



<3주차> 정리노트!

15조 (백승원, 최진형)

주석문

[?] 개발자가 자유롭게 붙인 특이 사항의 메모, 프로그램에 대한 설명 [?] 프로그램의 실행에 영향을 미치지 않음

main() 함수

#include <iostream>

cout << 연산자 이용

(C++ 프로그램 실행 시작 함수

전체 프로그램에게 보내는 메시지

(std::cout << "Hello\n"; // 화면에 Hello를 출력하고 다음 줄로 넘어감
std::cout << "첫 번째 맛보기입니다.";

main 함수의 C++ 표준 모양

main에서 return을 생략 가능

cout 객체

namespace

이름 충돌을 방지하는 경우

스크린 출력 장치에 연결된 표준 C++

출력 스트림 객체

키워드

(이름 충돌 해결

std 이름공간에 std::cout 사용
선언

(개발자가 사기함의 이름 공간을 생성할 수 있도록 함.

Cin - 입력버퍼 저장

Enter 입력될 때까지 입력된 것을 입력버퍼에 저장

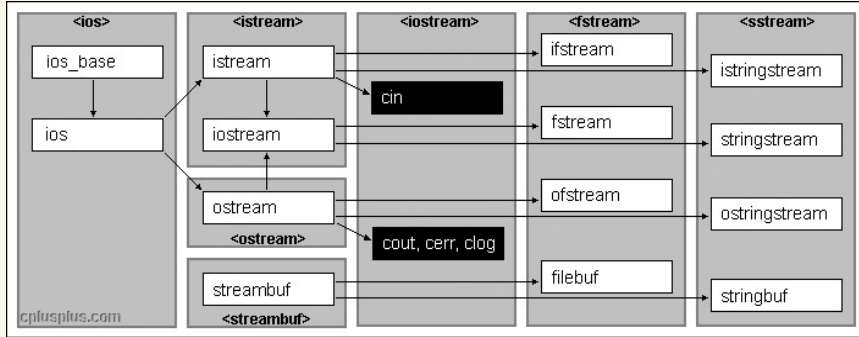
C++ 변수 선언

↓
이후 코딩 가능

장점 : 변수를 신중하게 적는 선언

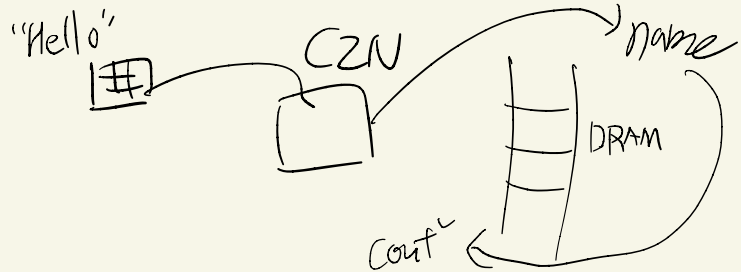
→ 변수 이름에 대한 타입 오류 ↓

Input output



C++ 문자열 표현 방식 2가지
'w/o'으로 끝나는 문자열

char name[6] = {'G', 'r', 'a', 'c', 'e', '\0'} // name은 문자열 "Grace"



```
char name5[10] = "Grace";
```

name5[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
'G'	'r'	'a'	'c'	'e'	'\0'	'\0'	'\0'	'\0'	'\0'

"Grace" 문자열 '\0'로 초기화

string class 이용

↳ 다양한 함수 제공, 문자열 비교, 복사, 수정 등
가능

string class → C++보다 다루기 쉽다

↳ C++에서 강력 추천

C++ 표준 클래스

문자열 크기 다른 제약 없음

객체 지향적

#include <헤더파일>

'헤더파일' 찾기 위치

컴파일러가 설치된 폴더에서 찾아보기

#include '헤더파일'

“
개발자의 프로젝트 폴더나 개발자가 컴파일 옵션으로
지정한 include 폴더에서 찾아보기 !

<연습문제 풀이>

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

```
int main (){
    char users[100];
    char longName[100];
    int a=0;
    cout << "5명의 이름을 ; 로 하여 입력하세요>> ";
```

```
    int cnt =1;
    while(1){
        cin.getline(users, 100, ';');
        cout << cnt << " : " << users << endl;
        cnt++;
        if(a<strlen(users)){
            a=strlen(users);
            strcpy(longName, users);
        }
```

```
        if(cnt==6){
            break;
        }
```

```
    }
    cout<< "가장 긴 이름은 " << longName << endl;
}
```

과정중 막히거나 어려웠던 점

↳ getline 으로 줄리명을 받아오는것까지는 구현,
but j로 구분하고 싶었는데 첫번째 이름만 계속 받아오는
현상 발생

두번째 세번째 이름에 접근하는 방법을 문자열

↳ 항상 귀찮음!!!