1. Modify the code from your HW1 as follows:

   Create an "overloaded" version of your `triangle` function that works with `doubles`. If `main` is executed with the first command-line argument "-d", it is to work with `doubles`. If the first command line argument is "-i", it is to behave as in HW1. If the first command-line argument is *NOT* "-d", it is to behave as in HW1.

2. The `void triangle (double a, double b, double 3)` function should use appropriate i/o manipulators to display the three lengths with 5 places to the right of the radix point.

3. The `double` version of your `triangle` function needs to define a constant called `EPSILON` with a value of 0.001. Values are to be considered equal if the absolute value of their difference is less than `EPSILON`.

4. The `main` function is only allowed to use `cin` and `cerr`. The `triangle` functions are only allowed to use `cout`. It is also responsible for rearranging the values `a, b, c` such that `c` is at least as large as the other two lengths.

5. Supply a complete program that exercises your class. (be sure you also exercise the program completely (both the interactive and command line versions, including any boundary conditions. Be certain to test that the program behaves as in HW1 when "-d" is not specified).

   Sample output (the first 3 when executed with "-d"):

   1.00000 1.00000 1.01000 Isosceles triangle
   1.00000 1.00000 1.00001 Equilateral triangle
   3.00000 4.00000 5.00002 Right Scalene triangle
   3 4 5 Right Scalene triangle
   1 1 9 Not a triangle

You must supply a listing of your program and sample output.