Program to demonstrate that the order of summation of even positive numbers matters

Programmer: Dick Furnstahl furnstahl.1@osu.edu

Revision history:
01/02/11 new version from demol.cpp

Notes:
* An example to try to understand summing upward vs. summing downward. Add a small number eps (slightly below single-precision machine precision) many times to 1. It makes a big difference (in single precision) whether you do 1 + eps + eps + ... or eps + eps + ... + 1.

* First pass: no subroutine
* Use single precision AND double precision at the same time
* Here is the output with eps=5e−7 added 10^7 times:

1+eps+eps+...  eps+eps+...+1
single precision:  1.0000000000  1.5323836803
double precision:  1.4999999992  1.4999999999

To do:

#include files
#include <iostream>    // note that .h is omitted
#include <iomanip>     // note that .h is omitted
using namespace std;  // we need this when .h is omitted

//*******************************************************************************
// include files
#include <iostream>    // note that .h is omitted
#include <iomanip>     // note that .h is omitted
using namespace std;  // we need this when .h is omitted

//********************************************************** begin main **********************************************************

int main ()
{
    int num_eps = 100000000; // number of times to add eps to 1
    float eps_sp = 5.e-7;    // single precision small increment
    double eps_dp = 5.e-7;   // double precision small increment

    float sum_first_sp = 1.; // adding 1 first (single precision)
    float sum_last_sp = 0.;  // adding 1 at the end (single precision)
    double sum_first_dp = 1.; // adding 1 first (double precision)
    double sum_last_dp = 0.; // adding 1 at the end (double precision)

    cout << endl << "Adding small numbers many times to a big number:"
    cout << endl;

    // add small numbers (in single or double precision) num_eps times
    for (int i = 0; i < num_eps; i++)
    {
        sum_first_sp += eps_sp;
        sum_last_sp += eps_sp;
        sum_first_dp += eps_dp;
        sum_last_dp += eps_dp;
    }

    sum_last_sp += 1.; // add 1 at the end
    sum_last_dp += 1.; // add 1 at the end

    cout << endl << "1+eps+eps+... eps+eps+...+1 
    single precision: " << fixed << setprecision (15) << setw (20) << sum_first_sp << endl;
    cout << "double precision: " << fixed << setprecision (10) << setw (20) << sum_first_dp << " \n";
    return (0);
}