**SWING 307I STUDY** 

# 2022년 C++1주차



김민성 SWING 29 (담당자)

박규리 SWING 29

이새나 SWING 29

#### C++ 1주차



C++ 개념

C++ 객체지향

C++ 개발과정

컴파일 방법



C++ 프로그래밍 기본

기본 요소 입출력 NAMESPACE & STD:: cstring



연산자

연산자

C++ 1주차

PART 01

# 

C++ 객체지향

C++ 개발과정

컴파일 방법

◎ 설계 목표

C언어와의 호환성 유지 객체 지향 타입 체크 엄격히 → 실행 시간 오류 가능성 최소화 효율성 저하 최소화

- ◎ C 언어에 추가된 기능
- ① 함수 중복 | ② 디폴트 매개 변수 | ③ 참조 | ④ 참조에 의한 호출 |
- ⑤ NEEW, DELETE 연산자 | ⑥ 연산자 재정의 | ⑦ 제네릭 함수와 클래스 |

# 절차 지향 vs 객체 지향

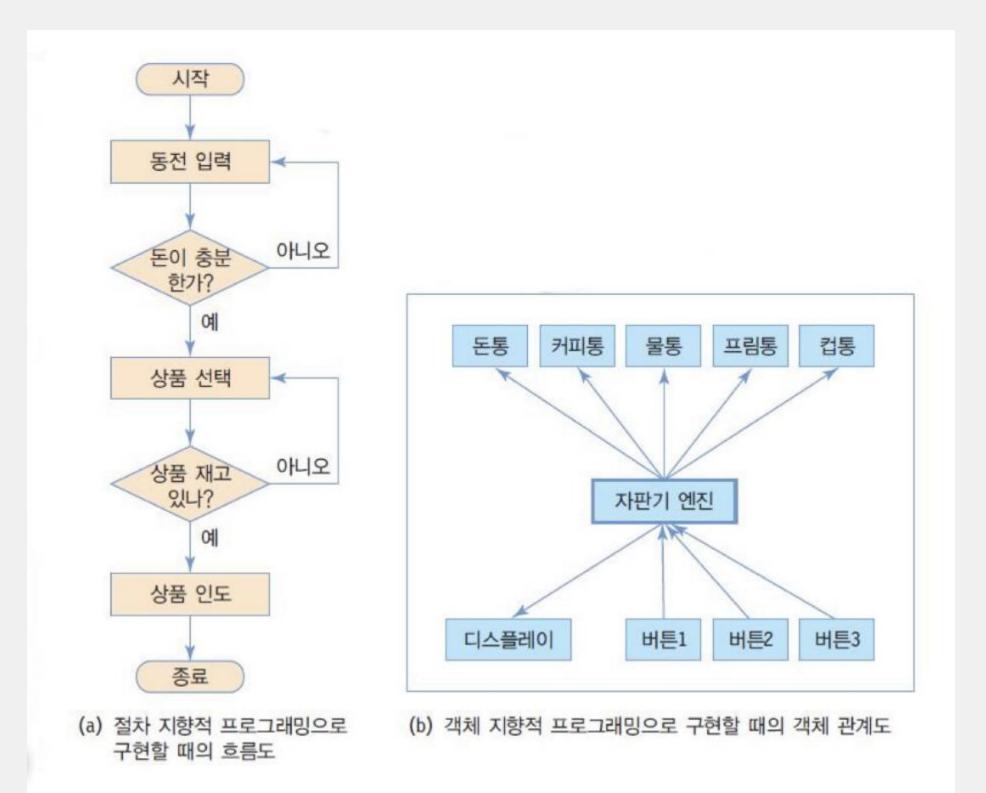
F

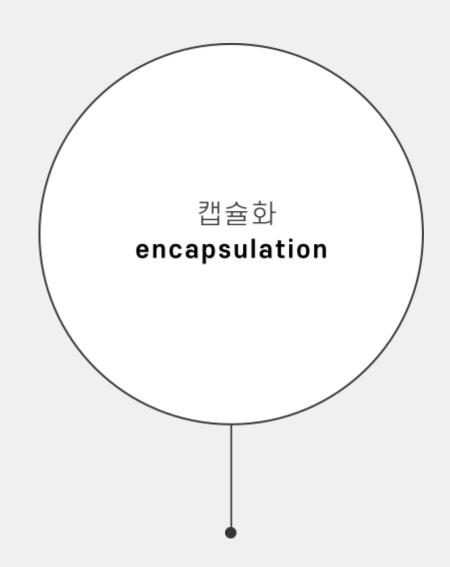
## ◎ 절차지향

- : 순자척으로 명령어를 나열하여 프로그래밍 (순서, 흐름 중심)
- \*절차 지향 언어: C

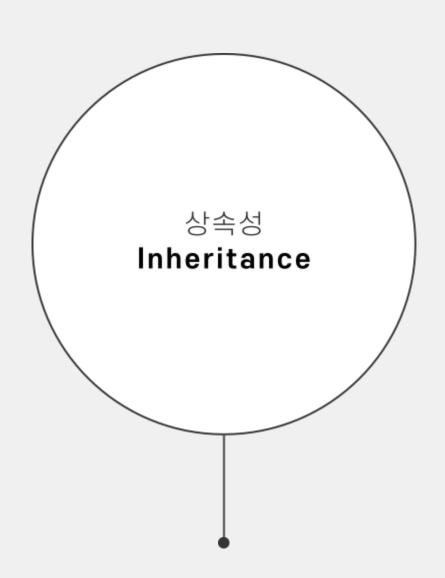
## ◎ 객체지향

- : 객체들의 관계를 결정하고 이들의 상호 작용을 멤버 함수와 멤버 변수로 구현
- \*객체 지향 언어: C++, PYTHON, JAVA, C#
- \*중복 연산 방지, 모듈 재활용으로 하드웨어 처리양 줄여준다

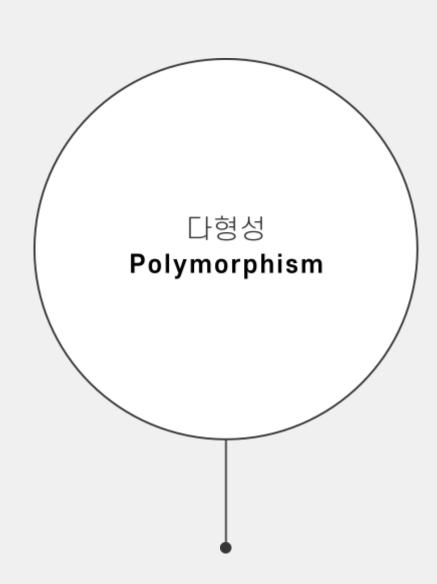




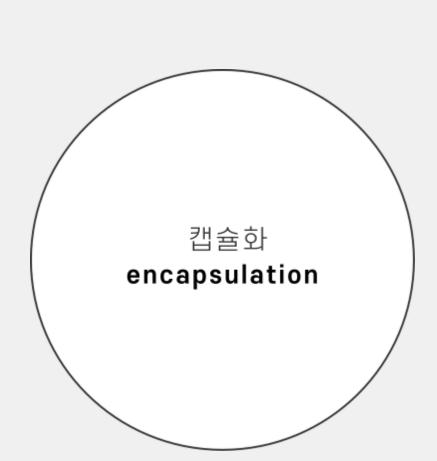
데이터를 캡슐로 싸서 외부의 접근으로부터 데이터를 보호 +) C++에서 클래스(CLASS 키워드)로 캡슐 표현



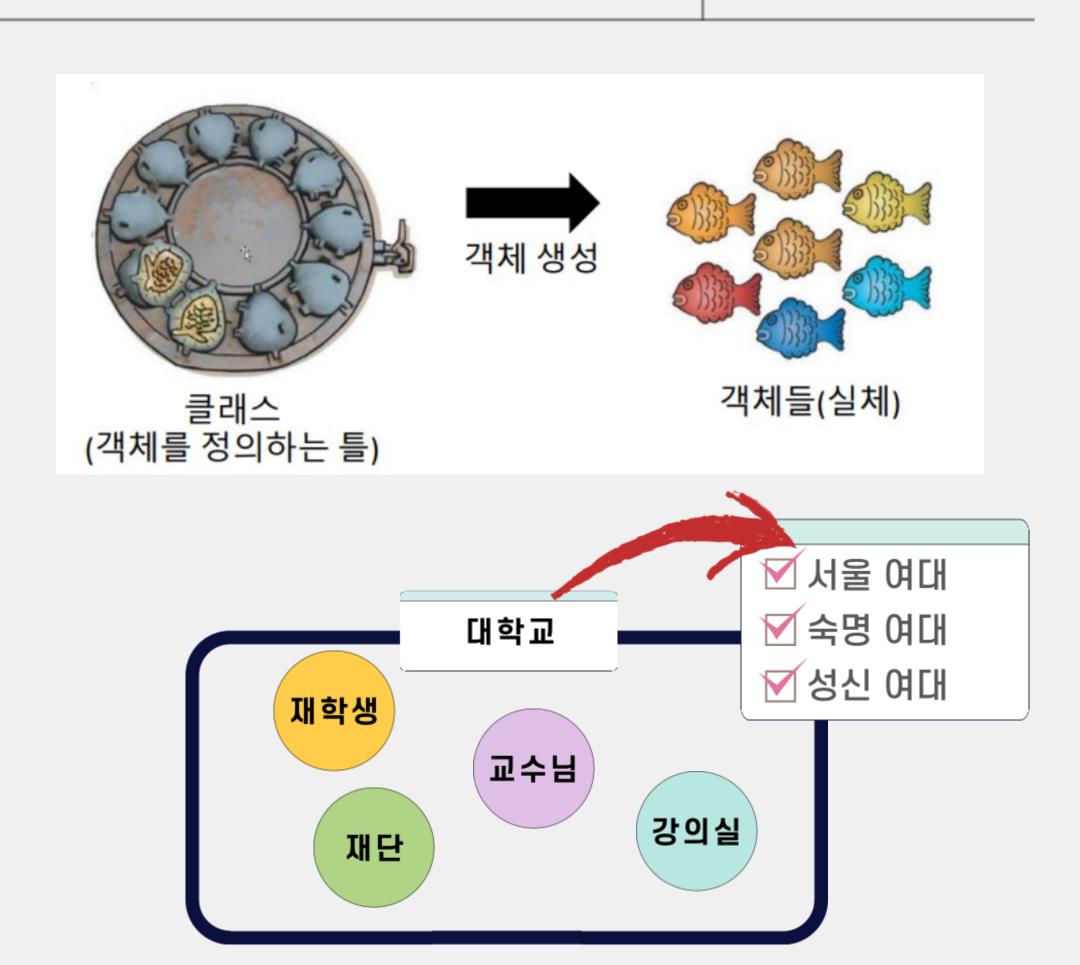
자식이 부모의 유전자를 물려받는다. 자식클래스 = 자식클래스 + 부모클래스 코드의 재사용성을 높여 소프트웨어 생산성을 높인다.

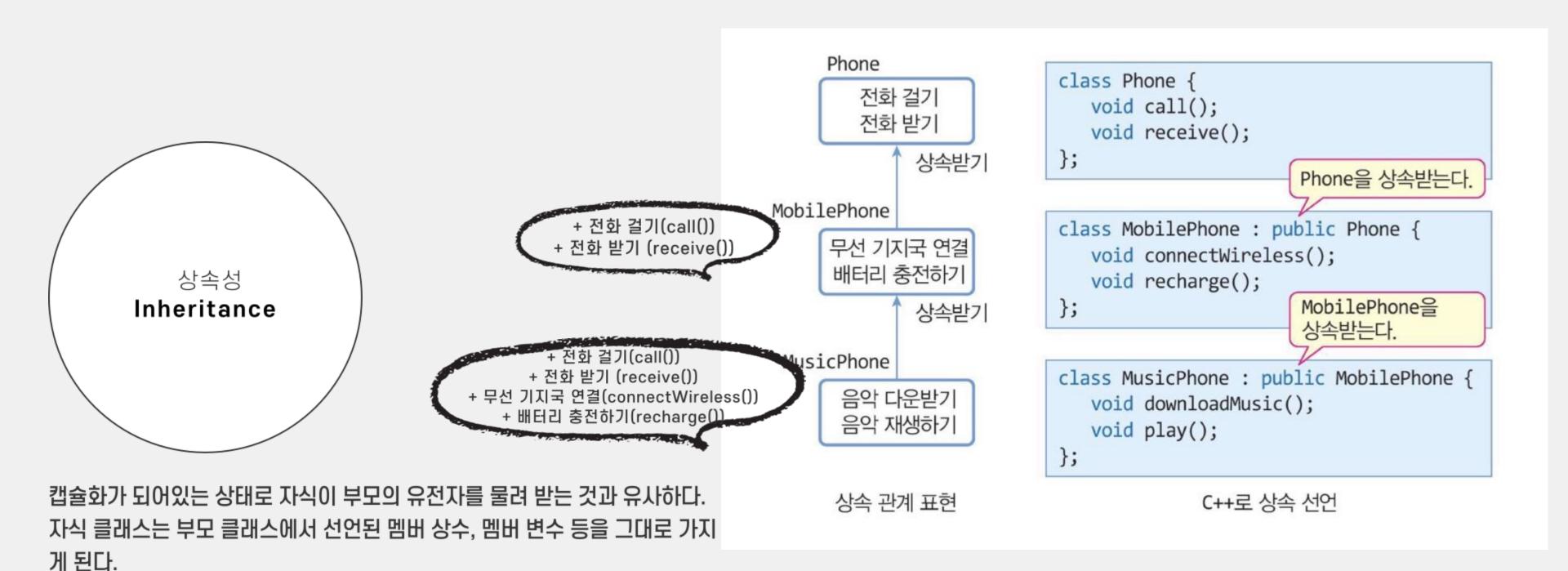


하나의 기능에 경우에 따라 다르게 보이거나 작동하는 현상 +) 연산자 중복, 함수 중복, 함수 재정의



데이터를 캡슐로 싸서 외부의 접근으로부터 데이터를 보호하는 성질 C++에서 클래스(CLASS 키워드)로 캡슐 표현된다.





코드 재사용성을 높여 소프트웨어 생산성을 높인다.





# 연산자 중복

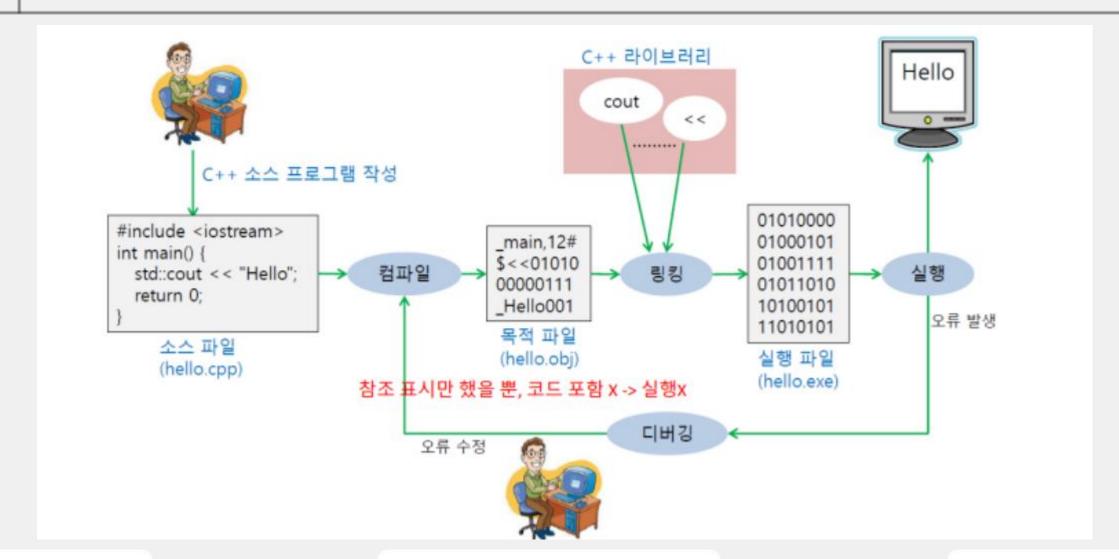
덧셈 문자열 연결 객체 더하기

# 함수 중복

함수이름 같을 때 (c에서는 오류) 매개 변수의 개수나 타입을 다양하게 하는 것 2 + 3 → 5 "swing" + "30th" → "swing30th" redColor 객체 + blueColor 객체→purpleColor 객체

void add(int a, int b){···}
void add(int a, int b, int c){···}
void add(float a, double b){···}

하나의 기능에 경우에 따라 다르게 보이거나 작동하는 현상 연산자 중복, 함수 중복, 함수 재정의(OVERRIDING)





- C++ 소스 프로그램을 기계어를 가진 목적 파일로 변환
- 문법에 맞지 않는다면 컴파일 오류 발생
- 컴파일 후 기계어로 구성된 obj파일 생성

- 목적 파일끼리 합쳐 실행 파일을 만 드는 과정
- 목적 파일과 C++ 표준 라이브러리의 함수 연결

- 프로그램 내에 오류 위치 발견
- 문제의 원인을 찾아 수정하는 과정





C++ 1주차

기본 요소

입출력

NAMESPACE & STD::

cstring

# 주석문

프로그램 실행에 영향을 주지 않으며, 프로그램 에 대한 설명 및 특이사항 메모하는 역할이다.

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

/* ~ */

pint main() {

/*cout << "Hello" << endl;
cout << "welcome" << endl;
cout << "swing" << endl;
the cout << "swing" << endl;
cout << pre>
cout 

cout 

cout
```

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

//
int main() {

//cout << "Hello" << endl;
cout << "welcome" << endl;
cout << "swing" << endl;
welcome
swing
return 0;
```

# main()

- C++ 프로그램의 실행 시작점
- 반드시 하나의 main()함수가 존재해야 함
- main함수 종료되면 프로그램 종료

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

Evoid swing() {
    cout << "Wellcome !!" << endl;
    cout << "함수 종료" << endl;
}

Eint main() // C++ 표준 리턴 타입은 int
    cout << "Hello SWING 30th" << endl;
    swing();
    cout << "main 함수 종료" << endl;

return 0;

Hello SWING 30th
Wellcome !!
함수 종료
main 함수 종료
main 함수 종료
```

main() 함수에서는 예외적으로 return 문이 생략 가능하다. 생략할 경우에 자동으로 return 0; 실행된다.

# #include <헤더>

\*전처리기에 내리는 지시문으로 C++ 코드를 읽기 전에 해더 파일을 읽어 소스 코드에 삽입 하도록 지시하는 역할이다.

\* 전처리기

: 프로그램을 컴파일할 때 컴파일 직전에 실행 되는 별도의 프로그램이다.

#### #include <iostream>

C++ 표준 입출력을 위한 클래스와 객체가 선언되어 있어 키보드 입력, 화면 출력시에 필요하다.

EX) cout, cin, >> etc..

```
#include <iostream>
 using namespace std;
⊟void swing() {
     cout << "Wellcome !!" << endl;
     cout << "함수 종료" << endl;
□int main() // C++ 표준 리턴 타입은 int
     cout << "Hello SWING 30th" << endl;
     swing();
     cout << "main 함수 종료" << endl;
     return 0;
```



# cout & << 출력

cout은 데이터를 콘솔 화면에 출력할 때 사용하는 명령어로 << 뒤에 작성한 피연산자를 출력한다.

#### endl

다음 줄로 넘어갈때, '₩n' 대신 << 뒤에 std:: endl 로 사용해도 된다. ※ ₩n 과 endl은 동일하지 않다.

```
pint main() {
    int num = 21;
    char c = '#';
    std::cout << "Hello" << std::endl;
    std::cout << "Hello#n" << "안녕#n"; Hello
    std::cout << "안녕 내 나이는" << numbello
    안녕 내 나이는 21이야.
```

```
std::cout << "Hello" << std::endl;
std::cout << "Hello\n" << "만녕\n";
```

# cin & >> 입력

키보드와 C++ 응용프로그램을 연결하는 C++ 입력 스트림 객체

>>

: 입력 스트림에서 데이터를 읽어 오른쪽 피연산자 에 지정된 변수 삽입

#### 특징

- 01. 입력 버퍼를 내장하고 있다.
- 02. <enter> 키가 입력될 때까지 입력된 키는 입력 버퍼에 저장된다.
- 03. <enter> 키가 입력되면 입력 버퍼에 있는 키 값을 읽어 변수에 전달되며 입력이 종료된다.

```
int main() {
   int month;
   std::cout << "지금은 몇월인가요?" << std::endl;
   std::cin >> month;
   std::cout << month << "월" << std::endl;
```

```
지금은 몇월인가요?

지금은 몇월인가요?
지금은 몇월인가요?
3 입력 후 <enter>
지금은 몇월인가요?
3 기금은 몇월인가요?
3 기금은 몇월인가요?
```

# namespace

#### 이름 충돌 방지

이름 공간을 생성하여 안에 선언된 이다른 이름 공간과 별도로 구분한다.

ex) 1호::SWU | 2호::SWU

→ 1호와 2호는 이름공간

#### std::

C++ 표준에서 정의한 이름 공간 중 하나

# using

std:: 생략하는 경우에 using 지시어를 사용한다.

```
#include<iostream>
pint main() {
    using std::cout; //cout에 대해서만 std:: 생략
    cout << "Hello" << std::endl; //std::cout에서만 std:: 생략
```

```
#include<iostream>
using namespace std; // std 이름 공간에 선언된 모든 이름에 std:: 생략
pint main() {

cout << "Hello" <<endl;
}
```

# using namespce std;

<iostream> 헤더 파일에 선언된 이름을 사용할 때 std:: 생략

**QUIZ.01** 

#### Q. 다음 중 오류가 나는 코드는?

01.

```
#include<iostream>
⊒int main(){
    std::cout << "SWING 307|" <<std::endl:
```

02.

```
#include<iostream>
 using namespace std;
⊟int main(){
     std::cout << "SWING 307|" <<std::endl:
```

03.

```
#include<iostream>
⊒int main(){
       cout << "SWING 307|" << endl;
```

04.

```
#include<iostream>
using std::cout;
gint main() {
    cout << "SWING 307|" <<std::endl:
```



# cstring

#### 01. ₩0 으로 으로 끝나는 문자 배열

<cstring> 또는 <string.h>
C라이브러리 함수 사용
strcpy(), strcmp(), strlen() 등

배열의 크기를 6으로 자동 설정 남은 원소들은 '₩0'으로 처리

#### 02. string 클래스 이용

<string> 헤더 파일에 선언된다.

문자열의 크기 제약 없다.

C++ 표준 라이브러리에서 제공하는 클래스, 문자열을 객체로 다룬다.

```
char name1[6] = { 'S','₩','I','N','G','₩0' }; // ₩0 널 문자로 끝나는 배열로 문자열 함수 사용 가능하다.
char name2[6] = { 'S','₩','I','N','G' }; // 단순 문자 배열로 문자열 관련 함수 사용 불가능하다.
```

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

Dint main() // C++ 표준 리턴 타입은 int

{
    char name3[6] = "SWING";
    char name4[] = "SWING";
    char name5[10] = "SWING";

    cout << "name3의 배열 크기: " << sizeof name3 << endl;
    cout << "name4의 배열 크기: " << sizeof name4 << endl;
    cout << "name5의 배열 크기: " << sizeof name5 << endl;
    return 0;

    return 0;
```

name3의 배열 크기: 6 name4의 배열 크기: 6 name5의 배열 크기: 1(



#### cin & >>

>> 연산자는 공백문자를 만나게 되면 이전의 입력된 문자들은 모두 하나의 문자열로 인식한 다.

# cin.getline()

F

공백 포함한 문자열 입력 받을 수 있다.

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

Dint main() {

cout << "이름을 입력하시오 : #n";
char name[12];
cin >> name;
cout << "이름은 " << name << "입니다.#n";
}
```

이름을 입력하시오 : 김 민성 이름은 김입니다.

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

Eint main() {

cout << "이름을 입력하시오 : \n";
char nameO[12];
cin.getline(nameO, 12, '\n");
cout << "이름은 " << nameO << "입니다.\n"; 김 민성
이름은 김 민성입니다.
```

C++ 1주차

PART 03

연산자

연산자

F

01

산술 연산자

산술 연산자	기호
덧셈	+
밸셈	-
곱셈	*
나눗셈	/
나머지	%



대입 연산자

복합 대립 연산자	의미
x += y	x= x + y
x -= y	x= x - y
x *= y	x= x * y
x /= y	x= x / y
x %= y	x= x % y



관계 연산자

관계 연산자	의미
x == y	x와 y가 같은가
x != y	x와 y가 다른가
x > y	x가 y보다 큰가
x < y	y가 x보다 큰가
x >= y	x가 y보다 크거나 같은가
x <= y	x가 y보다 작거나 같은가

참 =1 / 거짓 = 0



논리 연산자



증감 연산자

논리 연산자	의미
x && y	AND 연산(x,y 모두 참일 때만 참)
x II y	OR 연산(x, y중 하나만 참이어도 참)
!x	NOT 연산(x가 참이면 거짓, 거짓이면 참)

증감 연산자	의미
y = x++	y에 x값 대입 후 x값 증가
y = ++x (x = x+1)	x값 증가 후 y에 대입
y = x	y에 x값 대입 후 x값 감소
y =x (x = x-1)	x값 감소 후 y에 대입

참 =1 / 거짓 = 0

**QUIZ.02** 

#### Q. 아래 코드의 출력값은?

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

Eint main() {

int a = 2;
int b = 7;
int c = 4;
int d = 2;

int x = 30;

cout << ((a >= d) && (b != c)) << endl; // 1년

cout << ++x << endl;
cout << x << endl;
cout << x << endl;
// 2년
```

1번, 2번 답 보내주세요.(5min)

C++ 1주차

PART J J J

# 과제01

## 음료 가격 및 금액 입력 받기

#### < 조건 >

- 1. 이름 입력하고 출력하기 (단, 띄어쓰기 할 것 ex. 김 민성)
  - 2. 배운 내용 참고하여 음료 가격과 개수 입력 받기
    - 3. 500원 할인해주기
    - 4. 총 금액 출력하기

Microsoft Visual Studio 디버그 콘솔

# 당신의 이름은 무엇입니까? 김 민성 반갑습니다.김 민성님 -----SWING 30------음료의 가격 입력을 시작합니다. 콜라의 가격: 1200 삼다수의 가격: 700 환타의 가격: 1300 개수를 입력해주세요. 콜라 수량: 3 삼다수 수량: 2 환타 수량: 1

## 과제02

# <cstring> 주요 함수 strcmp(), strlen(), strcpy() 및 string 클래스 주요 멤버 함수 append, insert, length, clear 등 문서화

함수명, 사용법, 실습(코드, 실행창) 문서화하는 과제로 최소한 아래 예시처럼 하기

F

```
    <string>

    string str1 = "lamHungry";

    str1.at();

    1. at()

    Index에 숫자(n)가 들어가며, n에 해당하는 문자를 반환한다.

    Index가 string의 범위를 벗어나면 예외를 뱉는다.

    2. [n]

    Index에 숫자(n)가 들어가며, n에 해당하는 문자를 반환한다.

    Index 범위를 검사하지 않아 at 함수보다 빠르지만 예외를 뱉지 않는다.

    3. front()

    string의 맨 앞 인자를 반환한다.

    4. back()

    string의 맨 뒤 인자를 반환한다.
```

```
#include<iostream>
#include <string>
using namespace std;
int main() {
    string str1 = "Welcome";
    string str2 = "SWING 30th";
    cout << "str1: " << str1 << endl;
    cout << "str2: " << str2 << endl;
    cout << endl;
    cout << "01. at()" << endl;
    cout << "str1.at(3): " << str1.at(3) << endl;
    cout << endl;
   cout << "02, '[]'" << endl;
cout << "str1[3]:" << str1[3] << endl;
    cout << endl;
    cout << "03. front()" << endl;
    cout << "str2.front():" << str1.front()<< endl;
    cout << endl;
    cout << "04. back()" << endl;
    cout << "str2.back(): " << str2.back() << endl;
```

```
str1:Welcome
str2:SWING 30th

01. at()
str1.at(3):c

02. '[]'
str1[3]:c

03. front()
str2.front():S

04. back()
str2.back():h
```

# 제출방법

01. 코드 작성 과제 github 립로드 후 tistory에 링크 첨부 후 업로드 주석부분에 코드에 대한 설명 자세히 적기

02. 문서화 과제 tistory 업로드 비공개 설정하여 비번은 선배 카톡으로 개인적으로 알려주기

# 카페에 댓글로 tistory 링크 올리기

참고 [2021\_SWING]C++1주차,이은경 2021\_2 C++ 최주영 교수님

#### SWING 30 study

# SWING 30기 C++ 스터디

F

제출기한: | 질문 사항 (카톡)

29기 | 김민성, 이새나, 박규리