

TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

강사 : Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 : 황수정

sue100012@naver.com

69일차 (2018. 06. 07)

c++은 어디에서 사용될까

c++은 게임에서 많이 사용된다. c 언어는 사용하고 싶은데 개발속도를 내고 싶을 때 사용한다. 영상처리도 알고리즘을 빨리 만들어야 한다.

커널은 c++로 만들 수 없다.

지금 개발중인 go 라는 언어가 있는데 이것은 c, 자바 등의 장점을 취합한 것이다.

Inline 중요 함수 사이즈가 작을 때만 사용한다

함수가 크면 inline 써봤자 소용이 없다 오히려 성능이 떨어질 수 있다.

이름이 같은 함수여도 namespace 를 만들면 구동이 가능하다. 느낌은 구조체에 함수가 들어있는 것처럼 생각하면 된다

new 알아야 한다.,

abstraction 함수다 구조체 안에다가 변수를 집어넣는데 함수를 넣는 경우가 있다.

c 언어에서는 함수포인터를 만들 필요없이 함수 프로토 타입을 적으면 된다.

abstraction 의 진정한 정의는 무엇인가?

구조체 안에 변수와 그 변수를 제어하기 위한 것들이 함께 있으면 구조체 보면 무슨 함수인지 알 수 있다.

사용하는 변수들 그 변수들을 제어하는 함수가 같은 구조체에 묶여 있어야 한다.

일반적인 사용자가 쉽게 가져갈 수 없도록 만드는 것도 있다.

관계자만 볼수 있게 프라이빗으로 구성하는 것이 있다, 공용으로 사용하는 것과 섞어서 구현하면 된다.??

```
include <iostream>    // stdio.h 와 같은 것이다.
```

```
using std::cout;
```

```
using std::endl; // 개행 문자이다. \n 와 같은 것이다.
```

```
int main(void)
```

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

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

```
#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;
}
```

```
#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;
}
```

```
#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;
}
```

```

#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;
}

```

```

#include <iostream>
using namespace std;

int main (void)
{
    int val = 10;
    int &ref = val;
    /* 변수에 별명을 만드는 것
     * 포인터라고 생각해도 무방하다.
     * 그러나 포인터처럼 null 값을 만들 수 없다.
     */
    val++;
    cout << "ref:" << ref << endl;
    cout << "val:" << val << endl;

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

    return 0;
}

```

```

#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;
}

```

```

#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;
}

```

```

#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;
}

```

```

#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;
}

```

```

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

```

```

class Academy
{
    char *name;
    char *phone;
public:
    Academy(char *_name, char *_phone);
    ~Academy(void);    //~는 소멸자에서 반드시 사용되는 것이다.
    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;
}
```