REPORT



|  |  |
| --- | --- |
| 과 목 : | 객체지향프로그래밍 |
| 제출일자 : | 2023.05.09 |
| 담당교수 : | 황성호 |
| 학 과 : | 멀티디자인학과 |
| 학 번 : | 201522405 |
| 이 름 : | 최준하 |

1. (예제 6-2) sum() 함수 중복 연습.

#include <iostream>

using namespace std;

int sum(int a, int b) {

int s = 0;

for (int i =a; i <= b; i++)

{

s += i;

}

return s;

}

int sum(int a) {

int s = 0;

for (int i = 0; i <= a; i++)

{

s += i;

}

return s;

}

int main() {

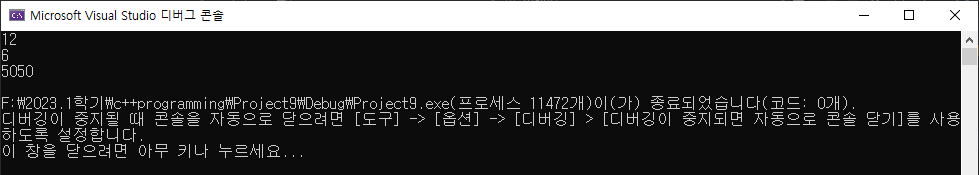
cout << sum(3, 5) << endl;

cout << sum(3) << endl;

cout << sum(100) << endl;

}

🡪출력 결과



2. (예제 6-4) 디폴트 매개 변수를 가진 함수 만들기 연습.

#include <iostream>

using namespace std;

void f(char c = ' ', int line = 1); //함수 원형

void f(char c, int line) {

for (int i = 0; i < line; i++)

{

for (int j = 0; j < 10; j++)

{

cout << c;

}

cout << endl;

}

}

int main() {

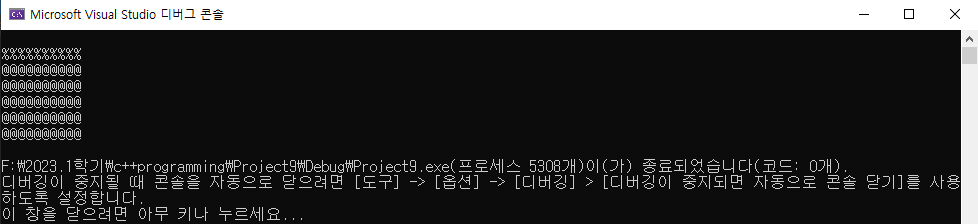
f();

f('%');

f('@', 5);

}

🡪출력 결과



3. (예제 6-6) 중복된 생성자들을 디폴트 매개 변수를 이용한 간소화 연습.

#include <iostream>

using namespace std;

class my\_vector {

int\* p;

int size;

public:

my\_vector(int n = 100) {

p = new int[n];

size = n;

}

~my\_vector() { delete[] p; }

};

int main() {

my\_vector\* v1, \* v2;

v1 = new my\_vector(); //디폴트 100

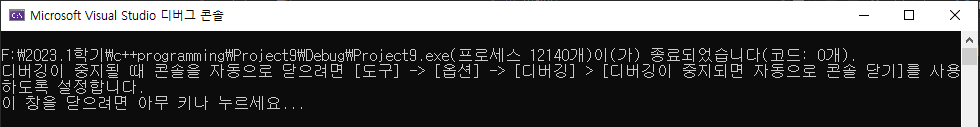
v2 = new my\_vector(1024);

delete v1;

delete v2;

}

🡪출력 결과



4. (예제 6-9) 디폴트 매개 변수로 인한 함수 중복의 모호성.

#include <iostream>

#include <string>

using namespace std;

void msg(int id) {

cout << id << endl;

}

void msg(int id, string s = "") {

cout << id << ":" << s << endl;

}

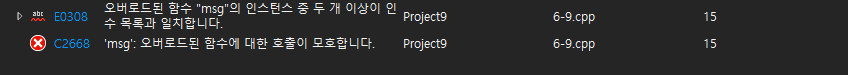
int main() {

msg(5, "good morning");

msg(6);

}

🡪출력 결과



5. (예제 6-10) static 맴버를 가진 math 클라스 작성.

#include <iostream>

using namespace std;

class \_math {

public:

static int abs(int a) { return a > 0 ? a : -a; } //a가 0보다 크면 t =a, f =-a

static int max(int a, int b) { return (a > b) ? a : b; }

static int min(int a, int b) { return (a > b) ? b : a; }

};

int main() {

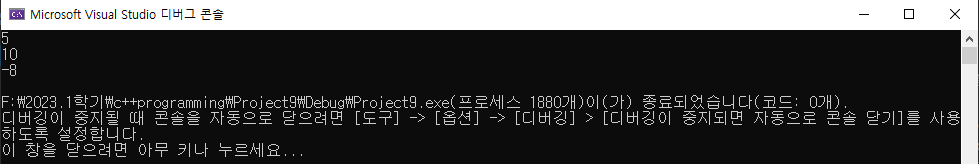
cout << \_math::abs(-5) << endl;

cout << \_math::max(10, 8) << endl;

cout << \_math::min(-3, -8) << endl;

}

🡪출력 결과



6. (예제 6-11) static 맴버를 공유의 목적으로 사용하는 예.

#include <iostream>

using namespace std;

class \_circle {

private:

static int num\_of\_circles;

int radius;

public:

\_circle(int r = 1);

~\_circle() { num\_of\_circles--; }

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

static int get\_num\_of\_circles() { return num\_of\_circles; }

};

\_circle::\_circle(int r) {

radius = r;

num\_of\_circles++;

}

int \_circle::num\_of\_circles = 0; //0 으로 초기화

int main() {

\_circle\* p = new \_circle[10];

cout << "생존하고 있는 원의 개수 = " << \_circle::get\_num\_of\_circles() << endl;

delete[] p;

cout << "생존하고 있는 원의 개수 = " << \_circle::get\_num\_of\_circles() << endl;

\_circle a;

cout << "생존하고 있는 원의 개수 = " << \_circle::get\_num\_of\_circles() << endl;

\_circle b;

cout << "생존하고 있는 원의 개수 = " << \_circle::get\_num\_of\_circles() << endl;

}

🡪출력 결과

