REPORT



|  |  |
| --- | --- |
| 과 목 : | 객체지향프로그래밍 |
| 제출일자 : | 2023.04.18 |
| 담당교수 : | 황성호 |
| 학 과 : | 멀티디자인학과 |
| 학 번 : | 201522405 |
| 이 름 : | 최준하 |

1.(예제 5-1) '값에 의한 호출'시 매개 변수의 생성자 실행되지 않음.

#include <iostream>

using namespace std;

class \_circle {

private:

int radius;

public:

\_circle();

\_circle(int r);

~\_circle();

double get\_area() { return 3.14 \* radius \* radius; } // 사용 안됨

int get\_radius() { return radius; }

void set\_radius(int radius) { this->radius = radius; }

};

\_circle::\_circle() {

radius = 1;

cout << "생성자 실행 radius = " << radius << endl;

}

\_circle::\_circle(int radius) {

this->radius = radius;

cout << "생성자 실행 radius = " << radius << endl;

}

\_circle:: ~\_circle() {

cout << "소멸자 실행 radius = " << radius << endl;

}

void increase(\_circle c) {

int r = c.get\_radius();

c.set\_radius(r + 1);

}

int main() {

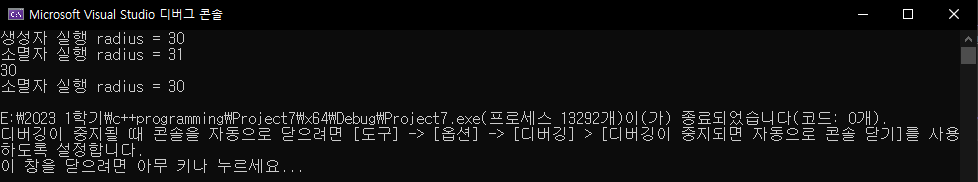
\_circle waffle(30);

increase(waffle);

cout << waffle.get\_radius() << endl;

}

🡪출력 결과



2.(예제 5-2) 문제를 쓰시오.

#include <iostream>

using namespace std;

class \_circle {

int radius;

public:

\_circle() { radius = 1; }

\_circle(int radius) { this->radius = radius; } // == \_circle.radius

void set\_radius(int radius) { this->radius = radius; }

double get\_area() { return 3.14 \* radius \* radius; }

};

\_circle get\_circle() {

\_circle tmp(30);

return tmp;

}

int main() {

\_circle c; //객체 생성

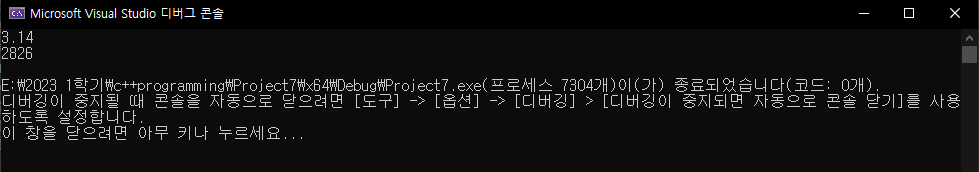
cout << c.get\_area() << endl;

c = get\_circle(); //tmp가 복사됨

cout << c.get\_area() << endl;

}

🡪출력 결과



3.(예제 5-3) 문제를 쓰시오.

🡪출력 결과

4.(예제 5-4) 문제를 쓰시오.

🡪출력 결과