

This assignment calls for doing a partial implementation of the abstract data type "MyFloat". This ADT is to serve a specialized need for small floats between 0 and 1 that can have up to 20 digits of accuracy. The creation of the MyFloat class will allow calculations with accuracy not possible with floats. Recall that to implement an abstract data type, you must:

1) **Choose a concrete data representation for the abstract data type.** When programming in C++, this usually requires coming up with a suitable class definition. If possible, this class should be chosen in such a way that

- a) memory is conserved ( for example using 16-bit int to store a single digit is wasteful.)
- b) the operations in step 2 will be efficient

2) **Implement the operations on the data values**, i.e. write the member functions of the class MyFloat. For this assignment you'll need to implement the following member functions:

Digits	Returns the number of digits in a MyFloat. Important: if Digits is 0 then your code has previously detected an illegally formatted float.
MaxDigits	Returns the maximum number of possible digits in a MyFloat.
Read	Reads a MyFloat from standard input. If an error is detected in reading the MyFloat X, X.Digits() should return a 0.
Write	Writes a leading 0, a decimal point, and the digits in a MyFloat to standard output. If the MyFloat has 0 digits (error flag), output should be '0.'
+	Adds two MyFloat variables X and Y together and stores results in MyFloat Z. Standard addition rules apply. Carry over into ones field is ignored for this assignment.
==	returns a 1 if two MyFloats are equal, 0 otherwise.
Constructor	Need to write a default constructor, and make sure set NumberofDigits to 0.

#### Hints, Suggestions and Warnings:

1) There is a driver program named "TestMyFl.cpp" on my Web site, copy this to use. This will facilitate the testing of your class. The test driver assumes that the specification and implementation file are combined into the same file called "MyFloat.cpp" and is in the current directory. Make sure to read this driver program first to understand what to do.

2) Recall good programmers write and debug functions one at a time! Again read "TestMyFl.cpp" first to help understand which one to write first. Note that function stubs will be needed in your class to get TestMyFl.cpp to compile and run. Also remember when coding your file MyFloat.cpp to just compile it and not MAKE it. Compile will find the syntax errors.

3) Do not underestimate the amount of help you can get by studying the examples that we looked at during class. They and the book(s) are very helpful.

4) When writing Read, **be sure to follow standard floating point read conventions.** Leading white space and zeros are ignored, a decimal point is required, and input stops when a non-numeric char is detected.

5) In the Read function, you will be storing the digits in an array of char. Be sure to convert chars such as '0', '1', etc. to corresponding ints 0, 1, etc. before storing in your array of digits. Otherwise you will pay a server penalty when doing next assignment! **Also need to use "cin.putback(Ch);" in the Read function.**

6) Remember if Ch is a char with value 1 (not '1'), "cout << Ch"; will display a happy face!

7) **Do not** turn in a listing of TestMyFl.cpp, I already have it.

8) Warning: In your comments, be sure to carefully describe how you store a typical MyFloat! If you don't you will lose points. Remember comments are very important.