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

<C++>

2018.06.08 – 70 일차

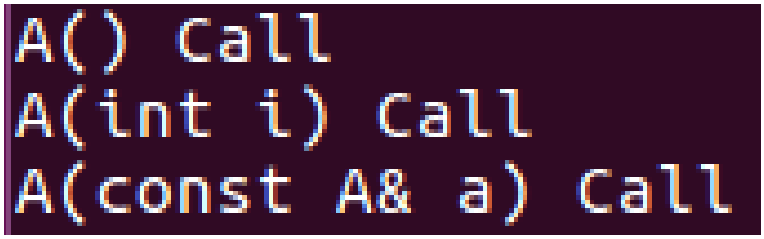
강사 - Innova Lee(이상훈)
gcccompil3r@gmail.com

학생 - 안상재
sangjae2015@naver.com

1. 복사생성자 1.cpp
- 함수명이 같아도 인자가 다르면, 다른 함수로 취급함.

```
 1 #include <iostream>
 2 using namespace std;
 3
 4 class A
 5 {
 6 public:
 7     A(void)
 8     {
 9         cout << "A() Call" << endl;
10     }
11     A(int i)
12     {
13         cout << "A(int i) Call" << endl;
14     }
15     A(const A& a)   // 객체를 생성자로 받을 때는 이런 타입으로 받음
16     {
17         cout << "A(const A& a) Call" << endl;
18     }
19 };
20
21 int main(void)
22 {
23     A obj1;
24     A obj2(10);
25     A obj3(obj2);  // 객체 생성
26
27     return 0;
28 }
```

1-1. 결과 분석



2. 복사생성자 2.cpp
- 생성자를 복사할 수 있음.

```
 1 #include <iostream>
 2 using namespace std;
 3
 4 class Point
 5 {
 6     int x, y;
 7
 8 public:
 9     Point(int _x, int _y)
10     {
11         x = _x;
12         y = _y;
13     }
14     void ShowData(void)
15     {
16         cout << x << ' ' << y << endl;
17     }
18 };
19
20 int main(void)
21 {
22     Point p1(10, 20);
23     Point p2(p1);   // 생성자가 복사가 됨
24
25     p1.ShowData();
26     p2.ShowData();
27
28     return 0;
29 }
```

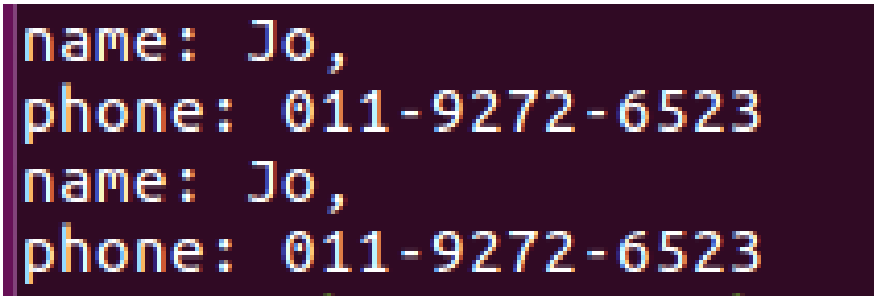
2-1. 결과 분석
10 20
10 20

3. 복사생성자 3.cpp

```
 1 #include <iostream>
 2 #include <stdlib.h>
 3 #include <string.h>
 4
 5 using namespace std;
 6
 7 class Person
 8 {
 9     char *name;
10     char *phone;
11
12 public:
13     Person(char *_name, char *_phone);
14     Person(const Person& p);
15     ~Person();
16     void ShowData();
17 };
18
19 Person::Person(char *_name, char *_phone)
20 {
21     name = new char[strlen(_name) + 1];
22     strcpy(name, _name);
23
24     phone = new char[strlen(_phone) + 1];
25     strcpy(phone, _phone);
26 }
27
28 Person::~Person(void)
29 {
30     delete []name;
31     delete []phone;
32 }
33
34 Person::Person(const Person& P)
35 {
36     name = new char[strlen(P.name) + 1];
37     strcpy(name, P.name);
38     phone = new char[strlen(P.phone) + 1];
39     strcpy(phone, P.phone);
40 }
41
42 void Person::ShowData(void)
```

```
43 {
44     cout << "name: " << name << endl;
45     cout << "phone: " << phone << endl;
46 }
47
48 int main(void)
49 {
50     Person p1("Jo,", "011-9272-6523");
51     Person p2 = p1;
52
53     p1.ShowData();
54     p2.ShowData();
55     return 0;
56 }
```

3-1. 결과 분석



4. 복사생성자 4.cpp

```
1 #include <iostream>
2 using namespace std;
3
4 class A
5 {
6     int val;
7 public:
8     A(int i)
9     {
10        cout << "A(int i) Call" << endl;
11        val = i;
12    }
13    A(const A& a)
14    {
15        cout << "A(const A& a) Call" << endl;
16        val = a.val;
17    }
18
19    void ShowData(void)
20    {
21        cout << "val: " << val << endl;
22    }
23 };
24
25 void function(A a)
26 {
27    a.ShowData();
28 }
29
30 int main(void)
31 {
32    A obj(30);
33    function(obj);
34
35    return 0;
36 }
```

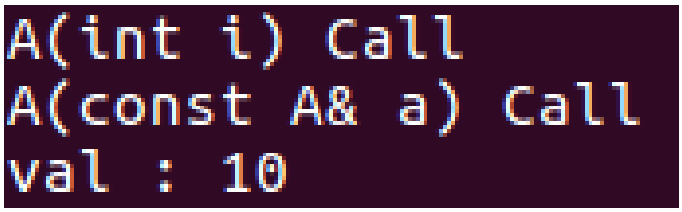
4-1. 결과 분석

```
A(int i) Call
A(const A& a) Call
val: 30
```

5. 복사생성자 5.cpp

```
 1 #include <iostream>
 2
 3 using namespace std;
 4
 5 class A
 6 {
 7     int val;
 8
 9 public:
10     A(int i)
11     {
12         cout << "A(int i) Call" << endl;
13         val = i;
14     }
15     A(const A& a)
16     {
17         cout << "A(const A& a) Call" << endl;
18         val = a.val;
19     }
20     void ShowData(void)
21     {
22         cout << "val : " << val << endl;
23     }
24 };
25
26 A function(A& a)
27 {
28     return a;
29 }
30
31 int main(void)
32 {
33     A a(10);
34     function(a).ShowData();
35     return 0;
36 }
```

5-1. 결과 분석



6. 연산자 1.cpp

```
#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);
     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;
}
```

6-1. 결과 분석
3 4
13 14

7. 연산자 2.cpp

```
 1 #include <iostream>
 2 using namespace std;
 3
 4 class point
 5 {
 6 private:
 7     int x,y;
 8 public:
 9     point(int _x=0, int _y=0):x(_x), y(_y){}
10     void showposition(void);
11     point operator+(const point& p);
12 };
13
14 void point::showposition(void)
15 {
16     cout << x << " " << y << endl;
17 }
18
19 point point::operator+(const point& p)
20 {
21     point temp(x + p.x, y + p.y);
22     return temp;
23 }
24
25 int main(void)
26 {
27     point p1(1,2);
28     point p2(3,7);
29     point p3 = p1 + p2;
30     p3.showposition();
31
32     return 0;
33 }
```

7-1. 결과 분석
4 9

8. 연산자 3.cpp

```
 1 #include <iostream>
 2 using namespace std;
 3
 4 class point
 5 {
 6 private:
 7     int x,y;
 8
 9 public:
10     point(int _x = 0, int _y = 0) : x(_x), y(_y) {}
11     void showposition(void);
12     point& operator++(void);
13     friend point& operator--(point& p);
14 };
15
16 void point::showposition(void)
17 {
18     cout << x << " " << y << endl;
19 }
20
21 point& point::operator++(void)
22 {
23     x++;
24     y++;
25     return *this;   // 현재 객체를 반환함
26 }
27
28 point& operator--(point& p)
29 {
30     p.x--;
31     p.y--;
32     return p;
33 }
34
35 int main(void)
36 {
37     point p(3,7);
38     ++p;
39     p.showposition();
40
41     --p;
42     p.showposition();
```

```
43
44     ++(++p);
45     p.showposition();
46
47     --(--p);
48     p.showposition();
49
50     return 0;
51 }
```

8-1. 결과 분석
4 8
3 7
5 9
3 7

9. 연산자 4.cpp

```
 1 #include <iostream>
 2 using namespace std;
 3
 4 class point
 5 {
 6 private:
 7     int x,y;
 8 public:
 9     point(int _x = 0, int _y = 0) : x(_x), y(_y) {}
10     void showposition(void);
11     point& operator++(void);
12     point operator++(int);
13 };
14
15 void point::showposition(void)
16 {
17     cout << x << " " << y << endl;
18 }
19
20 point& point::operator++(void)
21 {
22     x++;
23     y++;
24     return *this;
25 }
26
27 point point::operator++(int)
```

```
28 {
29     point temp(x,y);
30     // ++(*this);
31     x++;
32     y++;
33     return temp;
34 }
35
36 int main(void)
37 {
38     point p1(3,7);
39     (p1++).showposition();
40     p1.showposition();
41
42     point p2(33,77);
43     (++p2).showposition();
44     p2.showposition();
45
46     return 0;
47 }
```

9-1. 결과 분석

```
3 7
4 8
34 78
34 78
```

10. 연산자 5.cpp

```
 1 #include <iostream>
 2 using namespace std;
 3
 4 class point
 5 {
 6 private:
 7     int x, y;
 8 public:
 9     point(int _x = 0, int _y = 0):x(_x), y(_y) {}
10     void showposition(void);
11     point operator + (int val);
12 };
13
14 void point::showposition(void)
15 {
16     cout << x << " " << y << endl;
```

```
17 }
18
19 point point::operator+(int val)
20 {
21     point temp(x + val, y + val);
22     return temp;
23 }
24
25 int main(void)
26 {
27     point p1(3,7);
28     point p2 = p1 + 3;
29     p2.showposition();
30     return 0;
31 }
```

10-1. 결과 분석
6 10

11. 연산자 6.cpp

```
 1 #include <iostream>
 2 using namespace std;
 3
 4 class point
 5 {
 6 private:
 7     int x,y;
 8 public:
 9     point(int _x = 0, int _y = 0):x(_x), y(_y) {}
10      void showposition(void);
11      point operator+(int val);
12      friend point operator+(int val, point& p);
13 };
14
15 void point::showposition(void)
16 {
17     cout << x << " " << y << endl;
18 }
19
20 point point::operator+(int val)
21 {
22     point temp(x + val, y + val);
23     return temp;
```

```
24 }
25
26 point operator+(int val, point& p)
27 {
28     return p + val;
29 }
30
31 int main(void)
32 {
33     point p1(3,7);
34     point p2 = p1 + 3;
35     p2.showposition();
36
37     point p3 = 7 + p2;
38     p3.showposition();
39
40     return 0;
41 }
```

11-1. 결과 분석
6 10
13 17

12. counter.cpp

```
 1 #include <iostream>
 2 using namespace std;
 3
 4 class Counter
 5 {
 6 private:
 7     int val;
 8 public:
 9     Counter(void)
10     {
11         val = 0;
12     }
13
14     void Print(void)
15     {
16         cout << val << endl;
17     }
18
19     friend void SetVal(Counter& c, int val);    // friend 붙으면 private 에 접근 가능
20 };
```

```
21
22 void SetVal(Counter& c, int val)
23 {
24     c.val = val;
25 }
26
27 int main(void)
28 {
29     Counter cnt;
30     cnt.Print();
31     SetVal(cnt, 2002);
32     cnt.Print();
33
34     return 0;
35 }
```

12-1. 결과 분석
0
2002

13. friend.cpp

```
 1 #include <iostream>
 2 using namespace std;
 3
 4 class A
 5 {
 6 private:
 7     int data;
 8     friend class B;     // A 입장에서는 B 가 친구
 9 };
10
11 class B
12 {
13 public:
14     void SetData(A& a, int data)
15     {
16         a.data = data;
17     }
18     void print(A& c)
19     {
20         cout << c.data << endl;
21     }
22 };
23
```

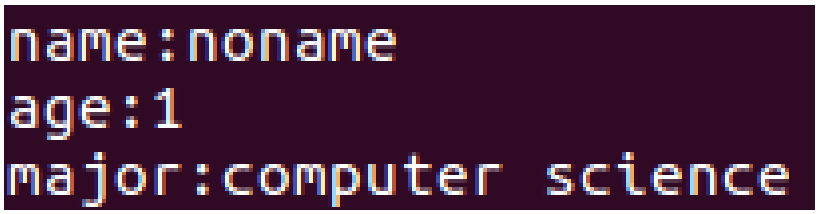
```
24 int main(void)
25 {
26     A a;
27     B b;
28     b.SetData(a, 10);
29
30     b.print(a);
31     return 0;
32 }
```

14. inheritance.cpp

```
 1 #include <iostream>
 2 #include <stdlib.h>
 3 #include <string.h>
 4 using namespace std;
 5
 6 class person
 7 {
 8     int age;
 9     char name[20];
10
11 public:
12     int getage(void) const
13     {
14         return age;
15     }
16     const char *getname(void) const // 함수 overloading 할 때, 함수 뒤에 const 를 붙임
17     {
18         return name;
19     }
20     person(int _age = 1, char *_name = "noname")
21     {
22         age = _age;
23         strcpy(name, _name);
24     }
25 };
26
27 class student: public person    // : public Person 붙이면 상속 person 에 있는 내용 사용 가능
28 {
29     char major[20];
30 public:
31     student(char *_major)
32     {
```

```
33        strcpy(major, _major);
34    }
35    const char *getmajor(void) const
36    {
37        return major;
38    }
39    void showdata(void) const
40    {
41        cout << "name:" << getname() << endl;
42        cout << "age:" << getage() << endl;
43        cout << "major:" << getmajor() << endl;
44    }
45 };
46
47 int main(void)
48 {
49    student park("computer science");
50    park.showdata();
51
52    return 0;
53 }
```

14-1. 결과 분석



15. template.cpp

```
 1 #include<iostream>
 2 using namespace std;
 3
 4 template <typename T>
 5 T add(T a, T b)
 6 {
 7     return a + b;
 8 }
 9
10 int main(void)
11 {
12     cout << add(10, 20) << endl;
13     cout << add(1.1, 2.2) << endl;
14
15     return 0;
16 }
```

15-1. 결과 분석

30
3.3