

# 04

객체 포인터와 객체 배열, 객체의 동적 생성



## 학습 목표

1. 객체에 대한 포인터를 선언하고 활용할 수 있다.
2. 객체의 배열을 선언하고 활용할 수 있다.
3. new를 이용하여 동적으로 메모리나 배열을 할당 받고 delete를 이용하여 반환할 수 있다.
4. new를 이용하여 동적으로 객체나 객체 배열을 할당 받고 delete를 이용하여 반환할 수 있다.
5. this 포인터의 개념을 이해하고, 활용할 수 있다.
6. string 클래스를 이용하여 문자열을 다룰 수 있다.

# string 클래스를 이용한 문자열 처리

3

- C++ 문자열
  - ▣ C-스트링 -> char 배열에 저장
  - ▣ C++ string 클래스 객체에 저장
- string 클래스
  - ▣ C++ 표준 라이브러리, <string> 헤더 파일에 선언
  - ▣ 가변 크기의 문자열 -> 사용자가 입력한 문자열 크기에 맞추어 메모리 크기 자동조절
  - ▣ 다양한 문자열 연산을 실행하는 연산자와 멤버 함수 포함
    - 문자열 길이, 문자열 복사, 문자열 비교, 문자열 연결 등

# <string> 헤더파일 포함

4

- string 클래스
  - ▣ C++ 표준 라이브러리, <string> 헤더 파일 포함해야 함
  - ▣ 이름공간 std 안에 클래스가 정의되어 있음

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

# string 객체 생성

5

## □ string 클래스 생성자 (표4-1 주요 생성자)

생성자	설명
string();	빈 문자열을 가진 string 객체생성
string( <b>string</b> & str);	str을 복사한 새로운 string 객체를 생성
string( <b>char</b> * str);	문자배열 str을 복사하여 string 객체 생성
string( <b>char</b> * str, <b>int</b> n);	문자배열 str에서 n개의 문자를 복사하여 객체 생성

# string 객체 생성

6

## □ string 객체의 생성 예

```
// 빈 문자열을 가진 스트링 객체  
string str;
```

```
// string객체 address를 복사한 copyAddress 객체 생성  
string copyAddress(address);
```

```
// 문자열 상수로 초기화  
string address("서울시 성북구 삼선동 389");
```

```
// C-스트링으로부터 string 객체 생성  
char text[] = "Love C++";  
string title(text);
```

```
// C문자열로 직접 초기화  
string title = "Love C++";
```

# string 객체 생성

7

- string 객체의 동적 생성

```
string* p = new string("C++");
```

...

```
delete p;
```

# string 객체 입출력

8

- string 객체 출력: cout과 << 연산자

```
string address("서울시 성북구 삼선동 389");
string title("Love C++");
cout << address << endl;           // "서울시 성북구 삼선동 389" 출력
cout << title << endl;             // "Love C++" 출력
```

- string 객체 입력 : cin과 >> 연산자

```
string name;
cin >> name;                      // 공백이 입력되면 공백 이전까지만 저장
```

```
string name;
getline(cin, name, '\n'); // 엔터키('\n') 이전까지 모두 저장
```

# 예제4-11 string클래스를 이용 문자열 생성 및 출력

9

```
#include <iostream>
#include <string>
using namespace std;
int main() {
    string str;
    string address("서울시 성북구 삼선동 389");
    string copyAddress(address);
    char text[] = { 'L', 'o', 'v', 'e', ' ', 'C', '+', '+', 'W0' };
    string title(text);
    cout << str << endl;
    cout << address << endl;
    cout << copyAddress << endl;
    cout << title << endl;
}
```

서울시 성북구 삼선동 389  
서울시 성북구 삼선동 389  
Love C++

# 예제 string클래스를 이용 문자열 입력

10

```
#include <iostream>
#include <string>
using namespace std;
int main() {
    string name, address;

    cout << "이름:";
    getline(cin, name, '\n');

    cout << "주소:";
    getline(cin, address, '\n');

    cout << name << endl;
    cout << address << endl;
    return 0;
}
```

이름 : 홍길동 <엔터>  
주소 : 군산시 대학로 558 <엔터>  
홍길동  
군산시 대학로 558

# string 클래스의 주요 멤버함수 및 연산자

11

- 멤버함수 : 표4-2 참조
- 연산자 : 표4-3 참조

```
string str("Hello");
```

```
"Hello"  
....  
string();  
string(char* s);  
int length();  
char& at(int pos);  
....
```

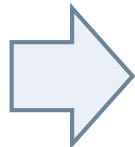
str

# 문자열 길이 계산

12

- string 클래스의 멤버 함수 length() 사용

```
char s[10] = "hello";
printf("%d\n", strlen(s));
```



```
string str("hello");
cout << str.length() << endl;
```

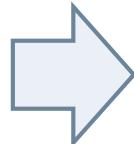
# 문자열 복사

13

- string 객체는 대입(=) 연산 가능

```
char str1[10] = "hello";
char str2[10];

strcpy(str2, str1);
printf("%s\n", str2);
//str2 = str1; //에러
```



```
string str1("hello");
string str2;

str2 = str1;
cout << str2 << endl;
```

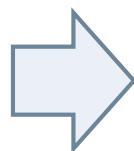
# 문자열 연결

14

- string 객체 연결은 + 연산자 사용

```
char str1[20] = "hello";
char str2[20] = "world";
```

```
strcat(str1, str2);
printf("%s\n", str1);
```



```
string str1("hello");
string str2("world");
```

```
str1 = str1 + str2;
cout << str1 << endl;
```

# 문자열 비교

15

- string 객체 비교는 ==, !=, <, <=, >, >= 연산자 사용

```
char str1[20] = "hello";
char str2[20] = "world";
if (strcmp(str1, str2) == 0) printf("같은 문자열\n");
```



```
string str1("hello");
string str2("world");
if (str1 == str2) cout << "같은 문자열" << endl;
if (str1 != str2) cout << "다른 문자열" << endl;
if (str1 < str2) cout << "str1이 먼저 나옴" << endl;
if (str1 > str2) cout << "str2이 먼저 나옴" << endl;
```

# 문자열의 각문자 다루기

16

- `at()` 멤버함수와 `[ ]`연산자는 문자열의 특정 위치에 있는 문자를 리턴

```
char str[20] = "hello";
str[0] = 'H';                                // "Hello"
```

```
string str("I love C++");
char ch1 = str.at(7);                          // ch1 = 'C'
char ch2 = str[7];                            // ch2 = 'C'
str[7] = 'D';                                // "I love D++"
```

# 예제4-12 string 배열 선언과 문자열 키입력 응용

17

```
#include <iostream>
#include <string>
using namespace std;
int main() {
    string names[5];
    for (int i = 0; i < 5; i++) {
        cout << "이름 >> ";
        getline(cin, names[i], '\n');
    }
    string latter = names[0];
    for (int i = 1; i < 5; i++)
        if (latter < names[i]) latter = names[i];

    cout << "사전에서 가장 뒤에 나오는 문자열은 " << latter << endl;
}
```

이름 >> Kim Nam Yun<엔터>  
이름 >> Chang Jae Young<엔터>  
이름 >> Lee Jae Moon<엔터>  
이름 >> Han Won Sun<엔터>  
이름 >> Hwang Su hee<엔터>

사전에서 가장 뒤에 나오는 문자열은 Lee Jae Moon

# 예제 : 문자열 처리 응용

18

```
#include <iostream>
#include <string>
using namespace std;
int main() {
    string str1, str2, result;
    cout << "문자열1" >> " ";
    getline(cin, str1, '\n');
    cout << "문자열2" >> " ";
    getline(cin, str2, '\n');

    cout << "문자열1의 길이: " << str1.length() << endl;
    cout << "문자열2의 길이: " << str2.length() << endl;
    cout << "연결한 문자열: " << str1+str2 << endl;

    if (str1 <= str2) result = str1;
    else result = str2;
    cout << "사전에서 앞에 나오는 문자열은" << result << endl;
}
```

문자열1 >> Hello<엔터>  
문자열2 >> World<엔터>  
문자열1의 길이 : 5  
문자열2의 길이 : 5  
연결한 문자열 : HelloWorld  
사전에서 앞에 나오는 문자열은 Hello

# string 기타기능

19

함수명	기능 설명
<code>string::compare</code>	문자열 비교 -> 관계연산자
<code>string::append</code>	문자열 연결 -> + 연산자
<code>string::insert</code>	문자열 삽입, "I love C++" -> "I really love C++"
<code>string::erase</code>	문자열 삭제, "I love C++" -> "I C++"
<code>string::substr</code>	문자열의 일부를 추출, "I love C++" -> "love"
<code>string::find</code>	문자열에서 특정 문자나 문자열을 검색
<code>std::stoi</code>	string 객체를 숫자로 변환, "100" -> 100
<code>std::to_string</code>	숫자를 string 객체로 변환, 3.14 -> "3.14"
<code>std::isdigit</code>	문자가 숫자인지 확인
<code>std::isalpha</code>	문자가 알파벳인지 확인
<code>std::toupper</code>	문자열 대문자로 변환, 'a' -> 'A'
<code>std::tolower</code>	문자열 소문자로 변환, 'A' -> 'a'

# string 기타기능

20

- C++ 표준 라이브러리의 클래스 또는 함수를 검색하는 방법
- <https://www.cppreference.com/> 접속
- 우측 상단 검색창에 키워드 입력

[cppreference.com](https://www.cppreference.com) Create account  X Search

[Page](#) [Discussion](#) Standard revision: [Diff](#) ▼ [View](#) [View source](#) [History](#)

## C++ reference

C++11, C++14, C++17, C++20, C++23, C++26 | Compiler support C++11, C++14, C++17, C++20, C++23, C++26

**Language**

- Keywords – Preprocessor
- ASCII chart
- Basic concepts
- Comments
- Names (lookup)
- Types (fundamental types)
- The main function
- Expressions
- Value categories
- Evaluation order
- Operators (precedence)
- Conversions – Literals
- Statements
  - if – switch
  - for – range-for (c++11)
  - while – do-while
- Declarations – Initialization
- Functions – Overloading
- Classes (unions)

**Memory management library**

- unique\_ptr (C++11)
- shared\_ptr (C++11)
- weak\_ptr (C++11)
- Memory resources (C++17)
- Allocators – Low level management

**Metaprogramming library (C++11)**

- Type traits – ratio
- integer\_sequence (C++14)

**General utilities library**

- Function objects – hash (C++11)
- Swap – Type operations (C++11)
- Integer comparison (C++20)
- pair – tuple (C++11)
- optional (C++17)
- expected (C++23)
- variant (C++17) – any (C++17)
- String conversions (C++17)
- Formatting (C++11)

**Iterators library**

**Ranges library (C++20)**

**Algorithms library**

- Execution policies (C++17)
- Constrained algorithms (C++20)

**Numerics library**

- Common math functions
- Mathematical special functions (C++17)
- Mathematical constants (C++20)
- Basic linear algebra algorithms (C++26)
- Numeric algorithms
- Pseudo-random number generation
- Floating-point environment (C++11)
- complex – valarray

**Date and time library**

- Calendar (C++20) – Time zone (C++20)

**Localization library**

- locale – Character classification

# 실습과제1

21

- string 객체는 문자열 입력 받을 때 몇 자를 입력할지 물어 보지 않고 알아서 크기를 자동 조절한다. 동작 원리를 검색해보고 설명하라.
- C++에서는 char배열을 사용하지 말고 string 클래스를 사용할 것

# 실습과제2

22

- 키보드로부터 문자열을 입력 받아 화면에 출력해주는 프로그램을 작성하시오. "quit"를 입력하면 종료한다. 단, string 클래스의 연산자와 멤버함수 만을 이용하여 작성하라.

문자열 입력 : hello<엔터>

입력된 문자열: hello

문자열 입력 : world<엔터>

입력된 문자열: world

문자열 입력 : C++<엔터>

입력된 문자열: C++

문자열 입력 : quit<엔터>

# 실습과제3

23

- 키보드로부터 문자열을 입력 받아 앞에서 받은 문자열에 이어서 화면에 출력해주는 프로그램을 작성하시오. “quit”를 입력하면 종료한다. 단, string 클래스의 연산자와 멤버함수만을 이용하여 작성하라.

문자열 입력 : hello<엔터>

이어진 문자열: hello

문자열 입력 : world<엔터>

이어진 문자열: hello world

문자열 입력 : C++<엔터>

이어진 문자열: hello world C++

문자열 입력 : quit<엔터>

# 실습과제4

24

- 키보드로부터 문자열을 입력 받아 문자열에 포함된 문자 'a'의 개수를 구하여 출력해주는 프로그램을 작성하시오. 단, string 클래스의 연산자와 멤버함수만을 이용하여 작성하라.

문자열 입력 : *Are you happy? I am so happy.*<엔터>  
문자 a는 3개 있습니다.

## 실습문제5

25

- 210~218페이지 문제 중에서 1, 2, 3(1), 4, 6, 8번 중에서 3문제를 풀어서 제출하시오.

# 과제제출방법

26

- 소스코드, 라인단위의 주석, 실행결과를 포함하는 pdf파일을 작성한 후 eclass 과제 게시판에 업로드, **반드시 하나의 pdf파일로 업로드할 것**
- 기한 : 과제 게시판에 마감시간 참조
- 실행결과를 캡쳐할 때 글자를 알아보기 쉽게 확대해서 캡쳐할 것.
- 소스코드의 첫 부분은 아래처럼 제목, 날짜, 작성자(학번, 이름)를 작성할 것

```
// *****
// 제 목 : 정수 4개의 평균을 구하는 프로그램
// 날 짜 : 2023년 9월 10일
// 작성자 : 15010101 홍길동
// *****

// 소스코드 작성
```