



C++

10:

String

2007. 5. 23.

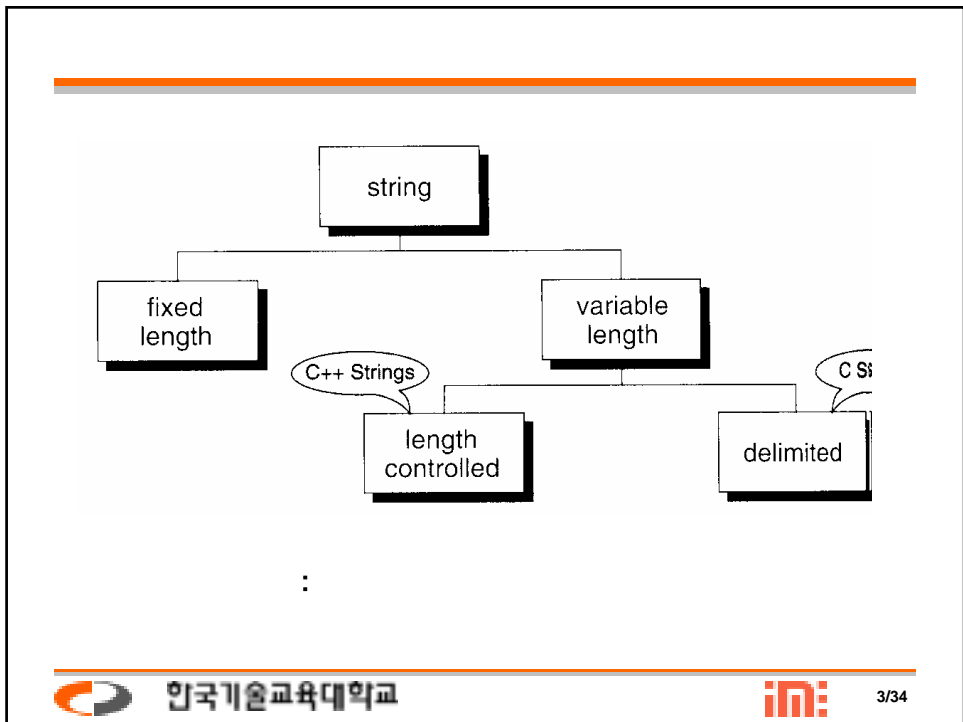
:

E mail: jaesoo27@kut.ac.kr

1

-
- - String
 - String
 - C C++
 -





- (fixed - length string)
 - .
 - ,
 - 가 ,
 - 가
 - 가



■ 가

- (length - controlled strings): C++
string

- : C string

■

- 가

- 255 ,

1

- 가

- C++



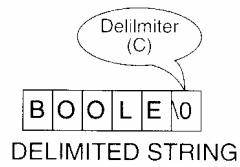
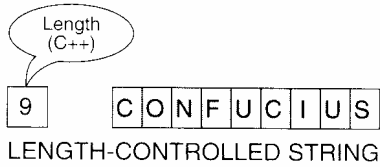
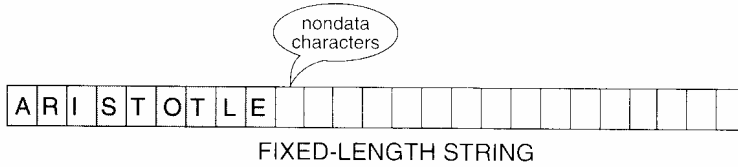
■

- , 가
- 가 ASCII

null (\ 0) .

- C

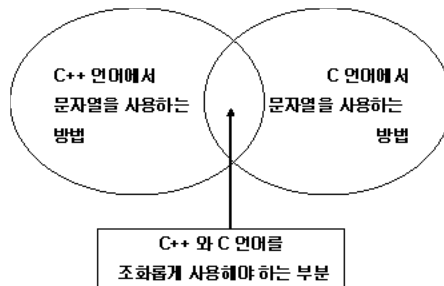




:



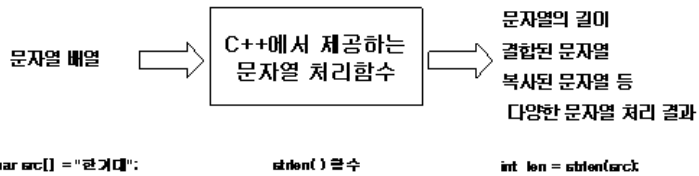
1.



[10-1] C C++



1.



[10-2] C



```
1: #include <iostream>
2: #include <cstring>
3: using namespace std;
4:
5: int main(){
6: //
7: char src[] = "관개대";
8:
9:
10: //
11: // 1.
12: int len = strlen(src);
13:
14: // 2.
15: char* dest = new char [len + 1];
16:
17: // 3.
18: strcpy(dest, src);
19:
20: //
21: cout << "src = " << src << "\n";
22: cout << "dest = " << dest << "\n";
23:
24: //
25: delete[] dest;
26: dest = NULL;
27: return 0;
28: }
```

```
1: #include <iostream>
2: #include <cstring>
3: using namespace std;
4:
5: int main()
6: {
7: //
8: char str1[50] = "관개대";
9: char str2[] = "관개대";
10:
11: //
12: strcat(str1, str2);
13:
14: //
15: if ( strcmp( str1, "관개대관개대" ) == 0 )
16: cout << "str1 and \n" << str2
17: \n are identical.\n";
18: //
19: if ( strcmp( "관개대", str1 ) != 0 )
20: cout << "\n" << str1 << "\n" << str2 << "\n" << "
21: \n and str1 are NOT identical.\n";
22:
23: return 0;
24: }
```



주소

99	
100	'a'
101	'b'
102	'c'
103	'd'
104	'e'
105	'f'
106	'g'
107	0
108	

char str[] = "abcdefg";



"str이라는 문자열 변수의 주소는 100
번지라고 말한다"

[10-3]



1.

- char* 가
- void Function(char* p);
 - 가
 - // char
 - char c = 'A';
 - Function(&c);
 - - //
 - char s[] = "ABCDEG";
 - Function(s); // &s[0]
- void Function(char *p)
 - {
 - // char p char ?
 - ..
 - }



1.



```
// p:
```

```
void Function(char* p);
```



```
char*
```

```
가 가
```



```
가
```

```
C++
```

```
cout
```



1.

```
1: #include <iostream>
2: using namespace std;
3:
4: int main()
5: {
6:     //
7:     char c = 'A';
8:
9:     //
10:    char s[] = "This is a string.";
11:
12:    //          cout
13:    cout << s << " \n";
14:    cout << &s[0] << " \n";
15:
16:    //          cout
17:    cout << &c << " \n";
18:    //
19:    cout << c << " \n";
20:    return 0;
21: }
```

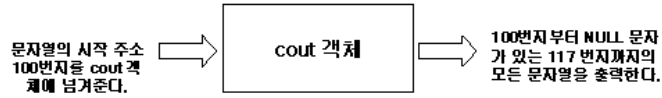
```
C:\Program Files\DesktopPC\Desktop\Wzawson\c++\10-4 >
This is a string.
This is a string.
A
A
Press any key to continue.
```

[10-4] C++ char*



1.

주소 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118
? 'T' 'h' 'i' 's' ' ' 'i' 's' ' ' 'a' ' ' 's' 't' 'r' 'i' 'n' 'g' ' ' 0 ?



```
cout << s ;  
cout << &s[0];
```

[10-5] cout 가 s &s[0]

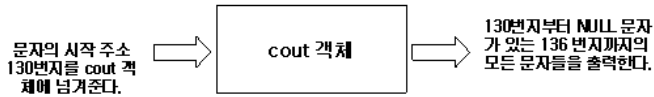


1.

주소 129 130 131 132 133 134 135 136 137 138 139 140
? 'A' ? ? ? ? ? 0 ? ? ? ? 0

↑
1바이트만 변수
c의 영역이다.

↑
우연히 136번지에 NULL
문자가 들어 있는 경우



```
cout << &c;
```

[10-6] cout 가 &c



- char* 가
- char ' 가
- ,
- .

- char* 가
- .



2. C++

- C++ 가
- C++ STL (Data Structure) (Algorithms)
- STL C++
- , STL
- STL



2. string

1) string1.cpp

```
#include <iostream>
#include <string>
using std::endl;
using std::cout;
using std::cin;

using std::string;
```

```
int main()
{
    string str1="Good ";
    string str2="morning";
    string str3=str1+str2;
    cout<<str1<<endl;
    cout<<str2<<endl;
    cout<<str3<<endl;

    str1+=str2;
    if(str1==str3)           //str1 str3
        cout<<"equal!"<<endl;

    string str4;
    cout<<"          : ";
    cin>>str4;
    cout<<"          : "<<str4<<endl;
    return 0;
}
```

2. string

■ string1.cpp

string

- , , , , : string
-
- string str1("Good");
- +
- <<
- +=
- ==
- >>

3. string

```
using std::ostream;
using std::istream;

class string{
    int len;
    char* str;
public:
    string(const char* s=NULL);
    string(const string& s);
    ~string();
    string& operator=(const string& s); //
    string& operator+=(const string& s);
    bool operator==(const string& s);
    string operator+(const string& s); // return type string&가 string ?

    friend ostream& operator<<(ostream& os, const string& s);
    friend istream& operator>>(istream& is, string& s);
};
```



3. string

```
string::string(const char* s){
    len=(s!=NULL ? strlen(s)+1 : 1);
    str=new char[len];

    if(s!=NULL)
        strcpy(str, s);
}
```

```
string::string(const string& s){
    len=s.len;
    str=new char[len];
    strcpy(str, s.str);
}

string::~string(){
    delete []str;
}
```



3. string

```
string& string::operator=(const string& s){
    delete []str; //
    len=s.len;
    str=new char[len];
    strcpy(str, s.str);
    return *this; //
}
```

```
string string::operator+(const string& s){
    char* tStr=new char[len+s.len-1];
    strcpy(tStr, str); //
    strcat(tStr, s.str); // 가

    string temp(tStr);
    delete []tStr; //
    return temp;
}
```



3. string

```
ostream& operator<<(ostream& os, const string& s){
    os<<s.str;
    return os;
}
```

```
string& string::operator+=(const string& s){
    len=len+s.len-1;
    char* tStr=new char[len];
    strcpy(tStr, str); //
    delete []str;

    strcat(tStr, s.str); // 가
    str=tStr;
    return *this;
}
```

operator + operator=

?



3. string

```
bool string::operator==(const string& s){  
    return strcmp(str, s.str)? false:true;  
}
```

```
istream& operator>>(istream& is, string& s){  
    char str[100];  
    is>>str;  
  
    s=string(str);  
    return is;  
}
```



4. String 가

-
-
-



C++

- C++ 2가
 - (operator >>)
 - getline
- (cin)
- String month
cin >> month; or fsln >> month;
- getline . ->
- getline

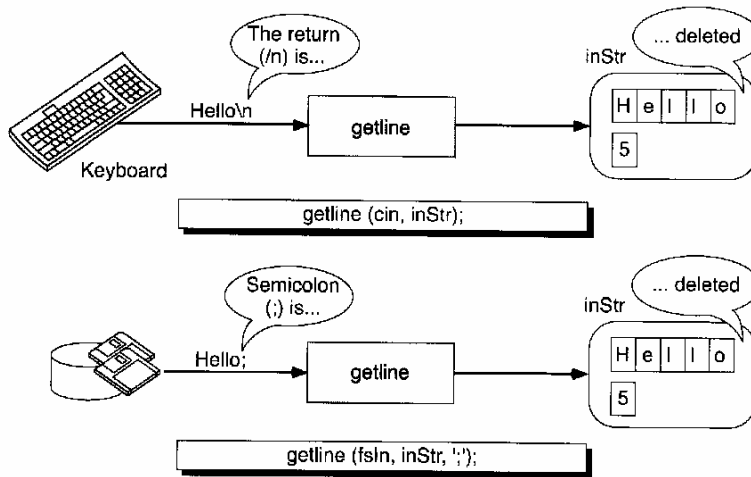


C++

- getline 2~3
 -
 - 가
 - newline ,



C++



C++

```
#include <iostream>
#include <string>
```

```
using namespace std;
```

```
int main()
{
```

```
    string temp;
```

```
    cin >> temp;
    cout << temp;
```

```
    cout << "Enter a name in the form <last, first>: \n";
```

```
    string lastName;
    getline(cin, lastName, ',');
```

```
    string firstName;
    getline(cin, firstName);
```

```
    cout << "Here is your name: \n \t|" << firstName << ' ' << lastName << "|\n";
```

```
    return 0;
```

```
}
```

C++

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

int main()
{
    ofstream fsOut;

    cout << "Begin file copy. Enter your text. \n" << "<EOF> to stop. \n";

    fsOut.open("test.txt");
    if(!fsOut)
    {
        cerr << " Could not open output file. \n";
        exit(100);
    }

    string str;
    while (getline(cin, str) // EOF      getline      , while
           fsOut << str << endl;
           fsOut << str << endl; // Write last line : EOF
           fsOut.close());

    cout << "End file copy " << endl;
    return 0;
}
```



- string upgrade .
- char * string
- char* 가 .




```

#include<iostream>
using std::cout;
using std::endl;

class Person
{
    char *name;
    char *phone;
    int age;

public:
    Person(char* _name, char* _phone, int _age);
    Person(const Person& p);
    ~Person();
    void ShowData();
};

Person::Person(const Person& p)
{
    name=new char[strlen(p.name)+1];
    strcpy(name, p.name);

    phone=new char[strlen(p.phone)+1];
    strcpy(phone, p.phone);

    age=p.age;
}

Person::Person(char* _name, char* _phone, int _age)
{
    name=new char[strlen(_name)+1];
    strcpy(name, _name);

    phone=new char[strlen(_phone)+1];
    strcpy(phone, _phone);

    age=_age;
}

Person::~Person()
{
    delete []name;
    delete []phone;
}

void Person::ShowData()
{
    cout<<"name: "<<name<<endl;
    cout<<"phone: "<<phone<<endl;
    cout<<"age: "<<age<<endl;
}

int main()
{
    Person p1("KIM", "013-333-5555", 22);
    Person p2=p1;

    p1.ShowData();
    p2.ShowData();
    return 0;
}

```



&

Thank You !

