .
```



## 17-04-19-File\_Input\_Output

```

/*main.cpp*/
#include <fstream>
#include <iostream>
#include <string>
using namespace std;

void main(){
    ofstream fout; // 파일 쓰기 위한 스트림 생성
    ifstream fin; // 파일 읽기를 위한 스트림 생성

    /* 파일 쓰기 */
    fout.open("class.txt"); // 파일 열기, 없으면 자동으로 생성됨

    if (!fout) // 파일이 없을 경우
        cout << "파일 열기에 실패" << endl;

    string name, dept;
    int id;
    double gpa;

    cout << "이름 학과 입력 >> ";
    cin >> name >> dept;
    cout << "학번 학점 입력 >> ";
    cin >> id >> gpa;

    // 파일에 입력
    fout << name << endl;
    fout << dept << endl;
    fout << id << endl;
    fout << gpa << endl;

    fout.close(); // 파일 닫기

    string myname, mydept;
    int myid;
    double mygpa;

    /* 파일 읽기 */
    fin.open("class.txt");
    if (!fin)
        cout << "파일 열기에 실패" << endl;

    // 파일 읽기
    fin >> myname;
    fin >> mydept;
    fin >> myid;
    fin >> mygpa;

    fin.close(); // 파일 닫기

    cout << myname << endl;
    cout << mydept << endl;
    cout << myid << endl;
    cout << mygpa << endl;
}

```

## 17-04-19-File\_p\_test2

```

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

void main(){
    char* file = "c:\\windows\\system.ini";

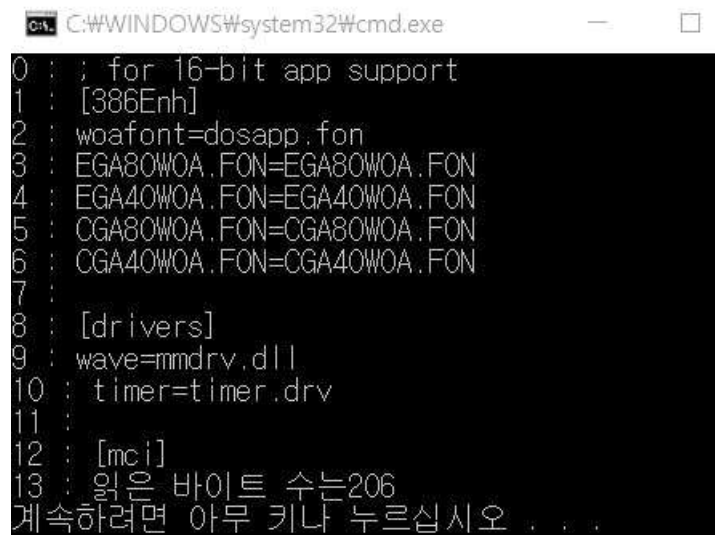
    ifstream fin(file);
    if (!fin)
        cout << file << "열기 오류" << endl;

    int count = 0;
    int n = 0;
    int c; // char를 써도 된다.

    //while (!fin.eof()){ // 이걸 하게 되면 eof의 값까지 읽어 들인다
    cout << n << " : ";
    while ((c = fin.get()) != EOF){
        // EOF : End Of File, 파일의 끝까지 문자 읽기
        cout << (char)c;
        if ((char)c == '\n'){
            n++;
            cout << n << " : ";
        }
        count++;
    }
    cout << "읽은 바이트 수는" << count << endl;

    fin.close(); // 파일 닫기
}

```



```

C:\WINDOWS\system32\cmd.exe
0 : ; for 16-bit app support
1 : [386Enh]
2 : woafont=dosapp.fon
3 : EGA80WOA.FON=EGA80WOA.FON
4 : EGA40WOA.FON=EGA40WOA.FON
5 : CGA80WOA.FON=CGA80WOA.FON
6 : CGA40WOA.FON=CGA40WOA.FON
7 :
8 : [drivers]
9 : wave=mmdrv.dll
10 : timer=timer.drv
11 :
12 : [mci]
13 : 읽은 바이트 수는206
계속하려면 아무 키나 누르십시오 . . .

```