<후위 연산 전위연산>

과제 : 전위 연산이 후위 연산보다 권장되는 이유

이유 : 두 가지 이유로 설명하자면, 첫 번째로 반복문에서 사용될 때 수행능력이 떨어지기 때문이다.

후위 연산자를 오버로딩하는 함수는 리턴타입이 클래스형 타입이다. 이때 함수가 return 문을 통해 객체를 반환할때 임시객체가 생성된다. 이 임시 객체가 만들어질 때 복사 생성자가 호출되는데, 문제는 이 복사 생성자가 반복 문의 반복 횟수만큼 다시 임시 객체를 만들고 복사 생성자가 호출되는 일이 반복되므로 수행능력을 떨어뜨린다.

두 번째로 후위 연산자의 반환형이 const 일 경우, 생성되는 임시 객체를 const 객체로 생성하게 되는데, 이 객체를 a라 하고 main에서 (a++)++이런 식으로 수행하게 되면 (const 임시 객체)++이므로 (const형 임시 객체). operator++(); 이런 식으로 해석된다. 그러나 const 객체는 const로 선언이 된 함수만을 호출할 수 있는데, operator++함수는 const 함수가 아니므로 컴파일 에러가 발생한다.

따라서 후위 연산보다 전위 연산이 권장된다

<pow() 사용 안하고 출력>

#include <iostream>

#include <iomanip>

using namespace std;

int main()

{

double amount,rate= .05;

int x=0;

double result=1,principal = 1000.0;

cout <<"Year"<<setw(21)<<"Amount on deposit"<<endl;

cout <<fixed<<setprecision(2);

cout <<"년수 입력:";

cin >>x;

for(int n=1;n<=x;n++)

{

result\*=(1+rate);

}

amount=principal\*result;

cout<<setw(4)<<x<<setw(21)<<amount<<endl;

}.