

1. 구조체와 포인터
2. 레퍼런스
3. 동적할당
4. 명령행에서 인자 받기

구조체와 포인터

```
struct namecard{  
    char name[20];  
    char job[30];  
    char tel[20];  
    char email[40];  
};
```

구조체 선언

```
namecard *p;
```

구조체 포인터 선언

```
namecard x={ "전수빈",  
             "치과의사",  
             "356-0868",  
             "eldy@naver.com"  
};
```

구조체 변수의 선언 및 초기화

```
p=&x;
```

구조체 포인터에 구조체 변수의 주소값 저장

멤버(필드) 접근

구조체와 멤버참조 연산자

(*p).name;

p->name;

종류	사용법
. 연산자	구조체 변수명.멤버
-> 연산자	구조체 포인터 변수명 -> 멤버

레퍼런스 (참조변수)

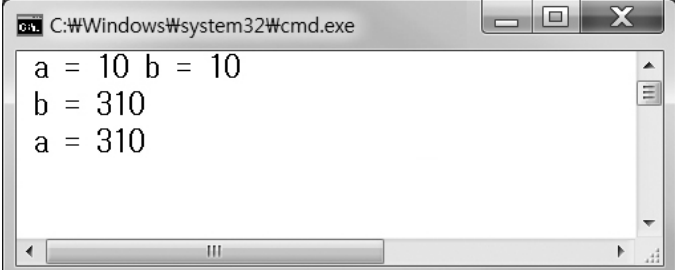
- ▶ 다른 변수를 가리키는 변수로 다른 변수와 동일한 메모리 위치를 서로 공유함
- ▶ 변수의 별명!

```
int x = 5;  
// x 앞에 앰퍼샌드를 두지 않는다는 점에 유의해야 한다!  
int &ref = x;
```

참조 변수를 선언할 때 주의할 점은 반드시 변수 선언 시 초기값을 주어야 한다는 것이다. 참조 변수는 별명이므로 이미 존재하는 변수에 대해 이름만 하나 더 추가하는 것이기 때문이다.

- ▶ 포인터와 다른점
 - ▶ NULL을 가리킬 수 없다.
 - ▶ 한번 초기화되면 자신이 가리키는 메모리를 변경할 수 없다.

```
01 #include <iostream>
02 using namespace std;
03 void main()
04 {
05 int a=10;
06 int &b = a;
07 cout<<" a = "<<a<<" b = "<<b<<endl;
08 b+=300;
09 cout<<" b = "<<b<<endl;
10 cout<<" a = "<<a<<endl;
11 }
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The output of the program is displayed as follows:

```
a = 10 b = 10
b = 310
a = 310
```

new로 더 많은 메모리 확보하기-동적 할당

- ▶ 프로그램이 실행하고 있는 동안에도 메모리를 필요한 만큼 요청하는 것

- ▶ new

- ▶ 자유 저장소(힙)의 메모리로 포인터를 초기화할 때는 new 키워드를 사용

```
int *p_int = new int;
```

- ▶ 정수 값이 담길 충분한 공간의 메모리를 리턴
 - ▶ p_int는 그 메모리를 가리키도록 설정됨
 - ▶ 메모리를 할당만 하고 반환하지 않으면 메모리 부족 현상이 발생

- ▶ delete

- ▶ new를 통해 할당된 메모리를 다시 거둬들임

```
delete p_int; delete p_int;  
p_int = NULL;
```

- ▶ 레퍼런스와 동적 할당? NO!!

```
int &val = *(new int);
```



이제 **new** 사용하여 동적으로 배열 만들 수 있다

- ▶ 포인터를 배열처럼!
- ▶ 즉, 배열을 동적으로 만들 수 있음
 - ▶ 배열을 동적으로 만들면 필요한 메모리를 런타임에 결정할 수 있어 편리함
 - ▶ 예제

```
int count_of_numbers;  
cin >> count_of_numbers;  
int *p_numbers = new int[ count_of_numbers ];
```

Main 함수와 명령행 인자

```
void main(int argc, char *argv[ ])
```

main() 함수의 형태가 위와 같이 매개변수 2개를 갖는 형태라면 명령행에서 입력한 문자열을 얻어올 수 있다.

- int argc: 정수 수치로 명령행에서 입력한 문자열의 개수를 알려준다.
- char *argv[]는 문자열을 가리키는 포인터 배열의 시작 주소

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

int main(int argc, char *argv[])
{
    cout<<" Number of string in the commandline =" << argc << "\n\n";

    cout<<" >> Strings in the commandline << \n";
    for(int i=0; i<argc; i++)
        cout<<" argv[ "<<i <<" ] =" << argv[i] <<"\n";

    return 0;
}
```

```
rhichaes-iMac:C++ chae.rhee$ ./test.exe  
Number of string in the commandline =1
```

```
>> Strings in the commandline <<  
argv[ 0 ] =./test.exe
```

```
rhichaes-iMac:C++ chae.rhee$ ./test.exe I  
Number of string in the commandline =2
```

```
>> Strings in the commandline <<  
argv[ 0 ] =./test.exe  
argv[ 1 ] =I
```

```
rhichaes-iMac:C++ chae.rhee$ ./test.exe I am fine  
Number of string in the commandline =4
```

```
>> Strings in the commandline <<  
argv[ 0 ] =./test.exe  
argv[ 1 ] =I  
argv[ 2 ] =am  
argv[ 3 ] =fine
```

```
rhichaes-iMac:C++ chae.rhee$ ./test.exe I am fine thank you  
Number of string in the commandline =6
```

```
>> Strings in the commandline <<  
argv[ 0 ] =./test.exe  
argv[ 1 ] =I  
argv[ 2 ] =am  
argv[ 3 ] =fine  
argv[ 4 ] =thank  
argv[ 5 ] =you
```

```
rhichaes-iMac:C++ chae.rhee$
```


TEAM EX1 레퍼런스를 사용한 SWAP

```
#include <iostream>

using namespace std;

void swap1 (int left, int right)
{
    int temp = left;
    left = right;
    right = temp;
}

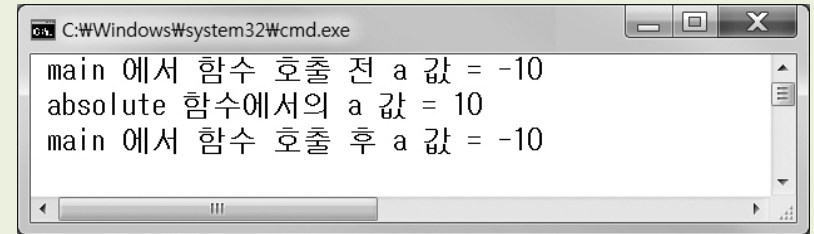
void swap2 (int *p_left, int *p_right)
{
    int temp = *p_left;
    *p_left = *p_right;
    *p_right = temp;
}

int main ()
{
    int x = 1, y = 2;
    swap1( x, y );
    cout << x << " " << y << '\n';
    swap2( &x, &y );
    cout << x << " " << y << '\n';
}
```

```
void swap3(int &r_left, int &r_right)
    swap3 (x, y);
```

TEAM EX2 다양한 형태의 매개변수 테스트

```
01 #include <iostream>
02 using namespace std;
03 void absolute(int a);
04 void main()
05 {
06     int a=-10;
07     cout<<" main 에서 함수 호출 전 a 값 = "<< a <<"\n";
08     absolute(a);
09     cout<<" main 에서 함수 호출 후 a 값 = "<< a <<"\n";
10 }
11
12 void absolute(int a)
13 {
14     if(a<0)
15         a=-a;
16     cout<<" absolute 함수에서의 a 값 = "<< a <<"\n";
17 }
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The output of the program is displayed as follows:

```
main 에서 함수 호출 전 a 값 = -10
absolute 함수에서의 a 값 = 10
main 에서 함수 호출 후 a 값 = -10
```

포인터 매개변수로 호출
레퍼런스 매개변수로 호출

TEAM EX3 동적 할당으로 생성된 배열 사용하기

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

```
int main()
{
```

이 코드에서 틀린 점을 찾아보세요

```
    int cnt;
    cin >> cnt;
```

```
    int*p_numbers = new int[cnt];
```

```
    for (int i = 0; i < cnt; i++)
        *(p_numbers + i) = i;
```

← 이해할 수 있나요?

```
    for (int i = 0; i < cnt; i++)
        cout << p_numbers[i];
```

← 이해할 수 있나요?

```
    return 0;
```

```
}
```

TEAM EX4 동적 할당으로 생성된 배열 사용하기

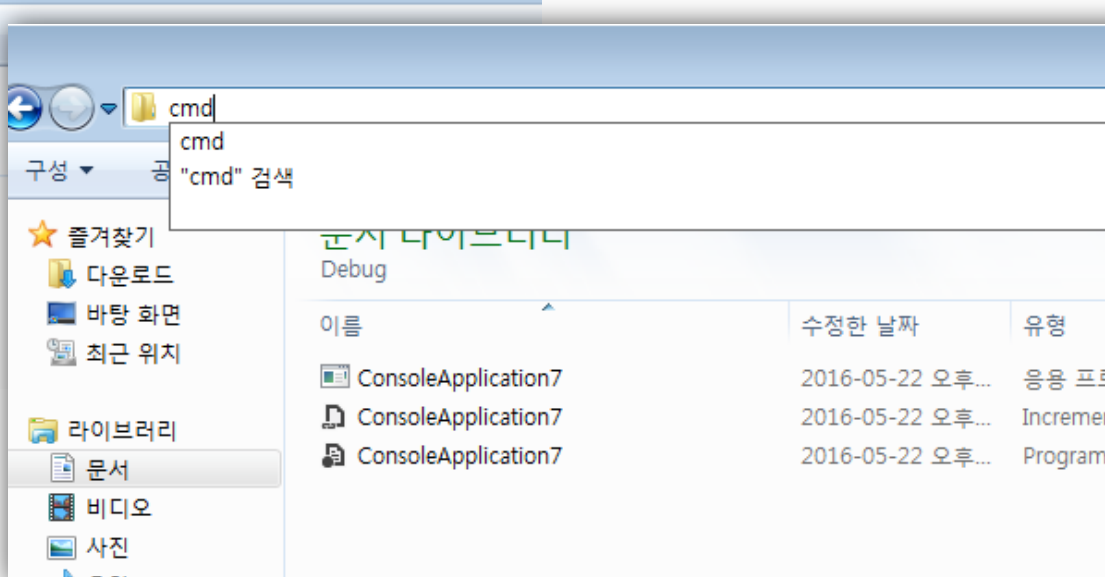
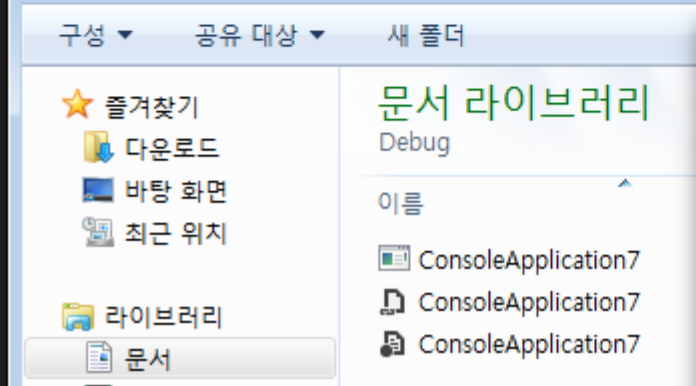
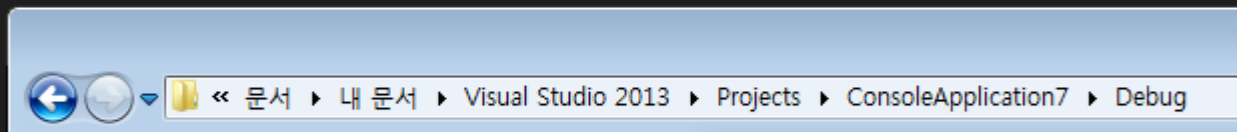
예제설명 사용자가 입력한 여러 숫자 중에서 짝수 합계만 출력하는 프로그램

실행결과

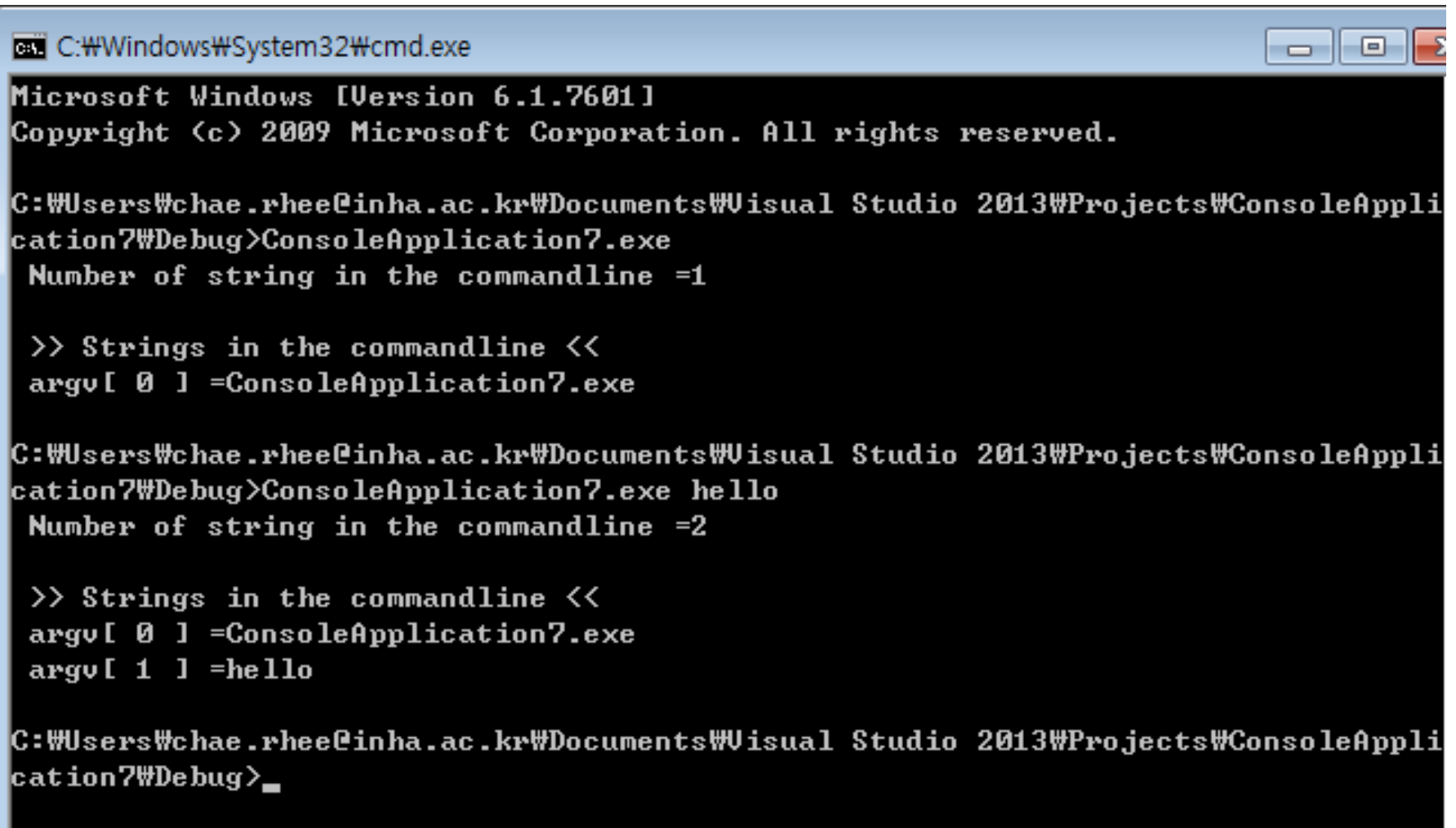
```
rhichaes-iMac:C++ chae.rhee$ ./test.exe
how many numbers? :6
1th numbers: 10
2th numbers: 4
3th numbers: 5
4th numbers: 3
5th numbers: 5
6th numbers: 90
total: 104
```

TEAM EX5 명령행에서 프로그램 실행해보기 (1)

```
Source.cpp  x
ConsoleApplication7 (Global Scope)
1  #include <iostream>
2  using namespace std;
3
4  int main(int argc, char *argv[])
5  {
6
7      cout << " Number of string in the commandline =" << argc << "\n\n";
8
9      cout << " >> Strings in the commandline << \n";
10     for (int i = 0; i<argc; i++)
11         cout << " argv[ " << i << " ] =" << argv[i] << "\n";
12
13     return 0;
14 }
15
```



TEAM EX5 명령행에서 프로그램 실행해보기 (2)



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\chae.rhee@inha.ac.kr\Documents\Visual Studio 2013\Projects\ConsoleApplication7\Debug>ConsoleApplication7.exe
Number of string in the commandline =1

>> Strings in the commandline <<
argv[ 0 ] =ConsoleApplication7.exe

C:\Users\chae.rhee@inha.ac.kr\Documents\Visual Studio 2013\Projects\ConsoleApplication7\Debug>ConsoleApplication7.exe hello
Number of string in the commandline =2

>> Strings in the commandline <<
argv[ 0 ] =ConsoleApplication7.exe
argv[ 1 ] =hello

C:\Users\chae.rhee@inha.ac.kr\Documents\Visual Studio 2013\Projects\ConsoleApplication7\Debug>_
```