

TI DSP,MCU 및 Xilinx Zynq FPGA

프로그래밍 전문가 과정

이름	문지희
학생 이메일	mjh8127@naver.com
날짜	2018/6/8
수업일수	70 일차
담당강사	Innova Lee(이상훈)
강사 이메일	gcccompil3r@gmail.com

목차

petalinux 설치

C++

petalinux

[Petalinux 다운로드]

<https://www.xilinx.com/support/download.html>

Embedded Development - Archive - 2015.4 - PetaLinux 2015.4 Installer(TAR/GZIP - 1.68GB) 선택하여 다운

다운 받은 파일을 권한 설정을 해줌

```
chmod 755 petalinux-v2015.4-final-installer-dec.run
```

[petalinux 툴 생성]

```
sudo apt-get update
```

```
sudo apt-get install tofrodos iproute tftpd-hpa gawk gcc git-core make net-tools libncurses5-dev zlib1g-dev libssl-dev flex bison lib32z1  
lib32ncurses5 lib32stdc++6 libseline1
```

```
./petalinux-v2015.4-final-installer-dec.run ~/petalinux_zynq/
```

~/petalinux_zynq/petalinux-v2015.4-final/components/linux-kernel/xlnx-4.0/drivers/uio 위치에서 uio_pdrv_genirq.c 소스코드 수정

```
#ifdef CONFIG_OF  
static struct of_device_id uio_of_genirq_match[] = {
```

```
{ .compatible = "generic-uid", },  
  
{ /* This is filled with module_parm */ },  
  
{ /* Sentinel */ },  
  
};
```

다운로드하고 압축 풀기

https://www.xilinx.com/support/documentation/university/vivado/workshops/vivado-embedded-linux-zynq/materials/2015x/LiveUSB_2015.4.zip

```
xeno@xeno-NH:~/Downloads/LiveUSB_2015.4$ cd ~/petalinux_zynq/petalinux-v2015.4-final/  
xeno@xeno-NH:~/petalinux_zynq/petalinux-v2015.4-final$ cp ~/Downloads/LiveUSB_2015.4/ZYBO_petalinux_v2015_4.bsp ./  
xeno@xeno-NH:~/petalinux_zynq/petalinux-v2015.4-final$ chmod 755 ZYBO_petalinux_v2015_4.bsp
```

home 으로 가서

vi ~/.bashrc

맨 마지막 부분에

```
source /home/xeno/petalinux_zynq/petalinux-v2015.4-final/settings.sh
```

추가한다.

source ~/.bashrc

```
xeno@xeno-NH:~$ vi ~/.bashrc
xeno@xeno-NH:~$ source ~/.bashrc
PetaLinux environment set to '/home/xeno/petalinux_zynq/petalinux-v2015.4-final'
INFO: Checking free disk space
INFO: Checking installed tools
INFO: Checking installed development libraries
INFO: Checking network and other services
```

테스트

```
xeno@xeno-NH:~$ mkdir fpga_test
xeno@xeno-NH:~$ cd fpga_test/
xeno@xeno-NH:~/fpga_test$ petalinux-create -t project -n test --template zynq
INFO: Create project: test
INFO: New project successfully created in /home/xeno/fpga_test/test
xeno@xeno-NH:~/fpga_test$ ls
test
xeno@xeno-NH:~/fpga_test$
```

위의 사진과 같이 실행하면 됨.

테스트 한 뒤 test 디렉토리를 지워줌

```
rm -r test
```

fpga_test 디렉토리에 ZYBO_petalinux_v2015_4.bsp 을 가져옴

```
cp ~/Downloads/LiveUSB_2015.4/ZYBO_petalinux_v2015_4.bsp ./
```

```
petalinux-create -t project -s ZYBO_petalinux_v2015_4.bsp
```

```
cd ZYBO_petalinux_v2015_4/
```

```
petalinux-build
```

에러나면

```
sudo dpkg-reconfigure dash
```

해서 NO 선택

순서대로 아래 명령어 치기.

```
sudo dpkg --add-architecture i386
```

```
sudo apt-get update
```

```
sudo apt-get install libbz2-1.0:i386
```

```
sudo apt-get install tofrodos iproute tftpd-hpa gawk gcc git-core make net-tools libncurses5-dev zlib1g-dev libssl-dev flex bison lib32z1  
lib32ncurses5 libselinux1
```

```
sudo apt-get install xinetd tftpd-hpa
```

```
sudo apt-get install qemu-user-static qemu-system
```

```
sudo apt-get install linaro-image-tools
```

```
sudo apt-get install gcc-arm-linux-gnueabi
```

다시

```
petalinux-build
```

fail 2 개가 뜨는데 이거 뜨면 성공한 것이어서 아래 명령어 수행하면 됨.

```
[INFO ] Package HDF bitstream  
[INFO ] Failed to copy images to TFTPBOOT /tftpboot  
webtalk failed:PetaLinux statistics:extra lines detected:notsent_nofile!  
webtalk failed:Failed to get PetaLinux usage statistics!
```

```
petalinux-boot --qemu --kernel
```

→ 맨 마지막 명령어에서 로그인하라는 창이 뜨면 아이디, 비밀번호 둘 다 root 치면 됨.

ax 치면 나가짐.


[vivado]

home 디렉토리에 hardware 디렉토리 생성 후 안에 lab6 디렉토리 생성

새 프로젝트 생성

‘driver_lab6’ 이름으로 생성, home/hardware/lab6 위치에 생성

Project Name
Enter a name for your project and specify a directory where the project data files will be stored.



Project name:

Project location:

☐ Create project subdirectory

Project will be created at: /home/xeno/hardware

Project Type

Specify the type of project to create.



☒ RTL Project

You will be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis.

☒ Do not specify sources at this time

Select: ☒ Parts ☒ Boards

Filter/ Preview

Vendor:

Display Name:

Board Rev:

Reset All Filters

Search: (3 matches)

Display Name	Vendor	Board Rev	Part	I/O Pin C
<input checked="" type="checkbox"/> Zybo Z7-10	digilentinc.com	B.2	<input checked="" type="radio"/> xc7z010clg400-1	400
<input checked="" type="checkbox"/> Zybo Z7-20	digilentinc.com	B.2	<input checked="" type="radio"/> xc7z020clg400-1	400
<input checked="" type="checkbox"/> Zybo	digilentinc.com	B.3	<input checked="" type="radio"/> xc7z010clg400-1	400

software 디렉토리 생성

```
xeno@xeno-NH:~/zunq_zybo/lab6$ petalinux-create -t project -n software --template zynq
```

```
INFO: Create project: software
```

```
INFO: New project successfully created in /home/xeno/zunq_zybo/lab6/software
```

```
xeno@xeno-NH:~/zunq_zybo/lab6$ ls
```

```
hardware  software
```

```
xeno@xeno-NH:~/zunq_zybo/lab6$ cd hardware/driver_lab.sdk/
```

```
xeno@xeno-NH:~/zunq_zybo/lab6/hardware/driver_lab.sdk$ petalinux-config --get-hw-description -p ~/zunq_zybo/lab6/software
```

```
창이 뜨면 exit → yes
```

```
xeno@xeno-NH:~/zunq_zybo/lab6$ petalinux-create -t project -s ~/petalinux_zynq/petalinux-v2015.4-final/ZYBO_petalinux_v2015_4.bsp
```

```
xeno@xeno-NH:~/zunq_zybo/lab6/ZYBO_petalinux_v2015_4$ petalinux-create -t apps --name gpio-dev-mem-test
```

~/zunq_zybo/lab6/software 에서 config.project 권한 설정하여 초록색으로 바꿈.(chmod 755 config.project)

```
xeno@xeno-NH:~/zunq_zybo/lab6/software$ petalinux-build
```

```
xeno@xeno-NH:~/zunq_zybo/lab6/software/images/linux$ petalinux-package --boot --fsbl zynq_fsbl.elf --fpga
```

```
~/zunq_zybo/lab6/hardware/driver_lab.runs/impl_1/system_wrapper.bit --u-boot
```

```
ERROR: This tool requires 'bootgen' and it is missing. Please source Xilinx Tools settings first
```

에러가 뜨면

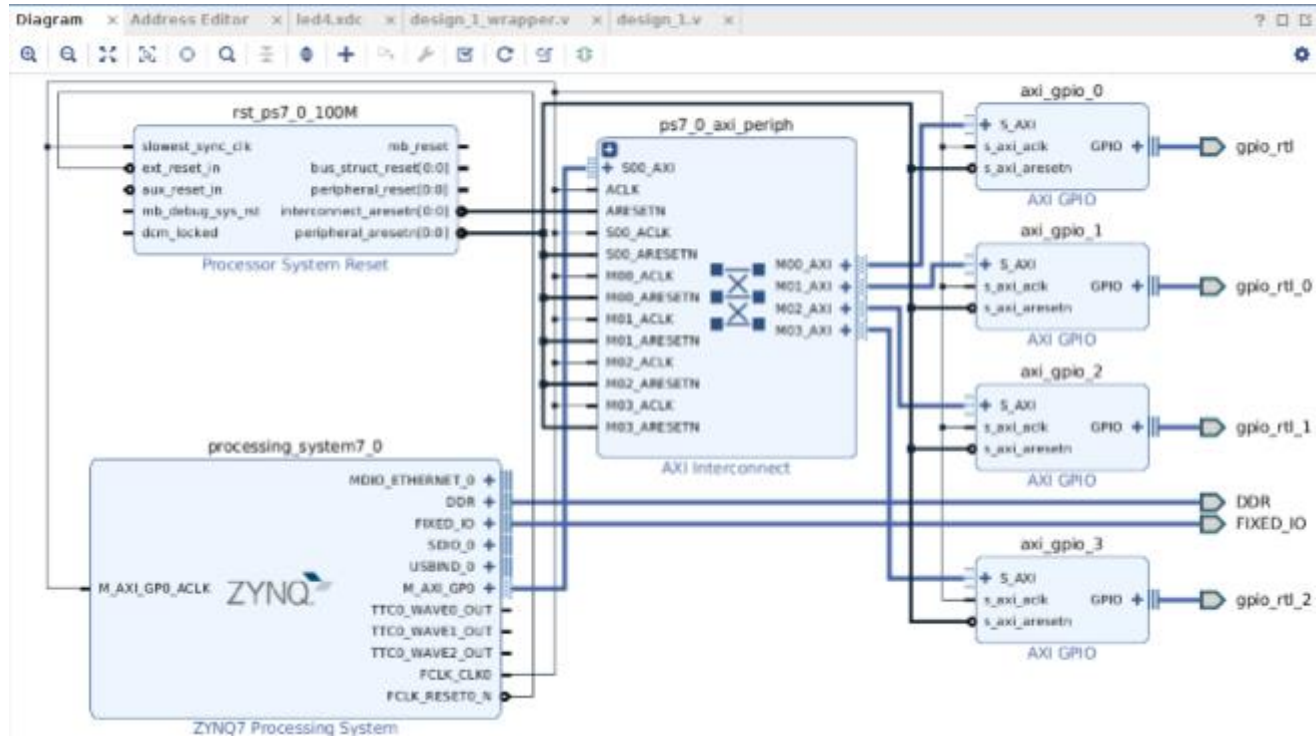
```
source /opt/Xilinx/Vivado/2017.1/settings64.sh
```

명령어 실행

다시 실행하려 했던 명령어실행

```
xeno@xeno-NH:~/zunq_zybo/lab6/software/images/linux$ petalinux-package --boot --fsbl zynq_fsbl.elf --fpga  
~/zunq_zybo/lab6/hardware/driver_lab.runs/impl_1/system_wrapper.bit -u-boot
```

Led 4 개 깜빡거리기



```
1 set_property -dict { PACKAGE_PIN V15 IOSTANDARD LVCMOS33 } [get_ports {gpio_rtl_tri_io[0]}];  
2 set_property -dict { PACKAGE_PIN W15 IOSTANDARD LVCMOS33 } [get_ports {gpio_rtl_0_tri_io[0]}];  
3 set_property -dict { PACKAGE_PIN T11 IOSTANDARD LVCMOS33 } [get_ports {gpio_rtl_1_tri_io[0]}];  
4 set_property -dict { PACKAGE_PIN T10 IOSTANDARD LVCMOS33 } [get_ports {gpio_rtl_2_tri_io[0]}];
```

[SDK]

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

```
#include <xgpio.h>
```

```
#include "xparameters.h"
```

```
#include "sleep.h"
```

```
int main(void)
```

```
{
```

```
XGpio out1, out2, out3, out4;
```

```
XGpio_Initialize(&out1, XPAR_AXI_GPIO_0_DEVICE_ID);
```

```
XGpio_Initialize(&out2, XPAR_AXI_GPIO_1_DEVICE_ID);
```

```
XGpio_Initialize(&out3, XPAR_AXI_GPIO_2_DEVICE_ID);
```

```
XGpio_Initialize(&out4, XPAR_AXI_GPIO_3_DEVICE_ID);
```

```
XGpio_SetDataDirection(&out1, 1, 0x0);
```

```
XGpio_SetDataDirection(&out2, 1, 0x0);
```

```
XGpio_SetDataDirection(&out3, 1, 0x0);
```

```
XGpio_SetDataDirection(&out4, 1, 0x0);
```

```
while(1)
```

```
{
```

```
Xil_Out32(0x41200000, 0xFFFFFFFF);  
  
sleep(1);  
  
Xil_Out32(0x41200000, 0x0);  
  
sleep(1);  
  
Xil_Out32(0x41210000, 0xFFFFFFFF);  
  
sleep(1);  
  
Xil_Out32(0x41210000, 0x0);  
  
sleep(1);  
  
Xil_Out32(0x41220000, 0xFFFFFFFF);  
  
sleep(1);  
  
Xil_Out32(0x41220000, 0x0);  
  
sleep(1);  
  
Xil_Out32(0x41230000, 0xFFFFFFFF);  
  
sleep(1);  
  
Xil_Out32(0x41230000, 0x0);  
  
sleep(1);  
  
}  
  
return 0;  
  
}
```


C++

컴파일 할 때 : g++

확장자 명 : cpp

- hello.cpp

```
#include <iostream> //stdil.h
using std::cout; //출력
using std::endl; //개행문자 == 'Wn'

int main(void)
{
    cout << "Hello World!!" << endl; // << == 연산자
    cout << "Hello" << "World!!" << endl;
    cout << 1 << 'a' << "String" << endl;
    return 0;
}
```

```
xeno@xeno-NH:~/proj/0607$ vi hello.cpp
xeno@xeno-NH:~/proj/0607$ g++ hello.cpp
xeno@xeno-NH:~/proj/0607$ ./a.out
Hello World!!
Hello World!!
```

1aString

- cin.cpp

```
#include <iostream>
using std::cout;
using std::endl;
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 << "덧셈 결과 : " << result << endl;
    return 0;
}
```

xeno@xeno-NH:~/proj/0607\$./a.out

1 번째 정수 입력:10

2 번째 정수 입력 :20

덧셈 결과 : 30

같은 함수명 사용 가능 - 인자수를 보고 파악한다

```
#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;
}
```

```
xeno@xeno-NH:~/proj/0607$ ./a.out
10
```

함수 Default 값 설정

```
#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;
}
```

```
xeno@xeno-NH:~/proj/0607$ ./a.out
```

```
1
```

```
8
```

- inline - 함수 크기가 작을 때만 사용

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

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

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

```
xeno@xeno-NH:~/proj/0607$ ./a.out
25
```

-namespace - 구조체랑 비슷한 개념

```
#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;
}

```

```
xeno@xeno-NH:~/proj/0607$ ./a.out
```

```
A 에서 정의한 함수
```

```
B 에서 정의한 함수
```

Reference - 변수의 별명을 만듦 (포인터라고 생각하지만 널값을 셋팅할 순 없음)

여기서 call by 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;
}
```

```
xeno@xeno-NH:~/proj/0607$ ./a.out
ref : 11
val : 11
ref : 12
val : 12
```

```
#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;

    return 0;
}
```

```
xeno@xeno-NH:~/proj/0607$ ./a.out
val1: 10 val2: 20
val1: 20 val2: 10
```


Malloc

```
#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;
}
```

할당하고자 하는 배열의 크기 : 5

arr[j] = 1

arr[j] = 2

arr[j] = 3

```
arr[j] = 4  
arr[j] = 5
```

Class - 구조체

Abstraction : 사용하는 변수, 제어하는 함수가 같은 class 에 묶여있는

private :

변수

public :

함수

와 같은 형태로 해야 good Abstraction ??:w

```
#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;
}

```

```
xeno@xeno-NH:~/proj/0607$ ./a.out
```

```
현재 문의 상태 : OPEN
```

```
현재 문의 상태 : CLOSE
```

```

#include <iostream>
using namespace std;

```

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

class Door
{
private:
    int state;
public:
    void Open();
    void Close();
    void ShowState();
};
```

```
#include "Door.h"
```

```
void Door::Open(void)
{
    state = OPEN;
}
```

```
void Door::Close(void)
{
    state=CLOSE;
}
```

```
void Door::ShowState(void){
    cout << "현재문의상태 : ";
```

```
        cout << ((state == OPEN ) ? "OPEN" : "CLOSE" ) << endl;
    }
```

```
#include "Door.h"
```

```
int main(void)
```

```
{
    Door door;

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

```
xeno@xeno-NH:~/proj/0607$ g++ Door.cpp Main.cpp
```

```
xeno@xeno-NH:~/proj/0607$ ./a.out
```

```
현재문의상태 : OPEN
```

```
현재문의상태 : CLOSE
```

Quiz 4

```
#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
```

```
#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;
}
```

```
#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;  
}
```

“::”는 namespace 에서 접근할 때 사용하는 연산자
생성자

```
#include <iostream>  
using namespace std;  
  
class A  
{  
    int i,j;  
public:  
    A(void)  
    {  
        cout << "생성자 호출 " << endl;  
        i=10, j=20;  
    }  
    void ShowDate(void)  
    {
```



```

        cout << i<< ' ' << j << endl;
    }
};

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

```

xeno@xeno-NH:~/proj/0607\$./a.out

생성자 호출

10 20

자동차 예제

```

#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);
};
```

```
#endif
```

```
#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;
}
```

```
#include "Car.h"
```

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

    return 0;
}
```

```
xeno@xeno-NH:~/Homework/sanghoonlee/lec/c++$ ./a.out
speed = 30 color = blue fuel = 11.1
```

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

class Academy
{
    char *name;
```

```

        char *phone;
public:
    Academy(char *_name, char *_phone);
    ~Academy(void);
    void ShowDate(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::ShowDate(void)
{
    cout << "name : " << name << endl;

```

```
        cout << "phone : " << phone << endl;
    }

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

```
xeno@xeno-NH:~/proj/0607$ ./a.out
name : Bit
phone : 02-111-2222
소멸자 호출
```

friend

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

class Counter
{
    private:
    int val;
    public:
```

```

    Counter(void)
    {
        val=0;
    }
    void Print(void)
    {
        cout << val << endl;
    }
    friend void SetVal(Counter& c, int val); //friend 가 붙으면 관련없는 객체도
접근할 수 있음
};

void SetVal(Counter& c, int val)
{
    c.val=val;
}

int main(void)
{
    Counter cnt;
    cnt.Print();
    SetVal(cnt,2002);
    cnt.Print();
    return 0;
}

```

xeno@xeno-NH:~/proj/0608\$./a.out

0
2002

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

class A
{
    private:
    int data;
    friend class B; //B 는 A 에 접근 가능. A 는 B 에 접근 불가능
};

class B
{
    public:
    void SetData(A& a, int data)
    {
        a.data = data;
    }
};

int main(void)
{
    A a;
```

```
    B b;  
    b.SetData(a,10);  
    return 0;  
}
```

```
#include <iostream>  
using namespace std;  
  
class A  
{  
public:  
    A(void)  
    {  
        cout << "A() call" << endl;  
    }  
    A(int i)  
    {  
        cout << "A(int i) Call" << endl;  
    }  
    A(const A& a)  
    {  
        cout << "A(const A& a) Call" << endl;  
    }  
};
```



```
int main(void)
{
    A obj1;
    A obj2(10);
    A obj3(obj2);
    return 0;
}
```

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

```
class A
{
    private:
    int data;
    friend class B;
};
```

```
class B
{
    public:
    void SetData(A& a, int data)
    {
        a.data = data;
    }
};
```

```
int main(void)
{
    A a;
    B b;
    b.SetData(a,10);
    return 0;
}
```

복사생성자

Copy constructor operate 동작할 때

1. 기존에 생성된 객체로 새로운 객체 초기화
2. 함수 호출 시 객체를 reference 가 아닌 형태로 전달할 경우
3. 함수 내에서 객체를 reference 가 아닌 형태로 return 하는 경우

1번케이스

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

class Point
{
    int x,y;
public:
    Point(int _x, int _y)
    {
```

```

        x = _x;
        y = _y;
    }
    void ShowData(void)
    {
        cout << x << ' ' << y << endl;
    }
};

int main(void)
{
    Point p1(10,20);
    Point p2(p1);

    p1.ShowData();
    p2.ShowData();
    return 0;
}

```

```
xeno@xeno-NH:~/proj/0608$ ./a.out
```

```
10 20
```

```
10 20
```

```

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

```

```
class Person
```

```
{
    char *name;
    char *phone;
public:
    Person(char *_name, char *_phone);
    ~Person();
    Person(const Person&);
    void ShowData();
};
```

```
Person::Person(char *_name ,char *_phone)
{
    name = new char[strlen(_name) +1];
    strcpy(name, _name);
    phone = new char[strlen(_phone) +1];
    strcpy(phone, _phone);
}
```

```
Person::~~Person(void)
{
    delete []name;
    delete []phone;
}
```

```
Person::Person(const Person& p)
{
```

```

        name = new char[strlen(p.name) + 1];
        strcpy(name, p.name);
        phone = new char[strlen(p.phone) + 1];
        strcpy(phone, p.phone);
    }

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

int main(void)
{
    Person p1("Jo", "011-9272-6523");
    Person p2 = p1;
    return 0;
}

```

2번케이스

```

#include <iostream>
using namespace std;
class A
{

```

```
        int val;
public:
    A(int i)
    {
        cout << "A(int i) Call" << endl;
        val = i;
    }
    A(const A& a)
    {
        cout << "A(const A& a)Call" << endl;
        val = a.val;
    }
    void ShowData(void)
    {
        cout<< "val : " << val << endl;
    }
};

void function(A a)
{
    a.ShowData();
}

int main(void)
{
    A obj(30);
```

```
function(obj);  
return 0;  
}
```

```
xeno@xeno-NH:~/proj/0608$ ./a.out  
A(int i) Call  
A(const A& a)Call  
val : 30
```

3번 케이스

```
#include <iostream>  
  
using namespace std;  
  
class A  
{  
    int val;  
  
public:  
    A(int i)  
    {  
        cout << "A(int i) Call" << endl;  
        val = i;  
    }  
}
```

```

    }
    A(const A& a)
    {
        cout << "A(const A& a) Call" << endl;
        val = a.val;
    }
    void ShowData(void)
    {
        cout << "val : " << val << endl;
    }
};

A function(A& a)
{
    return a;
}

int main(void)
{
    A a(10);
    function(a).ShowData();
    return 0;
}

```

```

A(int i) Call
A(const A& a) Call
val : 10

```


상속을 사용하는 이유

1. 우선 상속에 class 재활용성
2. 요구사항 변화에 따른 유연성

```
#include <iostream>
#include <string.h>

using namespace std;

class Person
{
    int age;
    char name[20];
public:
    int GetAge(void) const
    {
        return age;
    }
    const char *GetName(void) const
    {
        return name;
    }
    Person(int _age = 1, char *_name = "noname")
```

```
        {
            age = _age;
            strcpy(name, _name);
        }
};

class Student:public Person
{
    char major[20];
public:
    Student(char *_major)
    {
        strcpy(major,_major);
    }
    const char *GetMajor(void) const
    {
        return major;
    }
    void ShowData(void) const
    {
        cout << "name :" << GetName() << endl;
        cout << "age : " << GetAge() << endl;
        cout << "major : " << GetMajor() << endl;
    }
};
```

```
int main(void)
{
    Student Park("Computer Science");
    Park.ShowData();
    return 0;
}
```

```
xeno@xeno-NH:~/proj/0608$ ./a.out
name :noname
age : 1
major : Computer Science
```

Overloading

:새로운 규칙을 만드는 것

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

class Point
{
private:
    int x,y;
public:
    Point(int _x = 0, int _y = 0) : x(_y), y(_x) {}
    void ShowPosition(void);
    void operator + (int val);
};

void Point::ShowPosition(void)
```

```
{  
    cout << x << " " << y << endl;  
}
```

```
void Point::operator + (int val)  
{  
    x += val;  
    y += val;  
}
```

```
int main(void)  
{  
    Point p(3,4);  
    p.ShowPosition();  
  
    p.operator + (10);  
    p.ShowPosition();  
    return 0;  
}
```

```
xeno@xeno-NH:~/proj/0608$ ./a.out  
4 3  
14 13
```

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

```
class Point
{
private:
    int x,y;
public:
    Point(int _x = 0, int _y = 0) : x(_x), y(_y) {}
    void ShowPosition(void);
    Point operator + (const Point& p);
};

void Point::ShowPosition(void)
{
    cout << x << " " << y << endl;
}

Point Point::operator + (const Point& p)
{
    Point temp(x+p.x, y+p.y);
    return temp;
}

int main(void)
{
    Point p1(1,2);
    Point p2(3,7);
```

```
    Point p3 = p1 + p2;  
    p3.ShowPosition();  
    return 0;  
}
```

```
xeno@xeno-NH:~/proj/0608$ ./a.out  
4 9
```

```
#include <iostream>  
using namespace std;  
  
class Point  
{  
private:  
    int x,y;  
public:  
    Point(int _x = 0, int _y = 0) : x(_x), y(_y) {}  
    void ShowPosition(void);  
    Point& operator ++(void);  
    friend Point& operator--(Point& p);  
};  
  
void Point::ShowPosition(void)  
{  
    cout << x << " " << y << endl;  
}
```

```
Point& Point::operator++(void)
```

```
{  
    x++;  
    y++;  
    return *this;  
}
```

```
Point& operator--(Point& p)
```

```
{  
    p.x--;  
    p.y--;  
    return p;  
}
```

```
int main(void)
```

```
{  
    Point p(3,7);  
    ++p;  
    p.ShowPosition();  
  
    --p;  
    p.ShowPosition();  
  
    ++(++p);  
    p.ShowPosition();  
}
```

```
        --(--p);  
        p.ShowPosition();  
  
        return 0;  
}
```

```
xeno@xeno-NH:~/proj/0608$ ./a.out  
4 8  
3 7  
5 9  
3 7
```

```
#include <iostream>  
using namespace std;  
  
class Point  
{  
private:  
    int x,y;  
public:  
    Point(int _x = 0, int _y = 0) : x(_x), y(_y) {}  
    void ShowPosition(void);  
    Point& operator++(void);  
    Point operator++(int);  
};
```



```
void Point::ShowPosition(void)
{
    cout << x << " " << y << endl;
}
```

```
Point& Point::operator++(void)
{
    x++;
    y++;
    return *this;
}
```

```
Point Point::operator++(int)
{
    Point temp(x,y);
    // ++(*this);
    x++;
    y++;
    return temp;
}
```

```
int main(void)
{
    Point p1(3,7);
    (p1++).ShowPosition();
}
```

```
p1.ShowPosition();
```

```
Point p2(33,77);
```

```
(++p2).ShowPosition();
```

```
p2.ShowPosition();
```

```
return 0;
```

```
}
```

```
xeno@xeno-NH:~/proj/0608$ ./a.out
```

```
3 7
```

```
4 8
```

```
34 78
```

```
34 78
```

```
#include <iostream>
```

```
using namespace std;
```

```
class Point
```

```
{
```

```
private:
```

```
    int x,y;
```

```
public:
```

```
    Point(int _x = 0, int _y = 0) : x(_x), y(_y) {}
```

```
    void ShowPosition(void);
```

```
    Point operator+(int val);
```

```
};

void Point::ShowPosition(void)
{
    cout << x << " " << y << endl;
}

Point Point:: operator+(int val)
{
    Point temp(x + val, y + val);
    return temp;
}

int main(void)
{
    Point p1(3,7);
    Point p2 = p1 + 3;
    p2.ShowPosition();
    return 0;
}
```

```
xeno@xeno-NH:~/proj/0608$ ./a.out
6 10
```

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

```
class Point
{
private:
    int x,y;
public:
    Point(int _x = 0, int _y = 0) : x(_x), y(_y) {}
    void ShowPosition(void);
    Point operator+(int val);
    friend Point operator + (int val, Point& p);
};

void Point::ShowPosition(void)
{
    cout << x << " " << y << endl;
};

Point Point:: operator + (int val)
{
    Point temp(x + val, y + val);
    return temp;
}

Point operator + (int val , Point& p)
{
    return p + val;
}
```

```
}  
  
int main(void)  
{  
    Point p1(3,7);  
    Point p2 = p1 +3;  
    p2.ShowPosition();  
  
    Point p3 = 7+p2;  
    p3.ShowPosition();  
    return 0;  
}
```

```
xeno@xeno-NH:~/proj/0608$ ./a.out  
6 10  
13 17
```

Template

자료형 타입을 자동으로 지정

```
#include <iostream>  
using namespace std;  
  
template <typename T>  
T Add(T a, T b)  
{  
    return a+b;  
}
```

```
}  
  
int main(void)  
{  
    cout << Add(10,20) << endl;  
    cout << Add(1.1, 2.2) << endl;  
    return 0;  
}
```

```
xeno@xeno-NH:~/proj/0608$ ./a.out
```

```
30
```

```
3.3
```