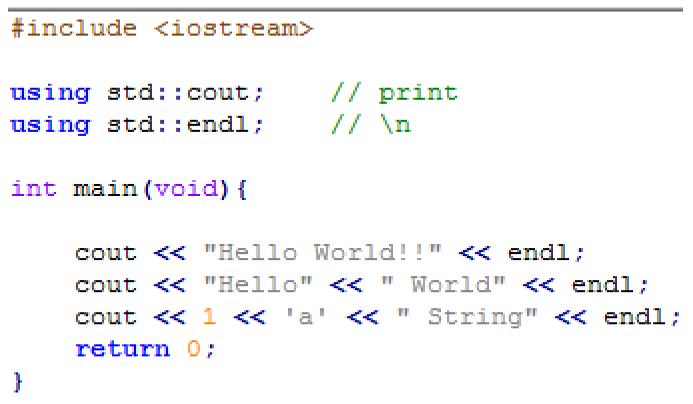
TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

강사 – Innova Lee(이상훈)
gcccompil3r@gmail.com
학생 – GJ (박현우)
uc820@naver.com

목차

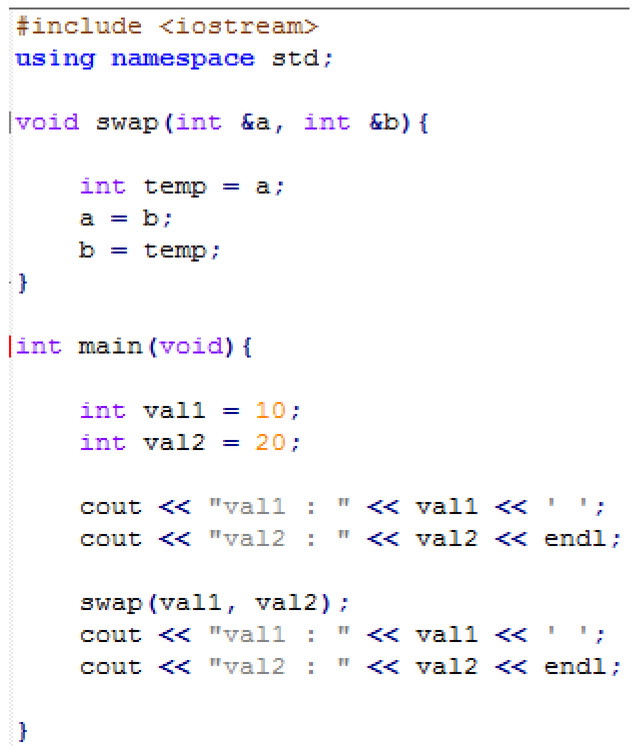
1) 기본 문법 1

hello



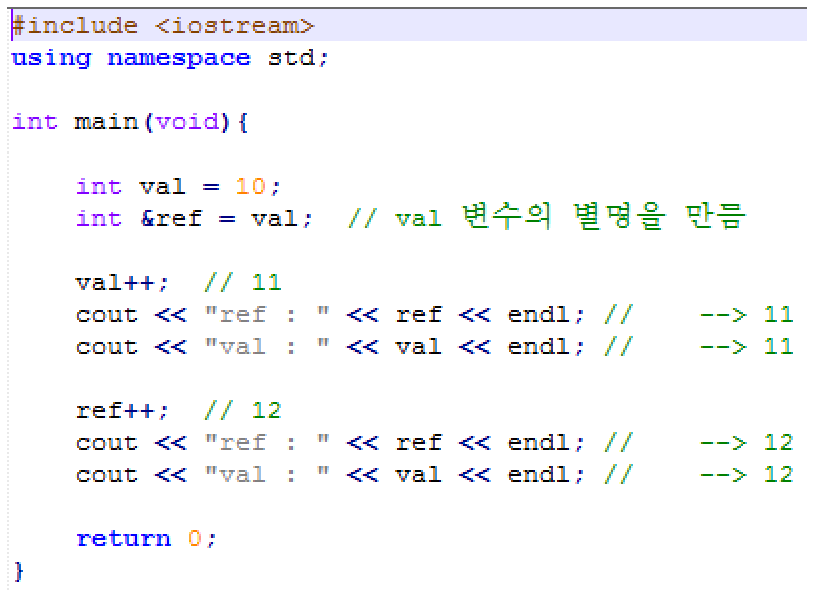
```cpp
#include <iostream>

using std::cout;    // print
using std::endl;    // \n

int main(void){

    cout << "Hello World!!" << endl;
    cout << "Hello" << " World" << endl;
    cout << 1 << 'a' << " String" << endl;
    return 0;
}
```

reference



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

void swap(int &a, int &b){

    int temp = a;
    a = b;
    b = temp;
}

int main(void){

    int val1 = 10;
    int val2 = 20;

    cout << "val1 : " << val1 << ' ';
    cout << "val2 : " << val2 << endl;

    swap(val1, val2);
    cout << "val1 : " << val1 << ' ';
    cout << "val2 : " << val2 << endl;

}
```

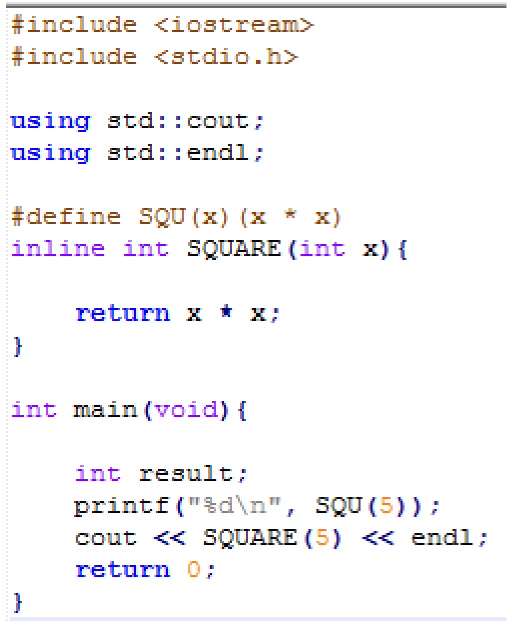
reference



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

int main(void){

    int val = 10;
    int &ref = val;  // val 변수의 별명을 만듬

    val++;  // 11
    cout << "ref : " << ref << endl; //    --> 11
    cout << "val : " << val << endl; //    --> 11

    ref++;  // 12
    cout << "ref : " << ref << endl; //    --> 12
    cout << "val : " << val << endl; //    --> 12

    return 0;
}
```

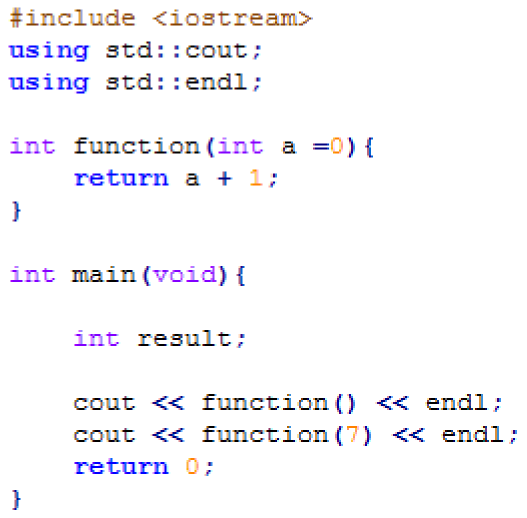
1) 기본 문법 2

inline



```cpp
#include <iostream>
#include <stdio.h>

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

#define SQU(x) (x * x)
inline int SQUARE(int x){

    return x * x;
}

int main(void){

    int result;
    printf("%d\n", SQU(5));
    cout << SQUARE(5) << endl;
    return 0;
}
```

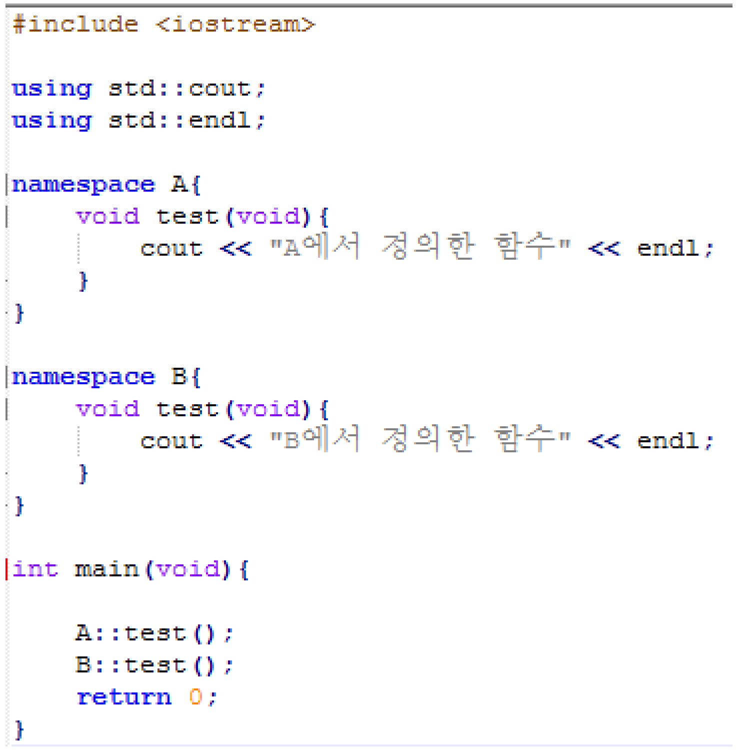
default



```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;
}
```

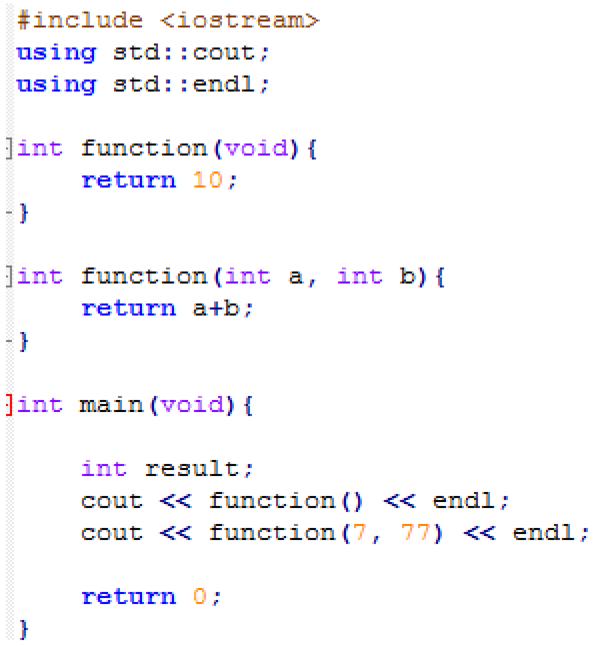
1) 기본 문법 3

namespace



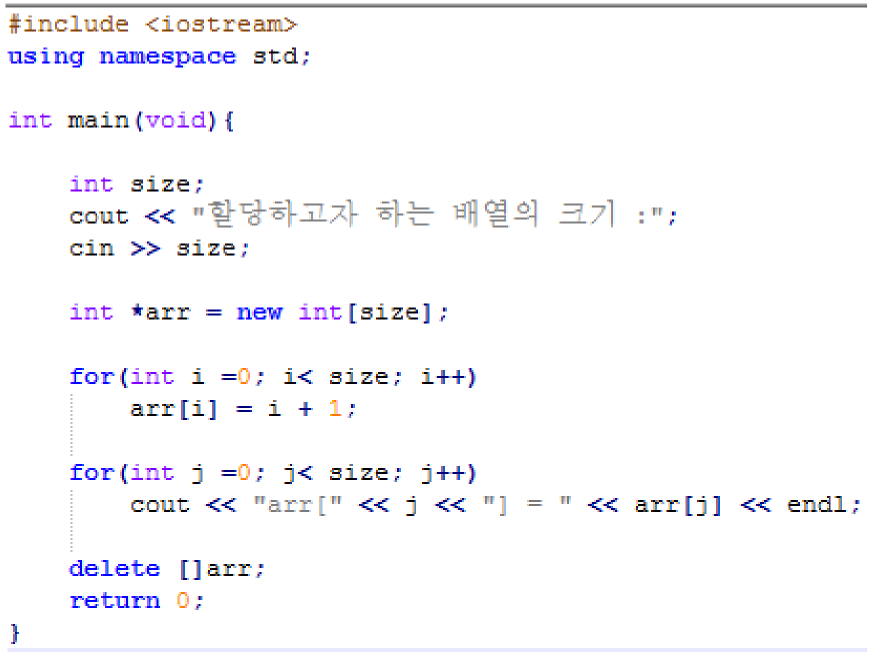
```cpp
#include <iostream>

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;
}
```

overloading



```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;
}
```

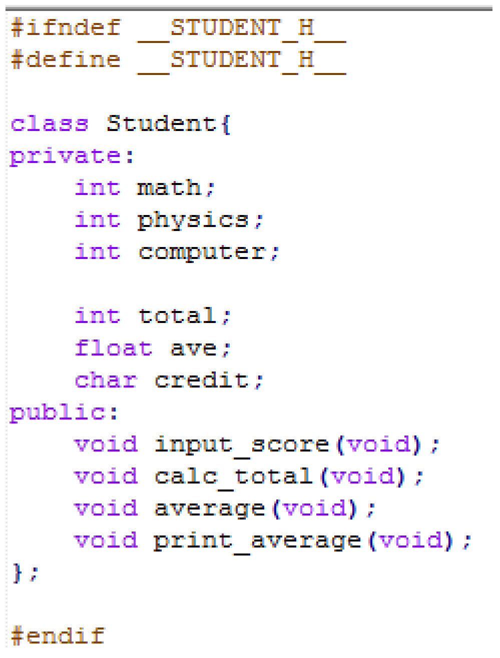
new



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

int main(void){

	int size;
	cout << "할당하고자 하는 배열의 크기 :";
	cin >> size;

	int *arr = new int[size];

	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;
}
```

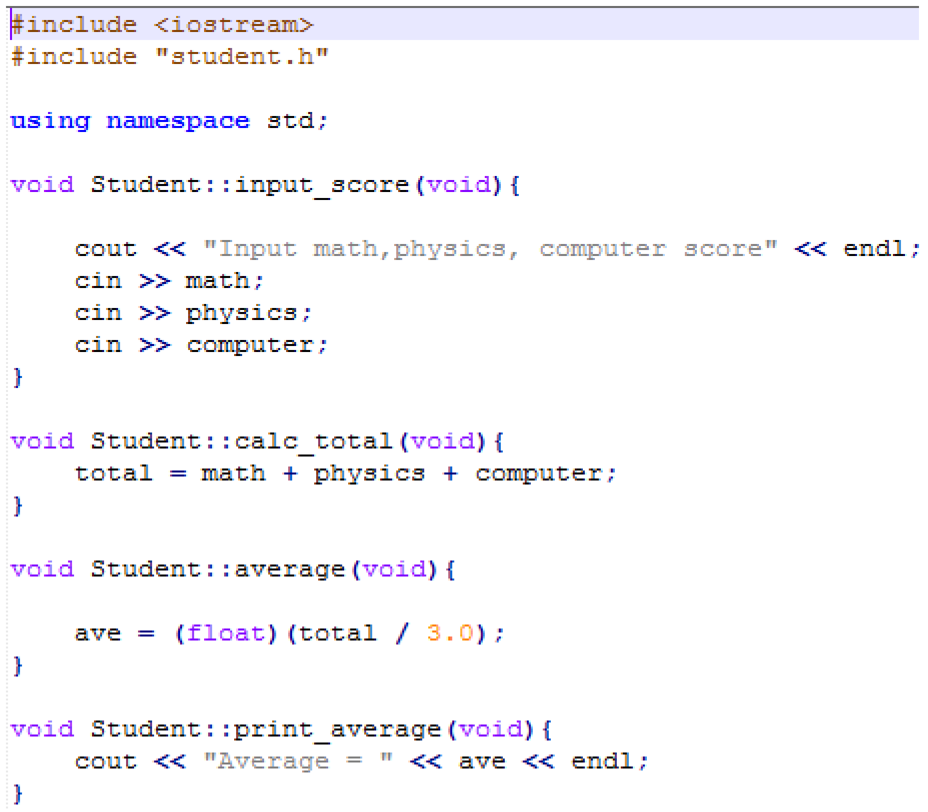
2) Class & Good abstraction

Header



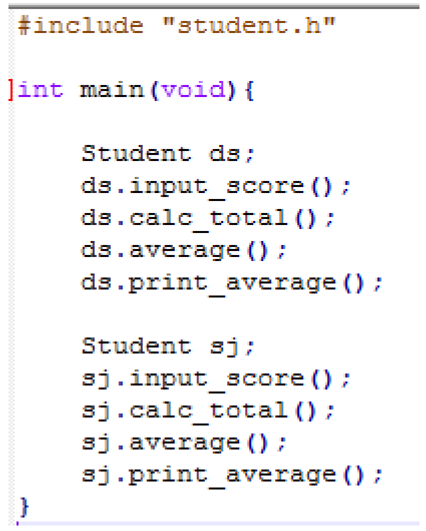
```cpp
#ifndef __STUDENT_H__
#define __STUDENT_H__

class Student{
private:
    int math;
    int physics;
    int computer;

    int total;
    float ave;
    char credit;
public:
    void input_score(void);
    void calc_total(void);
    void average(void);
    void print_average(void);
};

#endif
```

Student class



```cpp
#include <iostream>
#include "student.h"

using namespace std;

void Student::input_score(void){

    cout << "Input math,physics, computer score" << endl;
    cin >> math;
    cin >> physics;
    cin >> computer;
}

void Student::calc_total(void){
    total = math + physics + computer;
}

void Student::average(void){

    ave = (float)(total / 3.0);
}

void Student::print_average(void){
    cout << "Average = " << ave << endl;
}
```

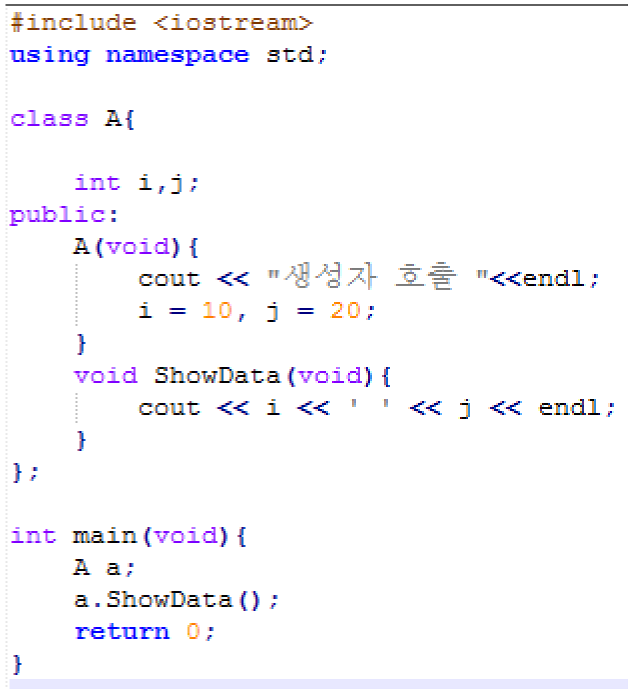
main



```cpp
#include "student.h"

int main(void){

    Student ds;
    ds.input_score();
    ds.calc_total();
    ds.average();
    ds.print_average();

    Student sj;
    sj.input_score();
    sj.calc_total();
    sj.average();
    sj.print_average();
}
```

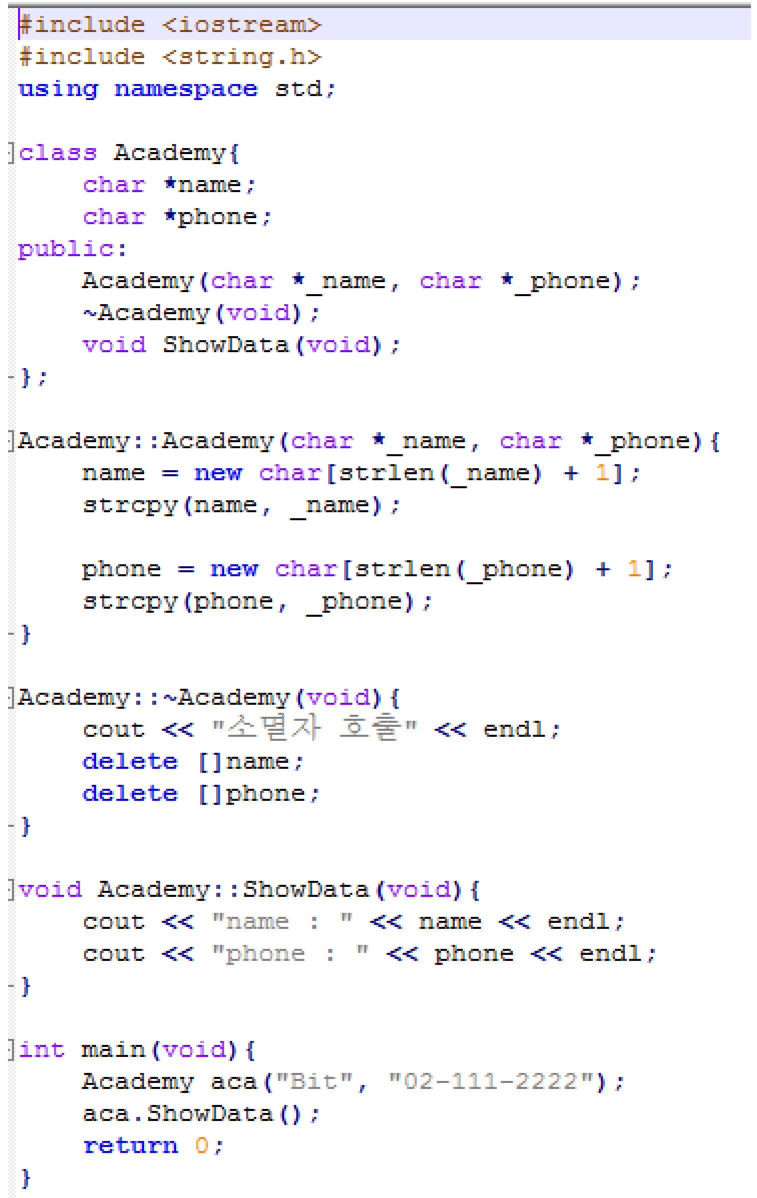
3) Constructor & deconstructor

constructor



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

class A{

    int i,j;
public:
    A(void){
        cout << "생성자 호출 "<<endl;
        i = 10, j = 20;
    }
    void ShowData(void){
        cout << i << ' ' << j << endl;
    }
};

int main(void){
    A a;
    a.ShowData();
    return 0;
}
```

deconstructor



```cpp
#include <iostream>
#include <string.h>
using namespace std;

class Academy{
    char *name;
    char *phone;
public:
    Academy(char *_name, char *_phone);
    ~Academy(void);
    void ShowData(void);
};

Academy::Academy(char *_name, char *_phone){
    name = new char[strlen(_name) + 1];
    strcpy(name, _name);

    phone = new char[strlen(_phone) + 1];
    strcpy(phone, _phone);
}

Academy::~Academy(void){
    cout << "소멸자 호출" << endl;
    delete []name;
    delete []phone;
}

void Academy::ShowData(void){
    cout << "name : " << name << endl;
    cout << "phone : " << phone << endl;
}

int main(void){
    Academy aca("Bit", "02-111-2222");
    aca.ShowData();
    return 0;
}
```