

TI DSP, MCU, Xilinx Zynq FPGA 기반의 프로그래밍 전문가 과정

<C++>

2018.06.07 - 69 일차

강사 - Innova Lee(이상훈)
gcccompil3r@gmail.com

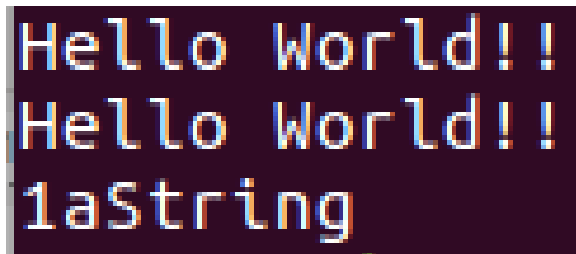
학생 - 안상재
sangjae2015@naver.com

1. hello.cpp

- Hello World 출력

```
1 #include <iostream>    // c 언어의 stdio.h 와 비슷한 개
2 using std::cout;       // 출력
3 using std::endl;       // endl : 개행 문자념
4
5 int main(void)
6 {
7     cout << "Hello World!!" << endl;    // << 는 집어넣으라는 연산자
8     cout << "Hello" << " World!!" << endl;
9     cout << 1 << 'a' << "String" << endl;
10    return 0;
11 }
```

1-1. 결과 출력



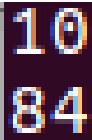
```
Hello World!!
Hello World!!
1aString
```

2. function.cpp

- c++언어에서는 함수명이 같아도 인자가 다르면 다른 함수로 취급한다.

```
1 #include <iostream>
2 using std::cout;
3 using std::endl;
4
5 int function(void)
6 {
7     return 10;
8 }
9
10 int function(int a, int b)
11 {
12     return a + b;
13 }
14
15 int main(void)
16 {
17     cout << function() << endl;
18     cout << function(7, 77) << endl;
19     return 0;
20 }
```

2-1. 결과 출력

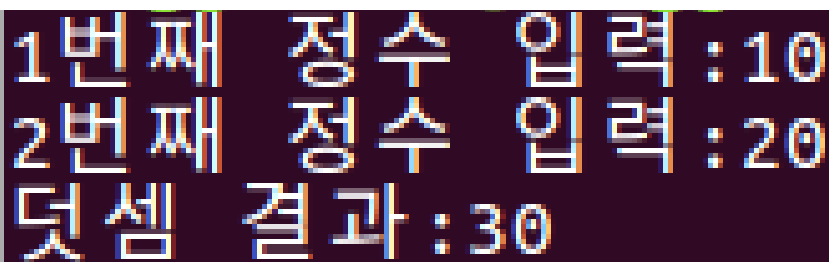


```
10
84
```

3. cin.cpp

```
1 #include <iostream>
2 using std::cout;
3 using std::endl;
4 using std::cin;
5
6 int main(void)
7 {
8     int val1, val2;
9     cout << "1 번째 정수 입력:"; // 출력
10    cin >> val1; // 입력
11
12    cout << "2 번째 정수 입력:";
13    cin >> val2;
14
15    int result = val1 + val2;
16    cout << "덧셈 결과:" << result << endl;
17
18    return 0;
19 }
```

3-1. 결과 출력



```
1 번째 정수 입력: 10
2 번째 정수 입력: 20
덧셈 결과: 30
```

4. inline.cpp

```
1 #include <iostream>
2 using std::cout;
3 using std::endl;
4
5 inline int SQUARE(int x)
6 {
7     return x*x;
8 }
9
10 int main(void)
11 {
12     cout << SQUARE(5) << endl;
13     return 0;
14 }
```

4-1. 결과출력

→ 25

5. call_by_ref.cpp

- c 언어에는 call by reference 라는 개념이 없지만, c++ 에는 call by reference 라는 개념이 있다.

```
1 #include <iostream>
2 using namespace std;
3
4 void swap(int& a, int& b)
5 {
6     int temp = a;
7     a = b;
8     b = temp;
9 }
10
11 int main(void)
12 {
13     int val1 = 10;
14     int val2 = 20;
15
16     cout << "val1:" << val1 << ' ';
17     cout << "val2:" << val2 << endl;
18
19     swap(val1, val2);
20     cout << " val1:" << val1 << ' ';
21     cout << "val2:" << val2 << endl;
22 }
```

```
23 return 0;
24 }
```

5-1. 결과 출력



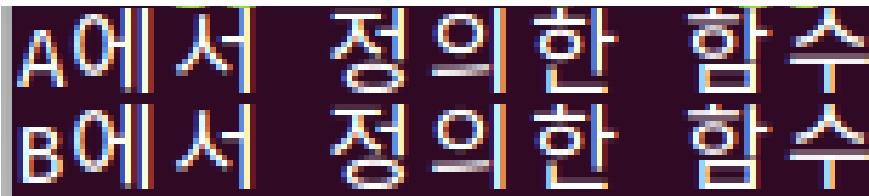
```
val1:10 val2:20
val1:20 val2:10
```

6. namespace.cpp

- namespace 를 구조체처럼 사용할 수 있다.

```
1 #include <iostream>
2 using std::cout;
3 using std::endl;
4
5 namespace A // 이름 공간(구조체에 함수가 들어있음)
6 {
7     void test(void)
8     {
9         cout << "A 에서 정의한 함수" << endl;
10    }
11 }
12
13 namespace B
14 {
15     void test(void)
16     {
17         cout << "B 에서 정의한 함수" << endl;
18     }
19 }
20
21 int main(void)
22 {
23     A::test();
24     B::test();
25
26     return 0;
27 }
```

6-1. 결과 출력

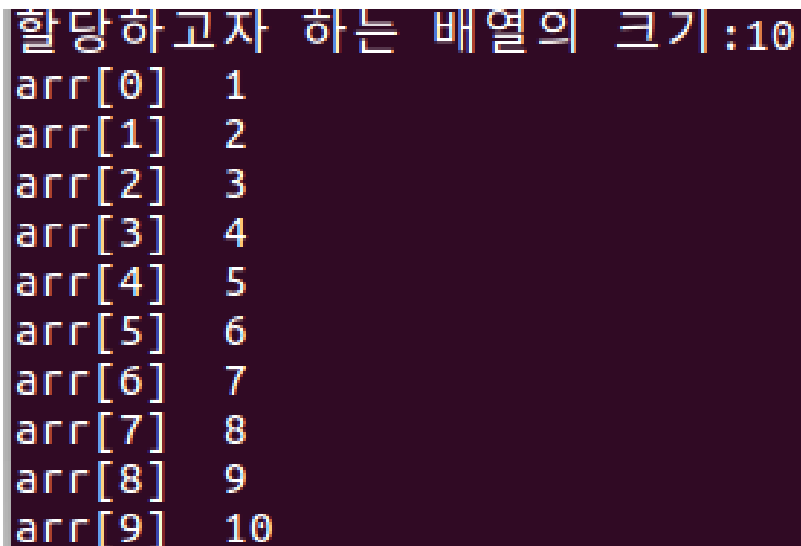


```
A에서 정의한 함수
B에서 정의한 함수
```

7. new.cpp

```
1 #include <iostream>
2 using namespace std;
3
4 int main(void)
5 {
6     int size;
7     cout << "할당하고자 하는 배열의 크기:";
8     cin >> size;
9
10    int *arr = new int[size];
11
12    for(int i=0;i<size;i++)
13        arr[i] = i+1;
14
15    for(int j=0;j<size;j++)
16        cout << "arr[" << j << "] \ " << arr[j] << endl;
17
18    delete []arr; // new 의 짝
19
20    return 0;
21 }
```

7-1. 결과 출력



```
할당하고자 하는 배열의 크기:10
arr[0]    1
arr[1]    2
arr[2]    3
arr[3]    4
arr[4]    5
arr[5]    6
arr[6]    7
arr[7]    8
arr[8]    9
arr[9]   10
```

8. constructor.cpp

```
1 #include <iostream>
2 using namespace std;
3
4 class A
5 {
6     int i,j;
7 public:
8     A(void) // 생성자
9     {
10         cout << "생성자 호출" << endl;
11         i=10, j=20;
12     }
13     void ShowData(void)
14     {
15         cout << i << ' ' << j << endl;
16     }
17 };
18
19 int main(void)
20 {
21     A a;
22     a.ShowData();
23
24     return 0;
25 }
```

8-1. 결과 출력



생성자 호출
10 20

9. destructor.cpp

```
1 #include <iostream>
2 #include <stdlib.h>
3 #include <string.h>
4 using namespace std;
5
6 class Academy
7 {
8     char *name;
9     char *phone;
10 public:
11     Academy(char *_name, char *_phone);
12     ~Academy(void);
13     void ShowData(void);
14 };
15
16 Academy::Academy(char *_name, char *_phone)
17 {
18     name = new char[strlen(_name) + 1];
19     strcpy(name, _name);
20
21     phone = new char[strlen(_phone) + 1];
22     strcpy(phone, _phone);
23 }
24
25 Academy::~Academy(void)
26 {
27     cout << "소멸자 호출" << endl;
28     delete []name;
29     delete []phone;
30 }
31
32 void Academy::ShowData(void)
33 {
34     cout << "name:" << name << endl;
35     cout << "phone:" << phone << endl;
```



```

36 }
37
38 int main(void)
39 {
40     Academy aca("Bit", "02-111-2222");
41     aca.ShowData();
42     return 0;
43 }

```

9-1. 결과 출력

```

name:Bit
phone:02-111-2222
소명자 호출

```

10. default_para.cpp

```

1 #include <iostream>
2 using std::cout;
3 using std::endl;
4
5 int func(int a=0)
6 {
7     return a+1;
8 }
9
10 int main(void)
11 {
12     cout << func() << endl;
13     cout << func(7) << endl;
14     return 0;
15 }

```

10-1. 결과 출력

```

1
8

```

11.

```
1 #include <iostream>
2 using namespace std;
3
4 const int OPEN = 1;
5 const int CLOSE = 2;
6
7 class Door
8 {
9     private:
10         int state;
11     public:
12         void Open();
13         void Close();
14         void ShowState();
15 };
16 void Door::Open(void)
17 {
18     state = OPEN;
19 }
20
21 void Door::Close(void)
22 {
23     state = CLOSE;
24 }
25
26 void Door::ShowState(void)
27 {
28     cout << "현재 문의 상태:";
29     cout << ((state == OPEN) ? "OPEN":"CLOSE") << endl;
30 }
31
32 int main(void)
33 {
34     Door door;
35
36     door.Open();
37     door.ShowState();
38     door.Close();
39     door.ShowState();
40
41     return 0;
42 }
```

11-1. 결과 출력

```
현재 문의 상태 : OPEN
현재 문의 상태 : CLOSE
```

12. ref.cpp

```
1 #include <iostream>
2 using namespace std;
3
4 int main(void)
5 {
6     int val = 10;
7     int &ref = val; // call by reference : 변수의 별명을 만듦 (포인터랑 같음)
8
9     val++;
10    cout << "ref:" << ref << endl;
11    cout << "val:" << val << endl;
12
13    ref++;
14    cout << "ref:" << ref << endl;
15    cout << "val:" << val << endl;
16
17    return 0;
18 }
```

12-1. 결과 출력

```
ref:11
val:11
ref:12
val:12
```

13. 성적 합, 평균 구하기

/* 헤더 파일 */

```
#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.cpp 파일 */

```
1 #include <iostream>
2 #include "student.h"
3
4 using namespace std;
5
6 void Student::input_score(void)
7 {
8     cout << "Input math, physics, computer score" << endl;
9     cin >> math;
10    cin >> physics;
11    cin >> computer;
12 }
13
14 void Student::calc_total(void)
15 {
16     total = math + physics + computer;
17 }
18
19 void Student::average(void)
20 {
```

```

21  ave = (float)(total / 3.0);
22 }
23
24 void Student::print_average(void)
25 {
26     cout << "Average = " << ave << endl;
27 }

```

/* main.cpp 파일 */

```

1  #include "student.h"
2
3  int main(void)
4  {
5      Student ds;
6      ds.input_score();
7      ds.calc_total();
8      ds.average();
9      ds.print_average();
10
11     Student sj;
12     sj.input_score();
13     sj.calc_total();
14     sj.average();
15     sj.print_average();
16
17     return 0;
18 }

```

13-1. 결과 출력

```

Input math, physics, computer score
50
50
50
Average = 50
Input math, physics, computer score
40
50
80
Average = 56.6667

```