REPORT



|  |  |
| --- | --- |
| 과 목 : | 객체지향프로그래밍 |
| 제출일자 : | 2023.05.23 |
| 담당교수 : | 황성호 |
| 학 과 : | 멀티디자인학과 |
| 학 번 : | 201522405 |
| 이 름 : | 최준하 |

1. (예제 8-1) point 클래스를 상속받는 ColorPoint 클래스 만들기.

#include <iostream>

#include <string>

using namespace std;

class point {

int x, y;

public:

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

void show\_point() {

cout << "(" << x << "," << y << ")" << endl;

}

};

class color\_point :public point {

string color;

public:

void set\_color(string color) { this->color = color; }

void show\_color\_point();

};

void color\_point::show\_color\_point() {

cout << color << ":";

show\_point();

}

int main() {

point p;

color\_point cp;

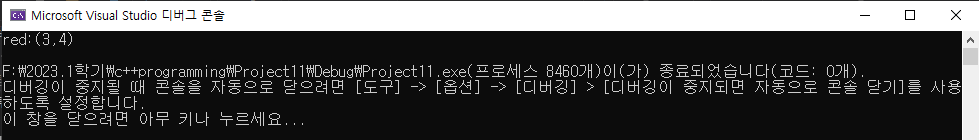
cp.set(3, 4);

cp.set\_color("red");

cp.show\_color\_point();

}

🡪출력 결과



2. (예제 8-3) tv, wide\_tv, smart\_tv의 상속 관계와 생성자 매개변수 전달.

#include <iostream>

#include <string>

using namespace std;

class tv {

int size;

public:

tv() { size = 20; }

tv(int size) { this->size = size; }

int get\_size() { return size; }

};

class wide\_tv :public tv {

bool video\_in;

public:

wide\_tv(int size, bool video\_in) :tv(size) {

this->video\_in = video\_in;

}

bool get\_video\_in() { return video\_in; }

};

class smart\_tv : public wide\_tv{

string ip\_add\_r;

public:

smart\_tv(string ip\_add\_r, int size) : wide\_tv(size, true) {

this->ip\_add\_r = ip\_add\_r;

}

string get\_ip\_add\_r() { return ip\_add\_r; }

};

int main() {

smart\_tv htv("192.0.0.1", 32);

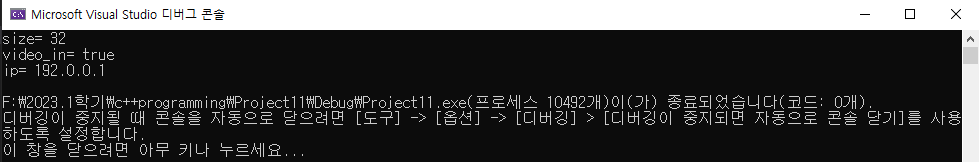
cout << "size= " << htv.get\_size() << endl;

cout << "video\_in= " << boolalpha << htv.get\_video\_in() << endl;

cout << "ip= " << htv.get\_ip\_add\_r() << endl;

}

🡪출력 결과



3. (예제 8-4) private 상속 사례.

#include <iostream>

using namespace std;

class base {

int a;

protected:

void set\_a(int a) { this->a = a; }

public:

void show\_a() { cout << a; }

};

class derived : private base {

int b;

protected:

void set\_b(int b) { this->b = b; }

public:

void show\_b() { cout << b; }

};

int main() {

derived x;

//x.a = 5;

//x.set\_a(10);

//x.show\_a();

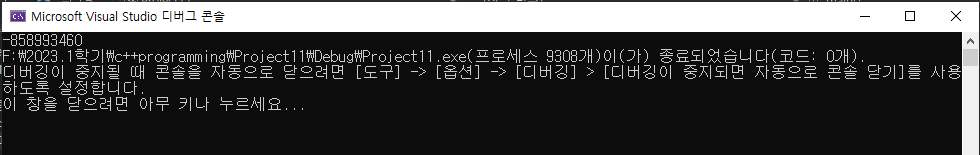
//x.b = 10;

//x.set\_b(10);

x.show\_b();

}

🡪출력 결과



4. (예제 8-5) protected 상속 사례.

#include <iostream>

using namespace std;

class base {

int a;

protected:

void set\_a(int a) { this->a = a; }

public:

void show\_a() { cout << a; }

};

class derived : protected base {

int b;

protected:

void set\_b(int b) { this->b = b; }

public:

void show\_b() { cout << b; }

};

int main() {

derived x;

//x.a = 5;

//x.set\_a(10);

//x.show\_a();

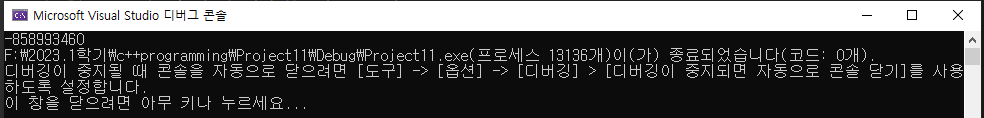
//x.b = 10;

//x.set\_b(10);

x.show\_b();

}

🡪출력 결과



5. (예제 8-6) 상속이 중첩될 때 접근 지정 사례.

#include <iostream>

using namespace std;

class base {

int a;

protected:

void set\_a(int a) { this->a = a; }

public:

void show\_a() { cout << a; }

};

class derived : private base {

int b;

protected:

void set\_b(int b) { this->b = b; }

public:

void show\_b() {

set\_a(5);

show\_a();

cout << b;

}

};

class grand\_derived : private derived {

int c;

protected:

void set\_ab(int x) {

set\_a(x);

show\_a();

set\_b(x);

}

};

🡪출력 결과

