**190402**

**\*부동소수점수**

#include <iomanip> 설명

**i/o manip**ulators 매개 변수를 조작하도록 해주는 헤더

std::setprecision(6) //Default, 소수점 여섯째자리까지 출력한다. (디폴트 값 수정 가능)

실수부나 가수부의 정밀도를 높일수록 오차가 생기는 예제

예제 (1)

float f(123456789.0f);

cout<<std::setprecision(9)<<endl;

cout<<f<<endl;

//123456789.0이 아닌 123456792가 나옴.

예제 (2)

double d(0.1);

cout<<std::setprecision(17)<<endl;

cout<<d<<endl;

//0.100000000... 이 나와야 정상이지만

//0.1000000000000001 이 나온다.

예제 (3)

double d1(1.0);

double d2(0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1);

cout << std::setprecision(17) << endl;

cout << d1 << endl << d2 << endl;

//1.0 1.0 이 나와야 하지만 1, 0.99999999999999989 가 나온다.

Q. 왜 오차가 생기는 걸까?

* 나도 모름;

**\*수 가 아닌 수**

<용어 정의>

Inf : (Positive) Infinite 무한대

-inf : Negative Infinite 음의 무한대

Nan : Not A Number 수가 아님

Ind : Indeterminate 결정할 수 없다

이론적으로 어떤 수를 0으로 나눌 수 없는데 실제로 try 해보면 다음 용어 정의에 있는 것 들이 나온다.

**\*BOOLEAN 자료형**

std::cout<<std::boolalpha; //true와 false값을 “true”, “false”라고 출력해준다.

std::cout<<std::noboolalpha; //출력 해제

\*char 자료형

//C-style Casting

cout << (char)65 << endl; //65를 character로 바꾼다.

cout << (int)'A' << endl;

//C++ style Casting

cout << char(65) << endl; //65로 초기화 되는 char를 생성한다.

cout << int('A') << endl;

Q. \n(New Line) 과 endl(End Line)의 차이점

* New Line은 단순히 줄 바꿈만 하지만, End Line은 줄 바꿈과 동시에 출력 버퍼에 있는 모든 내용을 출력하라는 의미도 포함된다.(간단한 프로그램의 경우 이 차이를 체감하기 어렵지만 코드가 길어지다 보면 차이점을 발견할 수 있다.)
* 참고로 std::flush; 는 줄 바꿈을 하지않고 버퍼에 있는 모든 내용을 출력하는 기능을 한다.

\a (Alert)의 용도

Running Time이 오래 걸리는 프로그램을 실행시킬 때

실행이 완료 됐다는 알림 소리를 프로그램상에 넣어줄 때 유용하다.

**\*리터럴 상수 (Literal Constants)**

리터럴 뒤에 붙는 알파벳들…

(1) f float

(2) u unsigned

(3) l long double

Ex) float f = 3.141592f;

int i = -12345u;