

# **TI DSP, MCU 및 Xilinx Zynq FPGA**

## **프로그래밍 전문가 과정**

**강사 - Innova Lee(이상훈)**  
**[gcccompil3r@gmail.com](mailto:gcccompil3r@gmail.com)**  
**학생 - 하성용**  
**[accept0108@naver.com](mailto:accept0108@naver.com)**

69 일차

c++쓰는이유

c는 쓰고싶는데 속도를 빠르게하고싶을때

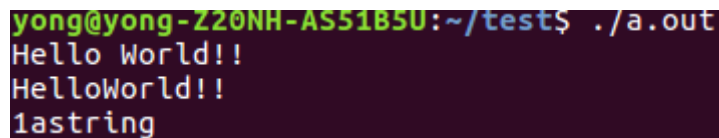
vi 확장자는 cpp

실행은 gcc가 아닌 g++

## 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;
}
```



```
yong@yong-Z20NH-AS51B5U:~/test$ ./a.out
Hello World!!
HelloWorld!!
1astring
```

c++에서는 iostream(아이오스트림)이 c의 stdio.h(스트디오 h)랑 같은역할

cout은 출력.

cout<<"Hello World!!" //Hello World를 출력하라

c의 printf는 c++에서의 endl

<< //꼭쉬는 연산자 집어넣으라는것 → 쉬프트연산자아님

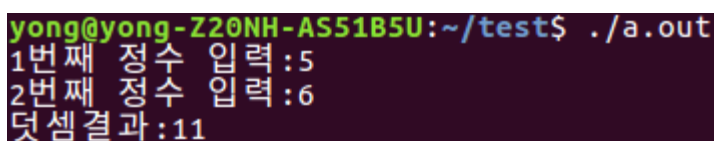
-----  
cin이란? c의 scanf

```
#include<iostream>
using std::cout;
using std::endl;
using std::cin;

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

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

    int result = val1+val2;
    cout<<"덧셈결과:"<<result<<endl;
    return 0;
}
```



```
yong@yong-Z20NH-AS51B5U:~/test$ ./a.out
1번째 정수 입력:5
2번째 정수 입력:6
덧셈결과:11
```

---

메인 멘들링이라고 함수이름이 똑같으면  
c는 이렇게하면 문법오류  
c++은됨  
c++컴파일러가 인자의갯수를 봄  
이거 둘이다른것인데 뒤에 서픽스를(점미사)를 하나붙여줌  
외부적으로 같지만 내부적으로는 이름이 달라짐  
c랑 개념은 같음 자동으로 해줄뿐

```
#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;
}
```

```
yong@yong-Z20NH-AS51BSU:~/test$ ./a.out
10
84
```

---

디폴트 파라미터란?  
함수에 보면 인자가있음  
인자가 디폴트에 들어간다는거  
int a=0 인부분에 0이 자동으로 들어감

```
#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;
}
```

```
yong@yong-Z20NH-AS51BSU:~/test$ ./a.out
1
8
```

-----

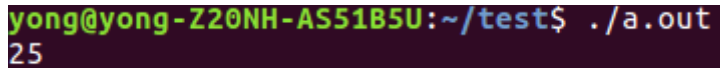
inline(인라인)이란?

기계어분석에서 인라인을 쓰면 스택프레임이 사라짐  
함수호출할때 스택프레임이 사라짐  
함수사이즈가크면 인라인하는 의미가없음

```
#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;
}
```



```
yong@yong-Z20NH-AS51B5U:~/test$ ./a.out
25
```

-----

namespace(네임스페이스) 쓰느이유  
여러분이 이름이 같은 함수를 만들었는데  
기능이 다를수도있고 같을수도있고 무엇이 더성능이 좋은지 비교해보고싶을때  
이름이 같으면 구동이 안됨

근데 namespace(네임스페이스를) 쓰면 구동이됨  
네임스페이스가 이름공간이라는건데 방식이 특이  
콜론이 두개오고 그안에있는걸 빼내는 것

```
#include<iostream>
using std::cout;
using std::endl;
```

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

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

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

```

        B::test();
        return 0;
}

```

```

yong@yong-Z20NH-AS51B5U:~/test$ ./a.out
A에서 정의한 함수
B에서 정의한 함수

```

---

### reference

```

#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;
}

```

```

yong@yong-Z20NH-AS51B5U:~/test$ ./a.out
ref:11
val:11
ref:12
val:12

```

int 하고서 & 하고 ref 하고 나오는게있음  
 레퍼런스는 변수의 별명을 만듦(개념적으로 포인터)  
 포인터처럼 널값을 세팅할순없음  
 어렵게 생각할필요없이 그냥 포인터구나 생각하면 아무지장없음

---

### call by reference

```

#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<<' '; // 여기 ' ' 안에 한칸 띄워서 공백있어야함
        cout<<"val2:"<<val2<<endl;

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

        return 0;
    }

```

```

yong@yong-Z20NH-AS51BSU:~/test$ ./a.out
val1:10 val2:20
val1:20 val2:10

```

reference(레퍼런스)가지고 호출

10 하고 20 이교환됨  
포인터라 하더라도 무방

---

emolloc

c 에서  
stack \*tmp  
tmp=(stack \*)malloc(sizeof(stack))  
하던걸  
c++에서는 new 하면 끝남

주소값을 array 가 받음  
for 문돌면서 출력을 하는것  
마지막에 new 를 하면서 emolloc 을 했으니까 emolloc 의 짝은 free  
new 의 짝은 delete

```

#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 i=0; i<size; i++)
        cout<<"arr["<<j<<"]="<<arr[j]<<endl;

    delete []arr;
    return 0;
}

```

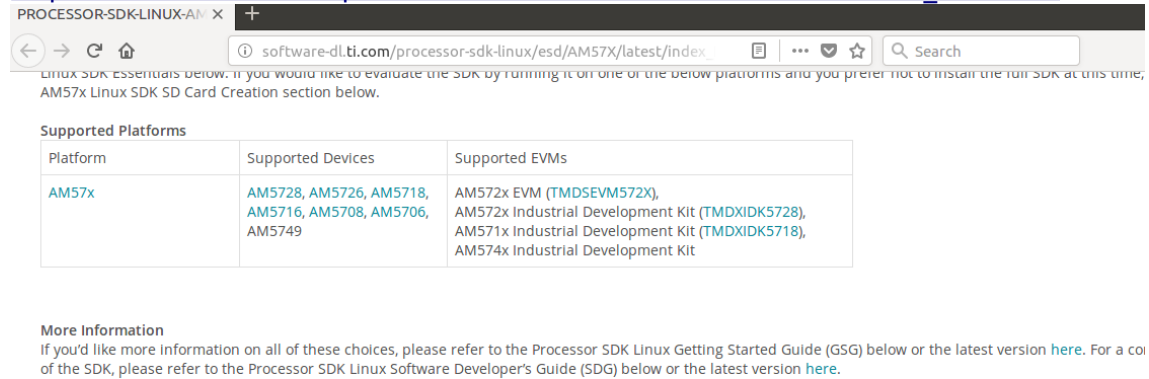
---

//

ti porocessor 다운

ti-processor-sdk-linux-am57xx-evm-04.03.00.05-Linux-x86-Install

[http://software-dl.ti.com/processor-sdk-linux/esd/AM57X/latest/index\\_FDS.html](http://software-dl.ti.com/processor-sdk-linux/esd/AM57X/latest/index_FDS.html)



PROCESSOR-SDK-LINUX-AM57X

Linux SDK essentials below. If you would like to evaluate the SDK by running it on one of the below platforms and you prefer not to install the full SDK at this time, AM57x Linux SDK SD Card Creation section below.

Platform	Supported Devices	Supported EVMs
AM57x	AM5728, AM5726, AM5718, AM5716, AM5708, AM5706, AM5749	AM572x EVM (TMDSEVM572X), AM572x Industrial Development Kit (TMDXIDK5728), AM571x Industrial Development Kit (TMDXIDK5718), AM574x Industrial Development Kit

**More Information**  
If you'd like more information on all of these choices, please refer to the Processor SDK Linux Getting Started Guide (GSG) below or the latest version [here](#). For a copy of the SDK, please refer to the Processor SDK Linux Software Developer's Guide (SDG) below or the latest version [here](#).

#### PROCESSOR-SDK-LINUX-AM57X Product downloads

Title	Description	Size
AM57xx Linux SDK Essentials		
<a href="#">ti-processor-sdk-linux-am57xx-evm-04.03.00.05-Linux-x86-Install.bin</a>	AM57xx EVM Linux SDK (64-bit Binary)	2360252K

설치전에 CCS 가 먼저 설치되어있어야함

CCS(씨씨에스) 리눅스 버전 설치 방법

<https://tuwlab.com/ece/27008>

#### 1. 의존성 패키지 설치

`sudo apt-get update`

```
sudo apt-get install libc6:i386 libx11-6:i386 libasound2:i386
libatk1.0-0:i386 libcairo2:i386 libcups2:i386 libdbus-glib-1-2:i386
libgconf-2-4:i386 libgdk-pixbuf2.0-0:i386 libgtk-3-0:i386 libice6:i386
libncurses5:i386 libsm6:i386 liborbit2:i386 libudev1:i386 libusb-0.1-
4:i386 libstdc++6:i386 libxt6:i386 libxtst6:i386 libgnomeui-0:i386
libusb-1.0-0-dev:i386 libcanberra-gtk-module:i386 gtk2-engines-
murrine:i386 unzip
```

Grace 를 사용하려는 경우 다음 패키지들도 추가로 설치해 줍니다.

```
sudo apt-get install libnss3:i386 libnss3-1d:i386 libnss3-
nssdb:i386
```

#### 2. CCS 설치

다음 페이지에서 Linux 용 CCS 설치파일을 다운로드합니다.

▶ <http://www.ti.com/tool/CCSTUDIO-TM4X?keyMatch=CCS&tisearch=Search-EN-Everything>

리눅스 버전 64 비트

offline 으로 설치

압축을 풀면 `ccs_setup_6.x.x.xxxxxx.bin` 파일이 나오는데, 이를 **User 권한으로 실행**합니다.

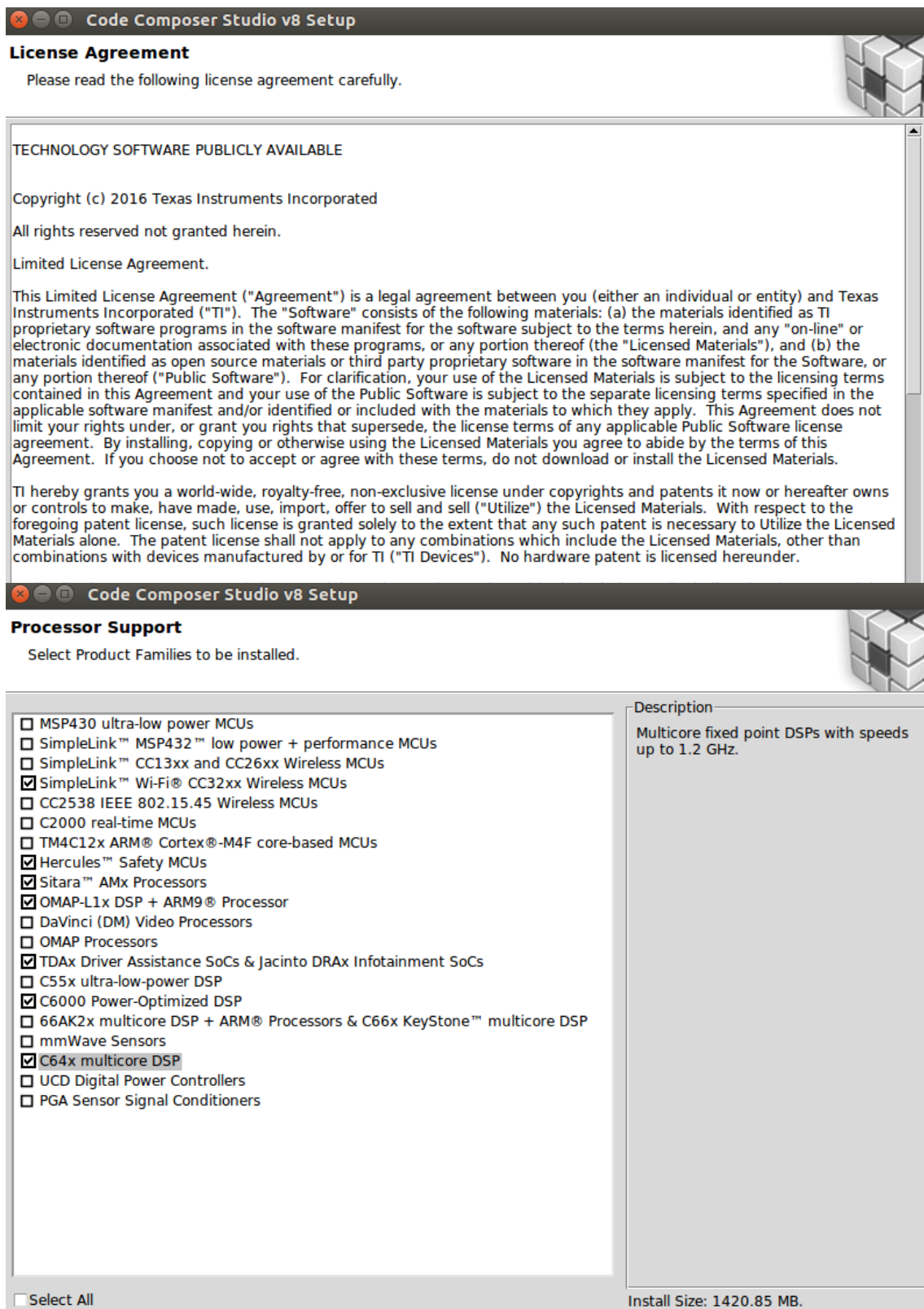
~/Downloads/CCS8.0.0.00016\_linux-x64\$ ./ccs\_setup\_linux64\_8.0.0.00016.bin

./ccs\_setup\_6.x.x.xxxxxx.bin

실행을 하면 조금 오랫동안 아무런 반응이 없는데, 원래 오래 걸리는 것이므로 기다리도록 합니다. (아무래도 Windows 의 InstallShield 를 에뮬레이션;;하느라 오래 걸리는것 같습니다.)

만약 아무 일도 일어나지 않고 그냥 종료되면 의존성 패키지가 올바르게 설치되지 않은 것입니다. 의존성 패키지 설치를 다시 진행하고 시도하세요. (오류메시지라도 좀 띄워주지;;)

이후 CCS 설치 과정은 Windows 에서와 같습니다. (글꼴이 좀 맘에 안드는것 같기는 하지만..) 설치 경로를 지정하고 설치할 구성요소들을 선택하고 설치를 진행합니다.





보드는 아무거나

CCS의 설치가 완료되면 위와 같이 Driver를 설치하라는 안내가 나옵니다. 이 과정을 그대로 수행하도록 합니다.

Driver 설치를 위한 스크립트는 {설치 경로}/ccsv6/install\_scripts/ 디렉토리에 있습니다. 설치 과정에서 설치 경로를 바꾸지 않고 그대로 설치했다면 /home/{UserName}/ti/ccsv6/install\_scripts/ 디렉토리에 있을 것입니다.

여기로 이동해서 install\_driver.sh를 Root 권한으로 실행합니다.

```
cd /home/{UserName}/ti/ccsv{vergiun number}/install_scripts
```

```
cd/home/yong/ti/ccsv8/install_scripts/
```

```
sudo ./install_drivers.sh
```

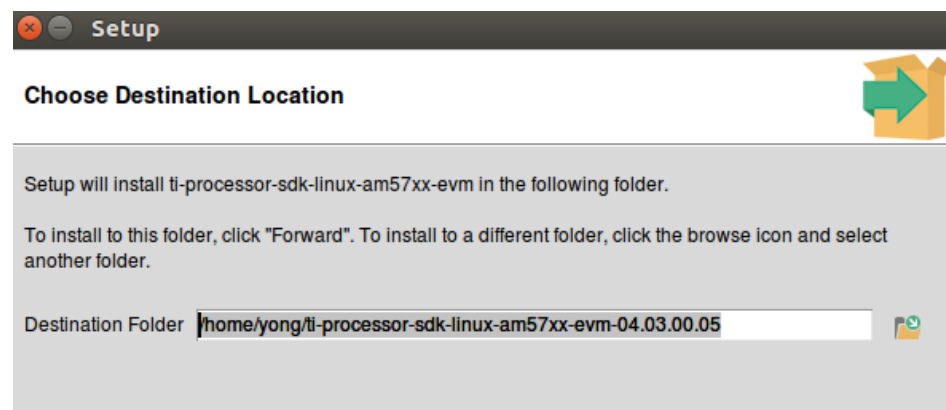
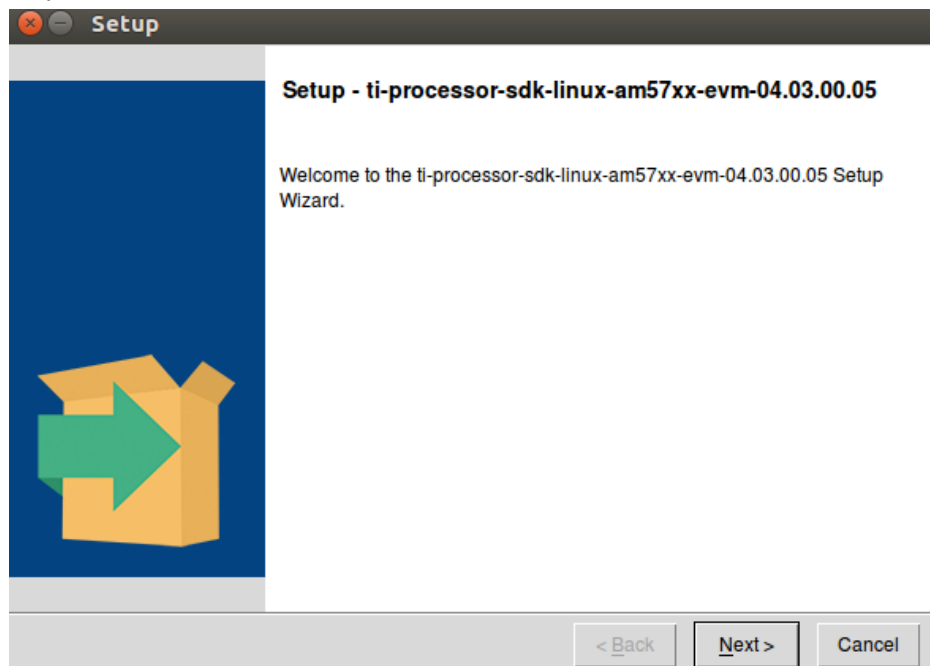
//

세팅-인스톨

```
ls -al ti-processor-sdk-linux-am57xx-evm-04.03.00.05-Linux-x86-Install.bin
```

```
chmod 755 ti-processor-sdk-linux-am57xx-evm-04.03.00.05-Linux-x86-Install.bin
```

```
./ti-processor-sdk-linux-am57xx-evm-04.03.00.05-Linux-x86-Install.bin
```



```
cd ti-processor-sdk-linux-am57xx-evm-04.03.00.05/bin
```

quiz  
설치했으면 4 번을 풀어보라

Vi student.h

```
#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
```

Vi studunt.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();

    return 0;
}

```

g++ Student.cpp main.cpp

./a.out

c++폴더안에 소스있음

땡땡은 네임스페이스에서 접근할때의 연산자임  
클래스라는 네임스페이스  
스튜던트 클래스라는 네임스페이스에 접근할때의 연산자

c++에는 생성자가있음

컨스트로덕트

클래스이름과 함수이름이 똑가탕  
여기선 A가 생성자

A a; //클래스 타입의 A에 a라는 변수를 선언하겠다

```
#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;
}

```

//디버그  
-gcc ?

////////////////

클래스

변수 private

함수 public

양식

실습문제  
자동차 클래스 만들기  
속도(속력, 방향)  
색상  
기름  
을 표현할수있도록 만들기  
생성자 활용

car 소스

**car\_main.cpp**  
#include "Car.h"

```

int main(void)
{
    Car tesla(30, "blue", 11.1);
    tesla.print_car_info();

    return 0;
}

```

**Car.h**  
#ifndef \_\_CAR\_H\_\_  
#define \_\_CAR\_H\_\_

```
#include <string.h>

class Car{
private:
    float speed;
    char color[10];
    float fuel;
public:
    Car(float s, char *c, float f)
    {
        speed = s;
        strncpy(color, c, strlen(c) + 1);
        fuel = f;
    }

    void input_data(void);
    void print_car_info(void);
};
Academy
#endif
```

### **Car.cpp**

```
#include "Car.h"
#include <iostream>

using namespace std;

void Car::input_data(void)
{
    cout << "Input your data(speed, color, fuel)" << endl;
    cin >> speed;
    cin >> color;
    cin >> fuel;
}

void Car::print_car_info(void)
{
    cout << "speed = " << speed;
    cout << " color = " << color;
    cout << " fuel = " << fuel << endl;
}
```

////////

생성자의 반대는 소멸자

디스트럭터

항상 물결이 붙는다

~/git/Homework/sanghoonlee/lec/c++

//ftd 코드로 시리얼코드만들기

```
#include<iostream>
```

```
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;
}

```

dsp //  
 전원 켤때는  
 뒤에다 꽃고 전원버튼 클릭  
 끌때는  
 뒤에 전원버튼 꾸욱 누르고 꺼질때까지 있다가 떼기