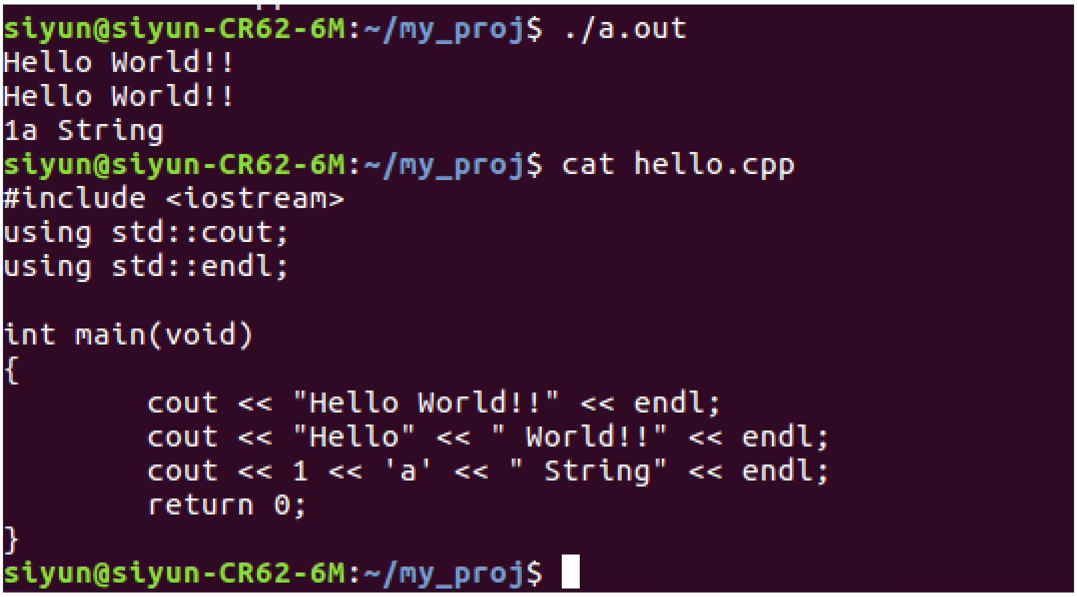
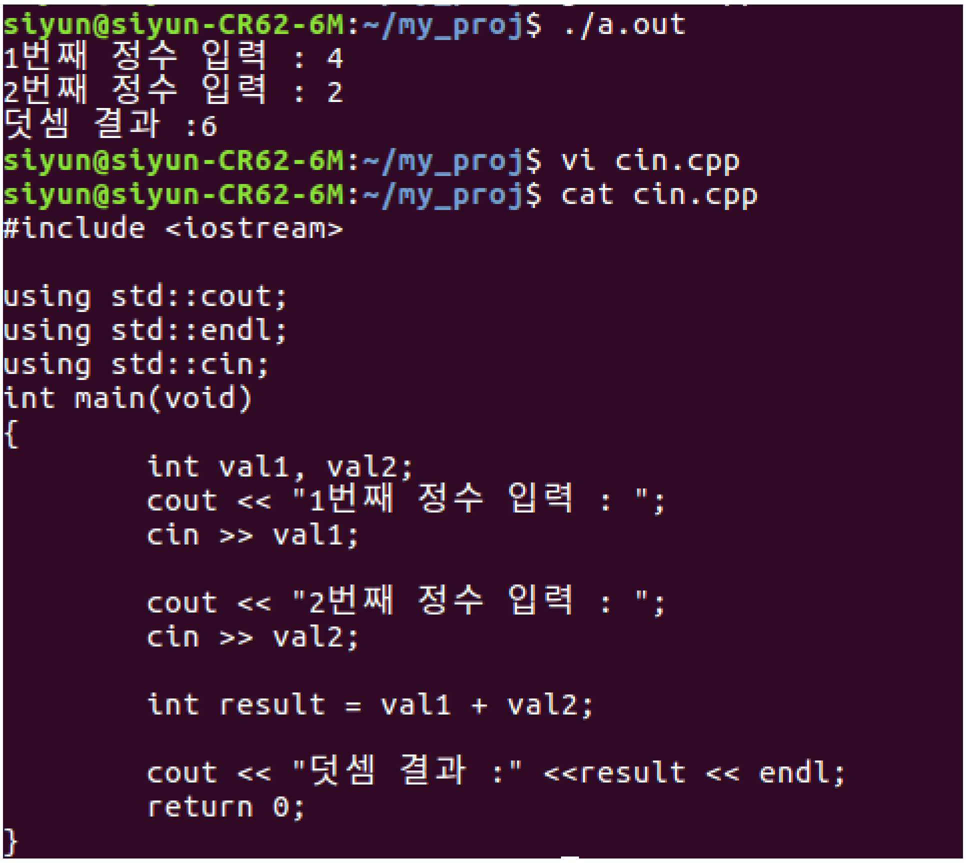
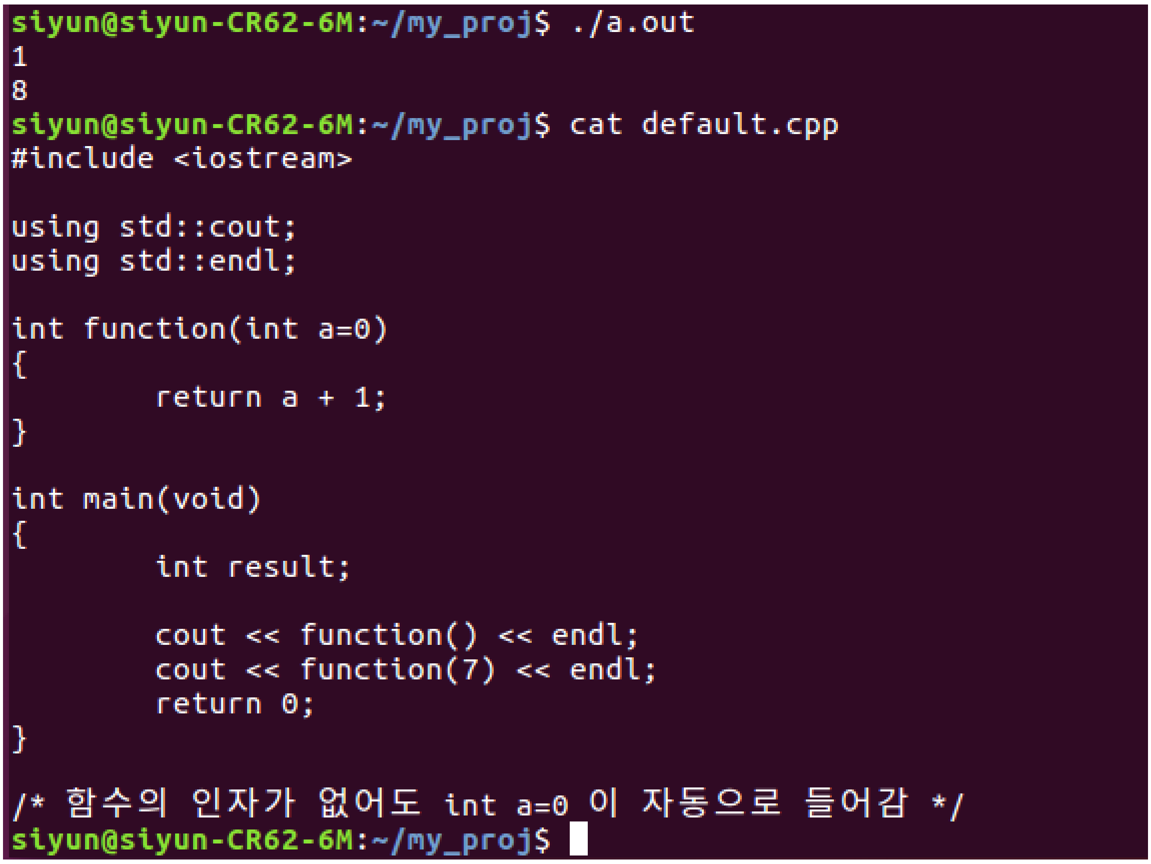


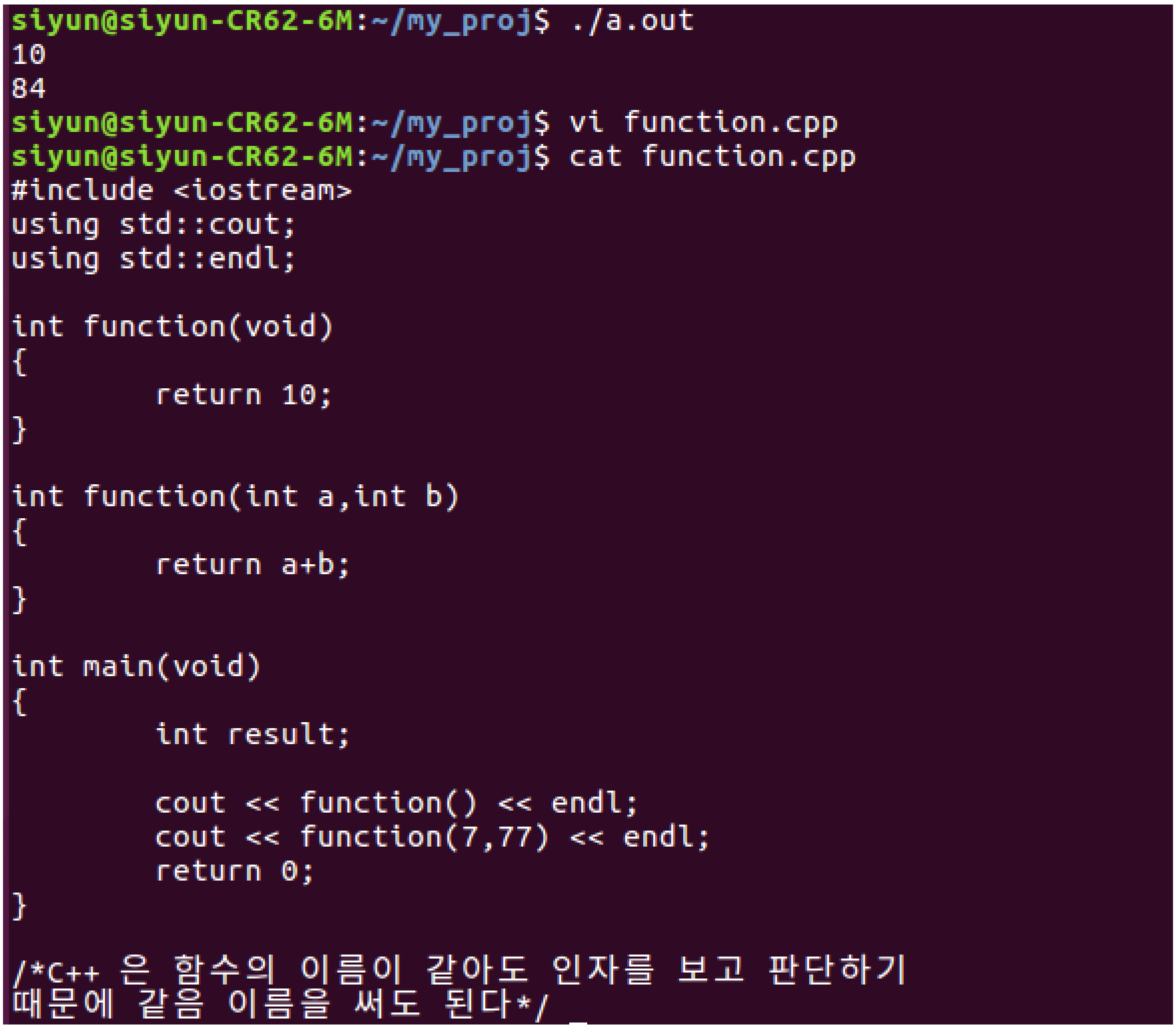
```
siyun@siyun-CR62-6M:~/my_proj$ ./a.out
Hello World!!
Hello World!!
1a String
siyun@siyun-CR62-6M:~/my_proj$ cat hello.cpp
#include <iostream>
using std::cout;
using std::endl;

int main(void)
{
        cout << "Hello World!!" << endl;
        cout << "Hello" << " World!!" << endl;
        cout << 1 << 'a' << " String" << endl;
        return 0;
}
siyun@siyun-CR62-6M:~/my_proj$ 
```



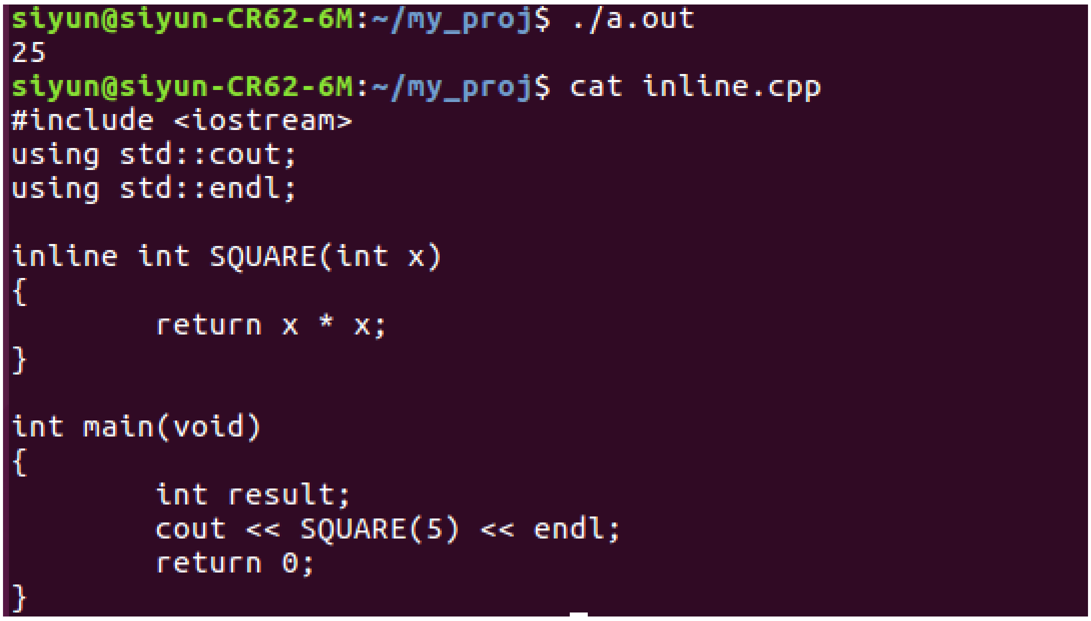
```
siyun@siyun-CR62-6M:~/my_proj$ ./a.out
1번째 정수 입력 : 4
2번째 정수 입력 : 2
덧셈 결과 :6
siyun@siyun-CR62-6M:~/my_proj$ vi cin.cpp
siyun@siyun-CR62-6M:~/my_proj$ cat cin.cpp
#include <iostream>

using std::cout;
using std::endl;
using std::cin;
int main(void)
{
        int val1, val2;
        cout << "1번째 정수 입력 : ";
        cin >> val1;

        cout << "2번째 정수 입력 : ";
        cin >> val2;

        int result = val1 + val2;

        cout << "덧셈 결과 :" <<result << endl;
        return 0;
}
```



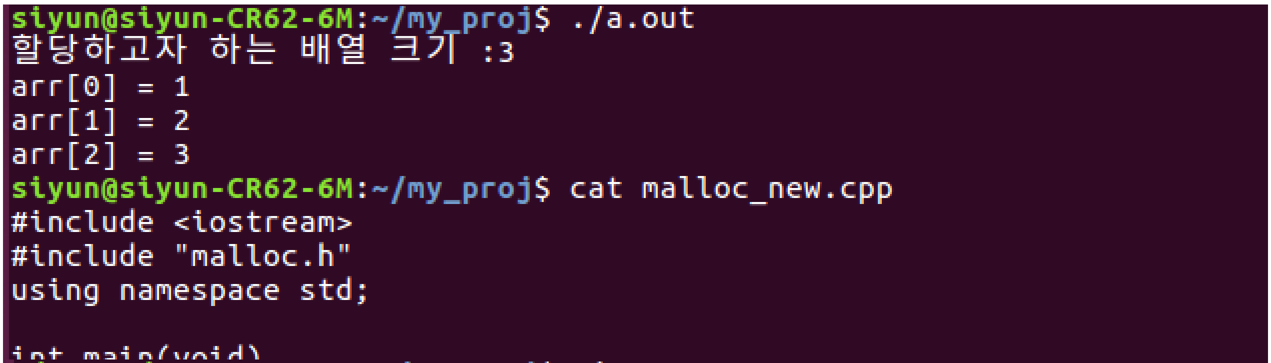
```
siyun@siyun-CR62-6M:~/my_proj$ ./a.out
1
8
siyun@siyun-CR62-6M:~/my_proj$ cat default.cpp
#include <iostream>

using std::cout;
using std::endl;

int function(int a=0)
{
        return a + 1;
}

int main(void)
{
        int result;

        cout << function() << endl;
        cout << function(7) << endl;
        return 0;
}

/* 함수의 인자가 없어도 int a=0 이 자동으로 들어감 */
siyun@siyun-CR62-6M:~/my_proj$ 
```



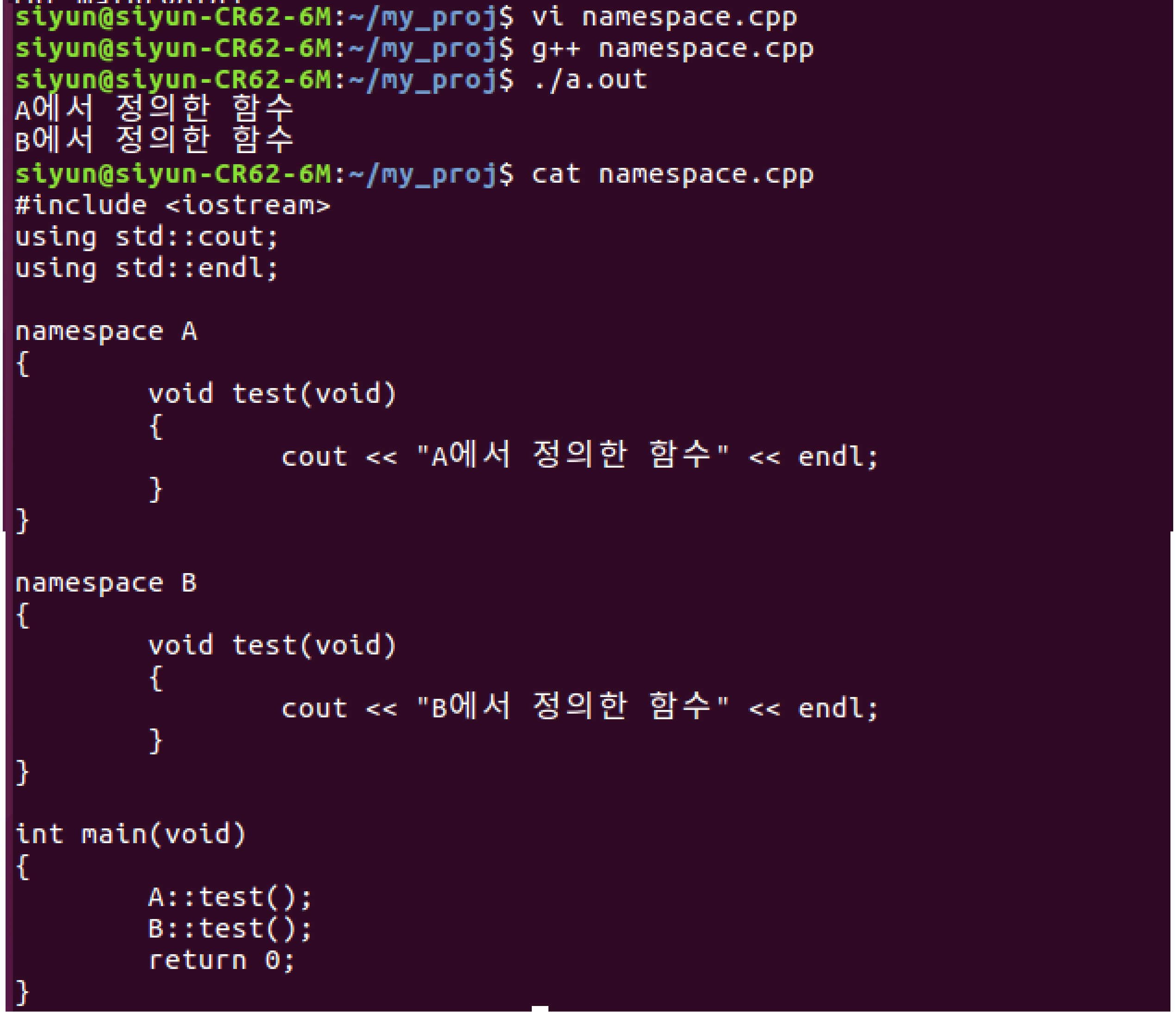
```
siyun@siyun-CR62-6M:~/my_proj$ ./a.out
10
84
siyun@siyun-CR62-6M:~/my_proj$ vi function.cpp
siyun@siyun-CR62-6M:~/my_proj$ cat function.cpp
#include <iostream>
using std::cout;
using std::endl;

int function(void)
{
        return 10;
}

int function(int a,int b)
{
        return a+b;
}

int main(void)
{
        int result;

        cout << function() << endl;
        cout << function(7,77) << endl;
        return 0;
}

/*C++ 은 함수의 이름이 같아도 인자를 보고 판단하기
때문에 같음 이름을 써도 된다*/
```



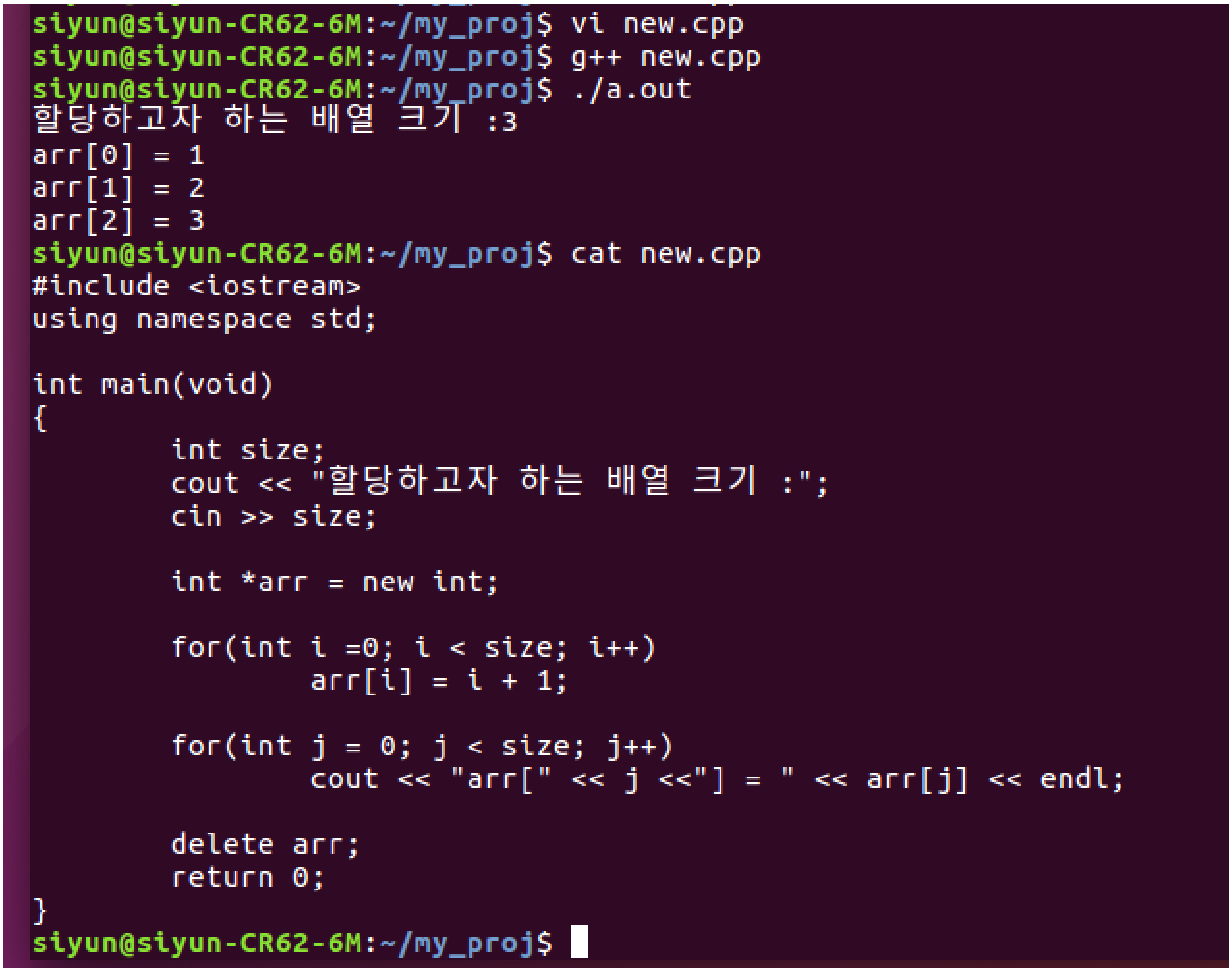
```
siyun@siyun-CR62-6M:~/my_proj$ ./a.out
25
siyun@siyun-CR62-6M:~/my_proj$ cat inline.cpp
#include <iostream>
using std::cout;
using std::endl;

inline int SQUARE(int x)
{
        return x * x;
}

int main(void)
{
        int result;
        cout << SQUARE(5) << endl;
        return 0;
}
```



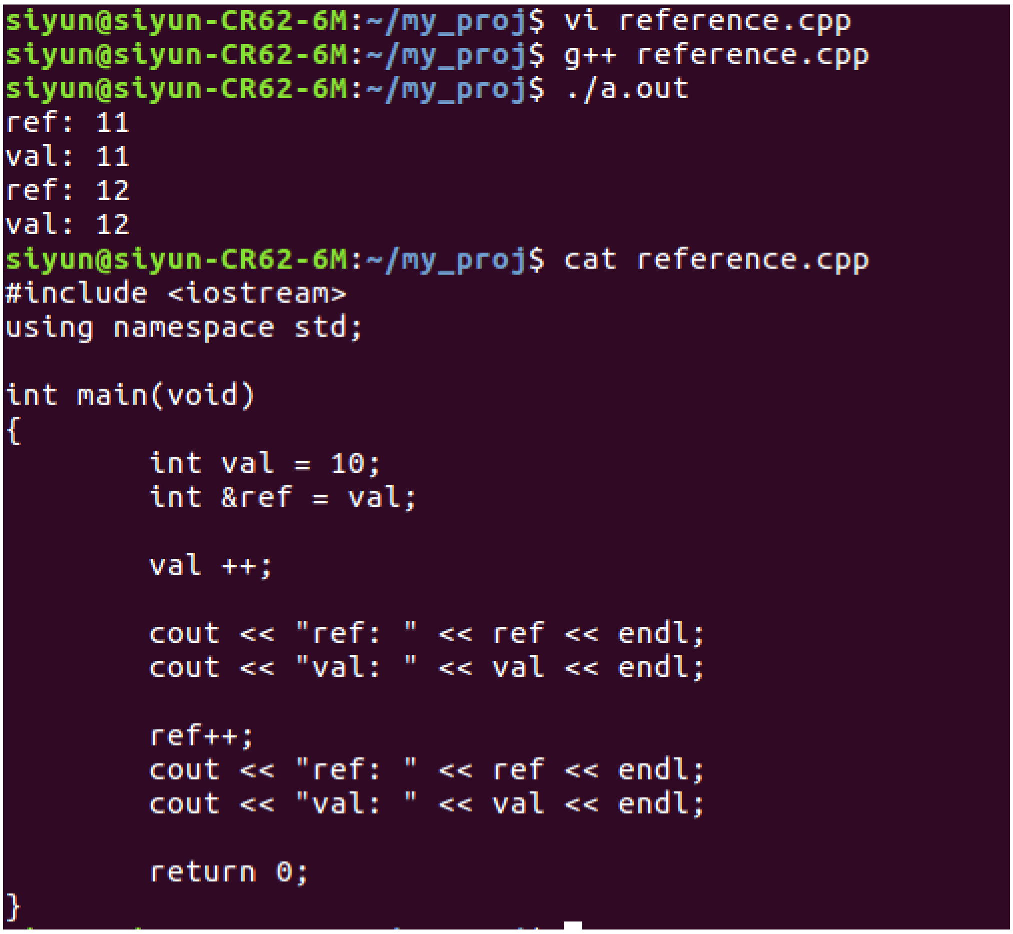
```
siyun@siyun-CR62-6M:~/my_proj$ ./a.out
할당하고자 하는 배열 크기 :3
arr[0] = 1
arr[1] = 2
arr[2] = 3
siyun@siyun-CR62-6M:~/my_proj$ cat malloc_new.cpp
#include <iostream>
#include "malloc.h"
using namespace std;

int main(void)
```



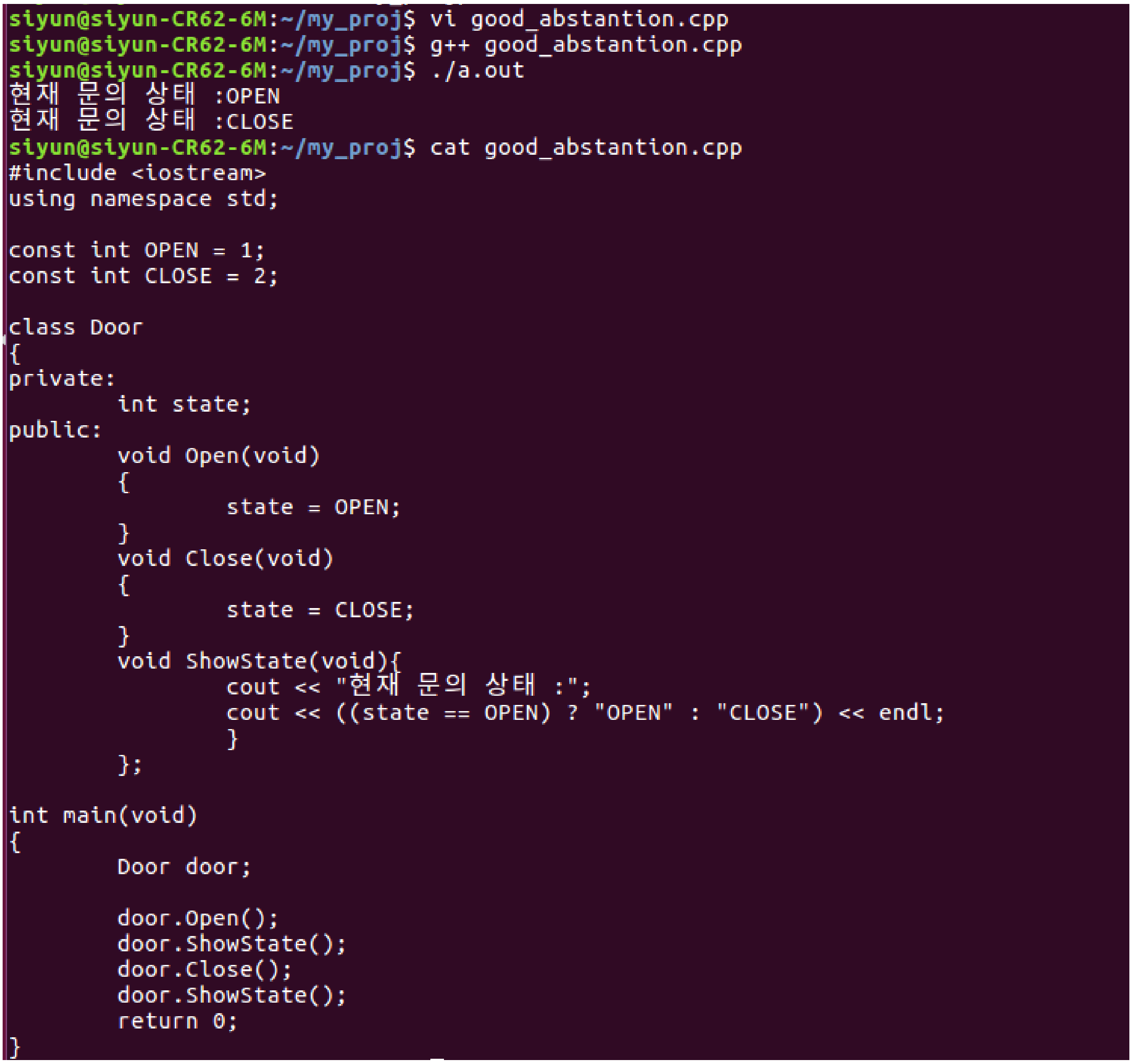
```
siyun@siyun-CR62-6M:~/my_proj$ vi namespace.cpp
siyun@siyun-CR62-6M:~/my_proj$ g++ namespace.cpp
siyun@siyun-CR62-6M:~/my_proj$ ./a.out
A에서 정의한 함수
B에서 정의한 함수
siyun@siyun-CR62-6M:~/my_proj$ cat namespace.cpp
#include <iostream>
using std::cout;
using std::endl;

namespace A
{
        void test(void)
        {
                cout << "A에서 정의한 함수" << endl;
        }
}

namespace B
{
        void test(void)
        {
                cout << "B에서 정의한 함수" << endl;
        }
}

int main(void)
{
        A::test();
        B::test();
        return 0;
}
```



```
siyun@siyun-CR62-6M:~/my_proj$ vi new.cpp
siyun@siyun-CR62-6M:~/my_proj$ g++ new.cpp
siyun@siyun-CR62-6M:~/my_proj$ ./a.out
할당하고자 하는 배열 크기 :3
arr[0] = 1
arr[1] = 2
arr[2] = 3
siyun@siyun-CR62-6M:~/my_proj$ cat new.cpp
#include <iostream>
using namespace std;

int main(void)
{
        int size;
        cout << "할당하고자 하는 배열 크기 :";
        cin >> size;

        int *arr = new int;

        for(int i =0; i < size; i++)
                arr[i] = i + 1;

        for(int j = 0; j < size; j++)
                cout << "arr[" << j <<"] = " << arr[j] << endl;

        delete arr;
        return 0;
}
siyun@siyun-CR62-6M:~/my_proj$ 
```



```
siyun@siyun-CR62-6M:~/my_proj$ vi reference.cpp
siyun@siyun-CR62-6M:~/my_proj$ g++ reference.cpp
siyun@siyun-CR62-6M:~/my_proj$ ./a.out
ref: 11
val: 11
ref: 12
val: 12
siyun@siyun-CR62-6M:~/my_proj$ cat reference.cpp
#include <iostream>
using namespace std;

int main(void)
{
        int val = 10;
        int &ref = val;

        val ++;

        cout << "ref: " << ref << endl;
        cout << "val: " << val << endl;

        ref++;
        cout << "ref: " << ref << endl;
        cout << "val: " << val << endl;

        return 0;
}
```



```
siyun@siyun-CR62-6M:~/my_proj$ vi good_abstantion.cpp
siyun@siyun-CR62-6M:~/my_proj$ g++ good_abstantion.cpp
siyun@siyun-CR62-6M:~/my_proj$ ./a.out
현재 문의 상태 :OPEN
현재 문의 상태 :CLOSE
siyun@siyun-CR62-6M:~/my_proj$ cat good_abstantion.cpp
#include <iostream>
using namespace std;

const int OPEN = 1;
const int CLOSE = 2;

class Door
{
private:
        int state;
public:
        void Open(void)
        {
                state = OPEN;
        }
        void Close(void)
        {
                state = CLOSE;
        }
        void ShowState(void){
                cout << "현재 문의 상태 :";
                cout << ((state == OPEN) ? "OPEN" : "CLOSE") << endl;
                }
        };

int main(void)
{
        Door door;

        door.Open();
        door.ShowState();
        door.Close();
        door.ShowState();
        return 0;
}
```

```
Car.h

#ifndef __CAR_H__
#define __CAR_H__

class car{
private:
        int velocity;
        char color[32];
        char direction[32];

public:
        void input_data(void);
        void print_data(void);
};

#endif
```

```
main_car.cpp

#include "car.h"

int main(void)
{
        car ds;
        ds.input_data();
        ds.print_data();
        return 0;
}

/* C++ 은 동작 중간에 변수 추가 가능 */
```

```
Car.cpp

#include <iostream>
#include "car.h"

using namespace std;

void car::input_data(void)
{
        cout << "velocity color direction" << endl;
        cin >> velocity;
        cin >> color;
        cin >> direction;

}


void car::print_data(void)
{
        cout << "velocity = " << velocity << "Km/h" << endl;
        cout << "color = " << color << " 색"   << endl;
        cout << "direction =" << direction << endl;

}
```

Xilinx Zynq FPGA, TI DSP, MCU 기반의 프로그래밍 및 회로 설계 전문가 과정

#69

강사 : Innova Lee(이 상훈)
학생 : 김 시윤