

```
/* SSS_Enhanced.cpp
```

This program solves the side-side-side (SSS) problem in trigonometry.

More specifically, if the lengths of three sides of a triangle are known, this program will solve for the three angles.

Units: The units of input for the three side lengths is actually unimportant, so long as all three sides are given in the same units. This program reports results in both radians and degrees.

Written August 2011 by John Bell, as a sample solution to HW1 for CS 109

This version enhances the basic assignment, by adding a number of optional enhancements:

1. User input is checked for validity.
2. The sum of the angles is compared to 180 for round-off error, using both doubles and floats.
3. Heights and areas of triangles are calculated
4. Multiple problems may be solved.

```
*/
```

```
#include <iostream>          // For input and output
#include <cmath>              // For sines and cosines, etc.
#include <iomanip>            // To format the output nicely.

using namespace std;        // For easy use of cin and cout

int main( void ) {

    // First to declare necessary variables.

    double sideA, sideB, sideC;          // 3 sides, any dimensions
    double angleA, angleB, angleC;       // in radians
    double angleA_deg, angleB_deg, angleC_deg; // in degrees
    double numerator, denominator;       // For use in cosine law
    double heightA, heightB, heightC;    // Relative to base sideA, etc.
    double areaA, areaB, areaC;          // sideA * heightA / 2.0, etc.
    double error;                        // 180 - sum( angles ), should be zero
    char answer;                         // To stop infinite loop.
    float fsideA, fsideB, fsideC, fangleA, fangleB, fangleC,
          fangleA_deg, fangleB_deg, fangleC_deg, fnumerator, fdenominator,
          ferror; // Same as double versions, to compare error of floats.

    // Next to explain the program to the user

    cerr << "\nWelcome to program SSS_Enhanced.\n\n";
    cerr << "Written August 2011 by John Bell, jbell, for CS 109.\n\n";
    cerr << "This program will find the three angles of a triangle,\n";
    cerr << "given the lengths of the three sides.\n\n";
    cerr << "Input can be given in any consistent units.\n";
    cerr << "Results are reported in both radians and degrees.\n\n";

    cerr << "This enhanced version also calculates heights and areas,\n";
    cerr << "and reports observed round-off errors.\n";
    cerr << "Multiple problems may also be solved.\n\n";

    // Now to begin the work
```

```

while( true ) { // Infinite loop to solve multiple problems.

    // Now get input from the user.
    // Check for valid input and repeat as needed.

    while( true ) {
        cerr << "Please enter the length of the longest side: ";
        cin >> sideC;
        if( sideC > 0.0 )
            break;
        cerr << "Error: " << sideC << " is not greater than zero.\n";
        cerr << "Please try again.\n";
    } // while loop checking sideC

    while( true ) {
        cerr << "Now please enter the length of the second side: ";
        cin >> sideA;
        if( sideA > 0.0 && sideA <= sideC )
            break;
        cerr << "Error: " << sideA << " is not between 0.0 and "
            << sideC << endl;
        cerr << "Please try again.\n";
    } // while loop checking sideA

    while( true ) {
        cerr << "And finally, please enter the length of the third side: ";
        cin >> sideB;
        if( sideB > sideC - sideA && sideB <= sideC )
            break;
        cerr << "Error: " << sideB << " is not between " << sideC - sideA
            << " and " << sideC << endl;
        cerr << "Please try again.\n";
    } // while loop checking sideB

    // Convert the double inputs into floats

    fsideA = ( float ) sideA;
    fsideB = ( float ) sideB;
    fsideC = ( float ) sideC;

    // First calculate angle C using the cosine law; Repeat with floats

    numerator = sideA * sideA + sideB * sideB - sideC * sideC;
    denominator = 2.0 * sideA * sideB;
    angleC = acos( numerator / denominator );

    fnumerator = fsideA * fsideA + fsideB * fsideB - fsideC * fsideC;
    fdenominator = 2.0f * fsideA * fsideB;
    fangleC = acos( fnumerator / fdenominator );

    // Next calculate angles A and B using the sine law; Repeat w. floats

    angleA = asin( sideA / sideC * sin( angleC ) );
    angleB = asin( sideB / sideC * sin( angleC ) );

    fangleA = asin( fsideA / fsideC * sin( fangleC ) );
    fangleB = asin( fsideB / fsideC * sin( fangleC ) );

    // Next calculates heights perpendicular to each side

```

```

heightC = sideB * sin( angleA );
heightA = sideC * sin( angleB );
heightB = sideC * sin( angleA );

// And the areas

areaA = sideA * heightA / 2.0;
areaB = sideB * heightB / 2.0;
areaC = sideC * heightC / 2.0;

// Convert angles from radians to degrees; Repeat with floats

angleA_deg = angleA * 180.0 / M_PI; // M_PI defined in <cmath>
angleB_deg = angleB * 180.0 / M_PI;
angleC_deg = angleC * 180.0 / M_PI;

fangleA_deg = fangleA * 180.0f / M_PI;
fangleB_deg = fangleB * 180.0f / M_PI;
fangleC_deg = fangleC * 180.0f / M_PI;

// Calculate round-off errors in degree totals, in double and float

error = 180.0 - ( angleA_deg + angleB_deg + angleC_deg );

ferror = 180.0f - ( fangleA_deg + fangleB_deg + fangleC_deg );

// And finally, echo input and report results.
// Formatting is applied in this version.

cout << "\n\nHere are your results:\n\n";
cout << "      Side      Opposing Angle      Height      Area\n";
cout << "      Length      ( radians ) ( degrees )\n";

cout.unsetf( ios::scientific );
cout.setf( ios::fixed );

cout << setw( 10 ) << setprecision( 4 ) << sideC
    << setw( 11 ) << setprecision( 4 ) << angleC
    << setw( 13 ) << setprecision( 4 ) << angleC_deg
    << setw( 12 ) << setprecision( 4 ) << heightC
    << setw( 13 ) << setprecision( 4 ) << areaC << endl;

cout << setw( 10 ) << setprecision( 4 ) << sideA
    << setw( 11 ) << setprecision( 4 ) << angleA
    << setw( 13 ) << setprecision( 4 ) << angleA_deg
    << setw( 12 ) << setprecision( 4 ) << heightA
    << setw( 13 ) << setprecision( 4 ) << areaA << endl;

cout << setw( 10 ) << setprecision( 4 ) << sideB
    << setw( 11 ) << setprecision( 4 ) << angleB
    << setw( 13 ) << setprecision( 4 ) << angleB_deg
    << setw( 12 ) << setprecision( 4 ) << heightB
    << setw( 13 ) << setprecision( 4 ) << areaB << endl << endl;

cout.unsetf( ios::fixed );
cout.setf( ios::scientific );
cout << "The error in the sum of the angles is\n\t"
    << error << " using doubles, and\n\t "
    << ferror << " using floats." << endl << endl;

// Now to check and see if they want to solve any more problems

```

```
        cerr << "Would you like to solve another problem? Answer Y or N: ";
        cin >> answer;
        if( answer != 'Y' && answer != 'y' )
            break;

    } // End of the infinite while loop

    // Tha tha tha that's all folks !

    system( "PAUSE" );    // Only needed for Dev C++

    return 0;

} // main
```