

## Assignment 2 要求

In the Shapes v.1 an input file was read and processed line-by-line. For possible use in a GUI program later, it would be important to read all of the input without immediate processing, and to process it all afterwards -- perhaps several times (for example, in a GUI window's "paint" function as the user pans and zooms).

So to make that easier, the shapes described in the input file should be read, parsed, made into objects, and stored in a bag. Afterwards, the objects can be retrieved from the bag and we can use their supporting functions to output their metrics. That way they are stored in memory and we don't have to go back to the disk file every time we want to do something with the objects stored there.

Copy Shapes.1.cpp from [Assignment 1, Shapes v.1](#) to **Shapes.2.cpp** for this assignment. Do *not* use more than one source file -- that means no **.h** files, and just a single **.cpp**. (*To use multiple files would needlessly complicate this solution.*) With one exception, the console output should match that of the Assignment 1 version. Unlike Assignment 1, *read all of the file data before producing any object output*. Only after the input file is read and closed, produce output for the objects. Here is the exception: any invalid object names will appear at the **TOP** of the console output list, outputted in the loop that reads the input file and stores the valid objects in a bag. Invalid objects are **NOT** stored in the bag, nor included in the **TEXT** output, as they are ignored in any further processing.

## Program Changes

1. Modify the program so that it includes **struct** s for each of the eight shapes that were the subject of Assignment 1. Name these as follows: Square, Rectangle, Circle, Triangle, Cube, Box, Cylinder, and Prism. Include attributes of the **double** data type for the required dimensions for each (for example, length and width for the Rectangle). Provide a descriptive name for each of these data elements (for example, "radius" instead of "r"). Do NOT include any other attributes, and do NOT include any "member functions" -- avoid the temptation to extend the specifications beyond what is required. If you include attributes for area, volume, or perimeter, *you are not doing this right!*
2. Write supporting functions to do console output each shape (for example, **void outputBox(ostream&, const Box&);** ), which should output to either **cout** or **fout** -- whichever is the first parameter in a call, using the same format as v.1. Note that **cout** is an object of type **ostream** . So use **cout** as the first parameter in the function call in order to produce console output. Use the exact same supporting functions for text file output, which should output to **fout** . Create, open, and close **fout** in **int main** . **fout** is also an object of type **ostream** . Remember, the **ROUNDING** Format -

the calculated results to 2 digits after the decimal (`cout << setprecision(2);`). But echo the input values *without formatting*. (`cout << setprecision(6);`)

3. Use any type of array for your bag -- C, C++, or STL vector -- your choice. STL vectors track their size, so if you use a C or C++ array instead, you'll have to use an `int` to track size yourself.
4. Write 4 loops -- one to process the input file and fill the bag, one for console output, one for TXT output, and one for deallocation. If you don't have 4 loops, or if you do more than one of these things in any single loop, you're not doing this right!