

LAB #11



경희대학교
KYUNG HEE UNIVERSITY

● 실습 목표 – 문자열의 이해와 활용

1. 문자열 개념의 이해
2. string 객체 생성자의 이해
3. 공백을 포함한 문자열 읽기
4. c++ string을 c string으로 상호 변환
5. 문자열의 배열 및 string 객체의 멤버 함수의 이해
6. string 객체의 멤버 함수의 이해
7. 문자열 탐색의 이해
8. 문자열을 이용하여 모르스 부호 전환 과정을 이해

14. 문자열 | Practice 1

• 실습 문제 1 <Section 14-1>

목표 : 문자열 개념의 이해

- 제시된 결과가 보여 지도록 문자열을 출력하는 부분을 완성하라.

```
int main ()
{
    char s1[12]={'S','t','r','i','n','g',' ',' ','t','e','s','t','\0'};
    //s1 출력, 마지막 '\0'가 없으면 문자열 보장되지 않음.

    s1[6]=0;
    //s1 출력, 결과의 원인을 생각 해보라.

    char s2[12]="String test";
    //s2 출력

    cout << endl;
    cout << "s1      |      s2" << endl;
    cout << "_____" << endl;
    for( int i=0 ; i<12 ; i++ )
        //s1과 s2의 i번째 요소를 int형으로 변환하여 출력

    //int형 변환 출력으로 문자열 관리에 있어서 '\0' 의
    //의미를 확인하라

    return 0;
}
```

output

```
s1 : String test
s1 : String
s2 : String test
```

s1		s2
83		83
116		116
114		114
105		105
110		110
103		103
0		32
116		116
101		101
115		115
116		116
0		0

두 번째 s1의 출력 결과에 영향

계속하려면 아무 키나 누르십시오 . . .

14. 문자열 | Practice 1

• 실습 문제 2 <Section 13-2>

목표 : string 객체 생성자의 이해

- 기본 정보 : string 객체의 생성자들

std::string 객체의 생성자

- `string ();`
 - 기본 생성자
- `string (const string& str);`
 - str 객체를 복사하는 생성자
- `string (const string& str, size_t pos, size_t n = npos);`
 - str 객체 중 pos번째 문자부터 n개의 문자로 초기화 하는 생성자
- `string (const char * s, size_t n);`
 - 입력된 s의 문자열 중 n개의 문자로 초기화 하는 생성자
- `string (const char * s);`
 - 입력된 s의 문자열로 초기화 하는 생성자
- `string (size_t n, char c);`
 - c문자 n개로 초기화하는 생성자
- `template<class InputIterator> string (InputIterator begin, InputIterator end);`
 - template class InputIterator로 지정된 시작과 끝으로 초기화하는 생성자

14. 문자열 | Practice 1

• 실습 문제 2 <Section 14-2>

목표 : string 객체 생성자의 이해

- string 객체의 다양한 생성자를 활용하여 아래 프로그램을 완성 하라.

```
int main ()
{
    string s0 ("Initial string");

    // constructors used in the same order as described above:
    //기본 생성자를 이용한 s1 객체 선언
    //s0 객체를 복사하는 s2 객체 선언
    //s0 객체 중 8번째 문자부터 3개의 문자로 초기화하는 s3 객체 선언
    //"A character sequence"의 문자열 중 6개의 문자로 초기화하는 s4 객체 선언
    //"Another character sequence"의 문자열로 초기화하는 s5 객체 선언
    //'x' 문자 10개로 초기화하는 s6 객체 선언

    string s7 ( s0.begin(), s0.end()-4 );
    //template class iterator로 지정 된 시작과 끝으로 초기화하는 객체 s7 선언

    cout << "s1: "    << s1 << "Wns2: " << s2 << "Wns3: " << s3;
    cout << "Wns4: "  << s4 << "Wns5: " << s5 << "Wns6: " << s6;
    cout << "Wns7: "  << s7 << endl;
    return 0;
}
```

output

```
s1:
s2: Initial string
s3: str
s4: A char
s5: Another character sequence
s6: xxxxxxxxxxxx
s7: Initial st
계속하려면 아무 키나 누르십시오 . . .
```

14. 문자열 | Practice 1

• 실습 문제 3 <Section 14-3>

목표 : 공백을 포함한 문자열 읽기

- getline함수를 이용하여 공백이 포함된 문자열을 입력 받아 출력하는 아래 프로그램을 완성 하라.

```
int main( void )  
{  
    string name1;  
    string name2;  
    string lastname ;  
    string firstname;
```

output

```
cout << "Enter your name ( Example : W"Hong Gil DongW" ): ";  
//공백이 포함된 성과 이름을 getline함수를 이용하여 name1에 저장  
cout << "Your name is " << name1 << endl << endl;  
  
cout << "Enter your name ( Example : W"Gil Dong, HongW" ) :";  
//getline 함수를 이용 ","로 종결 된 이름을 읽어 firstname에 저장  
//getline 함수를 이용 성을 읽어 lastname에 저장  
cout << "Your name is " << lastname << " " << firstname << endl << endl;  
  
cout << "Enter your name ( Example : W"Hong Gil DongW" ): ";  
cin >> name2;  
cout << "Your name is " << name2 << endl;  
//name2에서 공백이 포함된 경우 공백에서 읽기가 중단 됨을 확인 한다.  
  
return 0;  
}
```

```
Enter your name < Example : "Hong Gil Dong" >: Hong Gil Dong (Enter)  
Your name is Hong Gil Dong  
  
Enter your name < Example : "Gil Dong, Hong" > :Gil Dong, Hong (Enter)  
Your name is  Hong Gil Dong  
  
Enter your name < Example : "Hong Gil Dong" >: Hong Gil Dong (Enter)  
Your name is Hong  
계속하려면 아무 키나 누르십시오 . . . ■
```

14. 문자열 | Practice 1

• 실습 문제 4 <Section 14-5>

목표 : C++ string을 c string으로 상호 변환

- c++ string을 c string으로 상호 변환하는 아래의 프로그램을 완성하라.
 - strcpy(char* pdst, const char* psrc) : psrc를 pdst로 복사하는 함수
 - std::string객체의 연산자 및 함수
 - = 연산자 : c string(char*) 을 입력하여 변수 값의 초기화
 - c_str() : c string(char*)으로 변환하여 반환.

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

```
int main ( void )
{
    string str1 = "string convert test.";
    string str2 ;
    char *ary   = new char[str1.length()+1];
    //c_str() 메소드와 strcpy()함수를 이용하여 str1 C++ string object를 ary C string으로 복사
    //C++ string assignment operator( "=" )를 이용하여 ary C string을 str2 C++ string object로 복사
    cout << "ary : " << ary << endl;
    cout << "str2: " << str2 << endl;

    delete[] ary;
    return 0;
}
```

output

```
ary : string convert test.
str2: string convert test.
계속하려면 아무 키나 누르십시오 . . .
```

14. 문자열 | Practice 1

• 실습 문제 5 <Section 14-4>, <Section 14-5>

목표 : 문자열의 배열 및 string 객체의 멤버 함수의 이해

- 문자열을 배열로 선언하고, 각각의 배열 요소에 요일을 저장하여 테스트하는 아래 프로그램을 완성하라.
- string객체의 연산자 및 함수
 - = 연산자 : 변수 값의 초기화
 - + 연산자 : 문자열을 결합
 - += 연산자 : 문자열을 추가 대입
 - size(), length() : 객체의 문자열 길이를 넘겨 줌

```
int main( void )
{
    //7개의 원소를 갖는 string 객체 배열 daysAry 선언
    //daysAry 배열의 각 원소를 요일로 초기화
    string strDOW;

    cout << setw(24) << "The days of the week" << endl;
    for( int index = 0 ; index<7 ; index++ ) {
        //daysAry 배열의 index번째 객체 출력
        cout << ", length : ";
        //daysAry배열의 index번째 객체의 문자열 길이를 출력
        cout << endl;
        //공백( " ")과 daysAry 배열의 index번째 객체를 결합한 것을 strDOW에 추가 대입
    }
    cout << endl << strDOW << endl;
    return 0;
}
```

output

```
The days of the week
Sunday, length : 6
Monday, length : 6
Tuesday, length : 7
Wednesday, length : 9
Thursday, length : 8
Friday, length : 6
Saturday, length : 8
```

```
Sunday Monday Tuesday Wednesday Thursday Friday Saturday
계속하려면 아무 키나 누르십시오 . . .
```


14. 문자열 | Practice 1

• 실습 문제 6 <Section 14-5>

목표 : string 객체의 멤버 함수의 이해

- std::string 객체의 멤버 함수를 테스트하는 아래 프로그램을 완성하라.

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

int main( void )
{
    string s1, s2;
    //문자열 검색
    s1 = "01234560123456";
    int index = s1.find(/*파라메터 채우기*/); //0번째 위치부터 검색하여 "23"의 substring의 위치를 찾아라.
    cout<< setw(10) << "find : " << s1 << " -> " << index << " : " << s1[index] << s1[index+1] << endl;

    //문자열 추출
    s1 = "0123456";
    s2 = s1.substr(/*파라메터 채우기*/); //0번째부터 3개의 문자를 잘라 할당하라.
    cout << setw(10) << " substr : " << s1 << " (0~2) -> " << s2 << endl ;

    //문자열 삽입
    s1 = "0123456";
    s2 = s1;
    s1.insert(/*파라메터 채우기*/); // "add" 문자열을 0번째에 끼워 넣어라.
    cout << setw(10) << " insert : W"addW" + " << s2 << " = " << s1 << endl ;
```

14. 문자열 | Practice 1

• 실습 문제 6 <Section 14-5>

목표 : string 객체의 멤버 함수의 이해

//문자열 추가

```
s1 = "0123456";
s2 = "abcdef";
cout << setw(10) << " append : " << s1 << " append " << s2 << " = ";
s1.append(/*파라메터 채우기*/); //s2의 0~s2.npos 위치의 문자열을 s1의 맨뒤에 추가하라.
cout << s1 << endl;
```

//문자열 교체

```
s1 = "abcdef";
s2 = "0123456";
s1.replace(/*파라메터 채우기*/); //s1의 0번째 위치에서 1개의 문자를 "5555"로 대체 하라.
cout << setw(10) << "replace : " << s1 << endl;
```

//문자열 삭제

```
s1 = "0123456";
cout << setw(10) << "erase : " << s1 ;
s1.erase( /*파라메터 채우기*/ ); //s1의 1번째 위치에서 1개의 문자를 삭제하라.
cout << " -> " << s1 << endl;
```

//문자열 비교

```
int nCmp;
s1 = "1234567";
s2 = "1234999";
nCmp = s1.compare(/*파라메터 채우기*/); //s1과 s2를 비교 하라.
cout << setw(10) << "compare : " << ( (nCmp==0) ? "equal" : ( (nCmp<0) ? "less" : "greater" ) ) << endl;
nCmp = s1.compare(/*파라메터 채우기*/); //s1의 0~4의 문자열과 s2의 0~4의 문자열을 비교 하라.
cout << setw(10) << "compare : " << ( (nCmp==0) ? "equal" : ( (nCmp<0) ? "less" : "greater" ) ) << endl;
```

}

output

```
find : 01234560123456 -> 2 : 23
substr : 0123456 <0~2> -> 012
insert : "add" + 0123456 = add0123456
append : 0123456 append abcdef = 0123456abcdef
replace : 5555bcedf
erase : 0123456 -> 023456
compare : less
compare : equal
계속하려면 아무 키나 누르십시오 . . .
```

14. 문자열 | Practice 1

• 실습 문제 7 <Section 14-5>

목표 : 문자열 탐색의 이해

- 대칭 문자를 사용하면 아주 쉬운 암호화 문자열을 구성 할 수 있다. 아래의 대칭문자를 이용하여 입력 받은 문자열을 암호화하고, 암호를 푸는 프로그램을 완성하라.
- string객체의 함수
 - string::find_first_of(string str) : 입력된 str이 처음으로 나타나는 위치를 탐색하여, size_t형으로 리턴, 문자열 안에서 str을 탐색 할 수 없으면 string::npos(0xFFFFFFFF)의 상수가 반환 된다.
 - string::iterator : 문자열 내의 문자 탐색을 위한 반복자 자료 형. 포인터 형으로 string 객체에서 char 형으로 변환 됨. 예) string::iterator iter= string::begin(); iter++; char c = *iter;
 - string::begin() : string 객체에서 문자열의 시작 위치를 string::iterator형으로 반환.
 - string::end() : string 객체에서 문자열의 끝 위치를 string::iterator형으로 반환.
- 암호화 대칭 문자열
 - 입력 : ABCDEFGHIJKLMNOPQRSTUVWXYZ,?!;
 - 출력 : ?Q.W,EMRNTBXYUV!ICOPZA;SDFKJGH
 - A는 ?로 대체 되는 방법으로 암호 화하는 방법.
 - 예 : STRING → OPCNUM

```
//참고
string strSrc = "ABC";
size_t index = strSrc.find_first_of( 'B' );
strSrc.at( index );//B
strSrc[index]//B
```

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	.	,	?	!	;
?	Q	.	W	,	E	M	R	N	T	B	X	Y	U	V	!	I	C	O	P	Z	A	;	S	D	L	F	K	G	J	H

14. 문자열 | Practice 1

• 실습 문제 7 <Section 14-5>

목표 : 문자열 탐색의 이해

```
string strEnglish = "ABCDEFGHIJKLMNOPQRSTUVWXYZ.,?!:";
string strCode    = "?Q.W,EMRNTBXYUV!ICOPZA;SDFKGJH";

char findString( string& source, string& target, const char& c )
{
    //source에서 입력된 c를 탐색하여 target에서 같은 위치에서의 문자 리턴
    //입력된 c가 source에서 탐색 되지 않는다면, 입력과 같은 문자를 리턴
}

string Encode( string& str )
{
    string rst;
    string::iterator iter = str.begin(); // iter 을 통해 한 문자씩 접근
    while (iter != str.end()) // iter이 str.end()와 같지 않을 때까지 반복
    {
        //findString함수 호출 (source:strEnglish, target:strCode, *iter)
        //rst에 findString함수에서 반환 된 문자를 추가
        //iter 증가
    }
}

string Decode( string& str )
{
    //findString함수에서 strCode를 source로 strEnglish를 target으로 하여
    //입력된 str의 암호를 풀어 리턴
}
```

```
int main( void )
{
    string msg;
    cout << "Enter your string." << endl;
    getline( cin, msg );

    string enc = Encode( msg );
    string dec = Decode( enc );

    cout << "Encoded string : " << enc
    << endl;
    cout << "Decoded string : " << dec
    << endl;
    return 0;
}
```

output

```
Enter your string.
DO YOUR BEST.
Encoded string : WU DUZC Q,OPF
Decoded string : DO YOUR BEST.
계속하려면 아무 키나 누르십시오 . . .
```

14. 문자열 | Practice 1

• 실습 문제 8 <Section 14-6>

목표 : 문자열을 이용하여 모尔斯 부호 변환 과정을 이해

- 영문을 모尔斯 부호로, 모尔斯 부호를 영문으로 변환하는 Morse class를 완성하고, 사용자 선택 메뉴에 따라 문자열을 입력 받고 코드 변환을 수행하는 아래 프로그램을 완성 하라.

- Morse class

- private: bool convert(string& str1, int col, string& str2);
 - 영문에서 모尔斯 부호로 모尔斯 부호에서 영문으로 변환
 - col==0 : encode, col==1 : decode
- public: string encode(string& inStr)
 - 영문 inStr을 모尔斯 부호로 인코딩하여 반환
- public: string decode(string& inStr)
 - 모尔斯 부호 inStr을 영문으로 디코딩하여 반환

- 기타 참고 함수

- toupper() : 입력 문자가 영문 소문자라면, 영문 대문자로 변환 하여 반환

문자	부호	문자	부호
A	· -	N	--·
B	--··	O	---
C	--··	P	·---
D	--·	Q	---·
E	·	R	·--
F	····	S	···
G	--·	T	-
H	····	U	··-
I	··	V	··--
J	·---	W	·---
K	--·	X	---·
L	····	Y	---·
M	--	Z	----

14. 문자열 | Practice 1

• 실습 문제 8 <Section 14-6>

```
bool Morse::convert( string& str1, int col,
                    string& str2) {
```

```
    string encDec[27][2] = {
        { "A", ".-#" },
        { "B", "-...#" },
        { "C", "-.-.#" },
        { "D", "-..#" },
        { "E", ".#" },
        { "F", "-.-.#" },
        { "G", "--.#" },
        { "H", "...#" },
        { "I", "..#" },
        { "J", ".--#" },
        { "K", "-.-#" },
        { "L", "-.-.#" },
        { "M", "--#" },
        { "N", "-.#" },
        { "O", "---#" },
        { "P", ".--.#" },
        { "Q", "--.-#" },
        { "R", "-.-#" },
        { "S", "...#" },
        { "T", "-#" },
        { "U", "-.-#" },
        { "V", "...-#" },
        { "W", "-.-#" },
        { "X", "-.-.#" },
        { "Y", "-.-.#" },
        { "Z", "--.#" },
        { " ", "$$#" }
```

		col	
row			
		[0]	[1]
[0]	A		.-#
[1]	B		-...#
			⋮
[25]	Z		--.#
[26]	공백		\$\$#

```
};
```

```
bool found = false;
```

```
int row;
```

```
//입력된 str1이 encDec에서 같은 문자열이 존재 할 때까지만 탐색
//입력 된 col에 따라서 영문에서 탐색 할 수도 있고,
//모르스 부호에서 탐색 될 수 도 있다.
//col이 0이면 영문에서 탐색, 1이면 모르스 부호에서 탐색.
```

```
//encDec[row][col]의 이차원 문자열 배열에서,
```

```
//배열에서 검색된 결과는 (row-1)에 위치
```

```
//col이 0이면 1의 위치 문자열을 1이면 0의 위치의 문자열을 저장.
```

```
return found;
```

```
}
```

```
string Morse::encode( string& inStr ){
```

```
    string s1;
```

```
    string s2;
```

```
    string outStr = "";
```

```
    bool error = false;
```

```
    size_t curr = 0;
```

```
//입력된 모든 문자열을 탐색하여 모르스 부호로 변환
```

```
// inStr 탐색
```

```
{
```

```
    //inStr에서 cur번째 문자를 대문자로 변환하여 s1에 대입
```

```
    //convert함수 호출, (s1, 0, s2 );
```

```
    //outStr에 s2를 추가
```

```
    //cur을 1 증가
```

```
}
```

```
if( !error ) return outStr;
```

```
else return ("Invalid character in CodeWn");
```

```
}
```

14. 문자열 | Practice 1

• 실습 문제 8 <Section 14-6>

```
string Morse::decode( string& inStr ){  
    //모르스 부호의 끝은 #로 구분되어 있으므로,  
    //입력 문자열 inStr에서 #를 찾아 단위 모르스 부호 탐색.  
    //탐색 된 단위 모르스 부호를 영문으로 변환.  
}
```

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

```
#define FLUSH while( cin.get() != 'Wn' )
```

```
char menu      ( void );  
string getInput ( void );  
void printOutput ( string& inStr, string& outStr );
```

```
int main ( void )  
{
```

```
    string inStr ;  
    string outStr;  
    bool  done = false;  
    char  option;  
    Morse mCode;
```

```
    while( !done )  
    {  
        option = menu();  
        //메뉴에 따라 진행  
        //'E'라면 입력 영문을 모르스 부호로 변환  
        //'D'라면 입력 모르스 부호를 영문으로 변환  
    }  
    return 0;  
}  
char menu( void )  
{  
    bool validData = true;  
    char option;  
  
    //메뉴 출력  
  
    //입력된 메뉴가 올바른지 확인  
    //소문자가 입력 되면 대문자로 변환  
    //올바르지 않은 메뉴 입력 시 에러 출력  
  
    return option;  
}  
string getInput ( void ){  
    //사용자로부터 입력 받은 문자열을 반환  
}  
void printOutput ( string& inStr, string& outStr ){  
    //inStr, outStr을 화면에 각각 출력  
}
```

14. 문자열 | Practice 1

• 실습 문제 8 <Section 14-6>

output

```
M E N U
E>  encode
D>  decode
Q>  quit
```

메뉴 출력

Enter option : press return key : E

메뉴 선택

Please enter line of text to be coded :

I WILL BE THE BEST

문자열 입력

The information entered was:

I WILL BE THE BEST

The transformed information is

..#\$\$\$#.-#.#.-#.-#.#\$\$\$#-...#.#\$\$\$#-#....#.#\$\$\$#-...#.#...#-#

모르스 부호 인코딩 결과

```
M E N U
E>  encode
D>  decode
Q>  quit
```

Enter option : press return key : D

Please enter line of text to be coded :

..#\$\$\$#.-#.#.-#.-#.#\$\$\$#-...#.#\$\$\$#-#....#.#\$\$\$#-...#.#...#-#

모르스 부호 입력

The information entered was:

..#\$\$\$#.-#.#.-#.-#.#\$\$\$#-...#.#\$\$\$#-#....#.#\$\$\$#-...#.#...#-#

The transformed information is

I WILL BE THE BEST

모르스 부호 디코딩 결과

```
M E N U
E>  encode
D>  decode
Q>  quit
```

Enter option : press return key : Q

계속하려면 아무 키나 누르십시오 . . .