**CS 2413: 002/004**

**FALL - 2018**

**PROJECT**

Implement a menu-driven program for managing a software store. All information about the available software is stored in a file called software.dat. This information includes the *name*, *version*, *quantity*, and *price* of each package. A sample software.txt is shown below:

Adobe Photoshop CS5 21 580

Norton Utilities 1.0 10 50

Norton System Works 2009 6 50

Visual Studio 2017 19 700

Microsoft Office 365 27 150

Every time the program starts, it reads the software.dat file and creates a **binary search tree** with each node having *name* of the package as its key value. A node should also store other details such as version, quantity etc. as described above. You may read the file line by line or use a binary file to instantiate your binary tree (See **Useful** **Resources** section at the end). If software.dat doesn’t exist, then ask the user to specify the total number of packages and input the respective details and save it to the file.

The program should allow the file and tree to be updated when new software packages arrive at the store and when some packages are sold. The tree is updated in the usual way. All packages are entry ordered in the file software i.e. if a new package arrives, then it is put at the *end of the file*. If the package already has an entry in the tree (and the file), then only the *quantity* field is updated. If a package is sold out, the corresponding node is deleted from the tree, and the quantity field in the file is changed to **0**. To make the program more efficient, you can additionally store an integer in the tree node to store its corresponding location in the file. For e.g., the software.dat file after selling all six copies of Norton System Works, should look like:

Adobe Photoshop CS5 21 580

Norton Utilities 1.0 10 50

Norton System Works 2009 0 50

Visual Studio 2017 19 700

Microsoft Office 365 27 150

If an exit option is chosen from the menu, the program cleans up the file by moving entries from the end of the file to the positions marked with 0 quantities. For example, the previous file becomes:

Adobe Photoshop CS5 21 580

Norton Utilities 1.0 10 50

Microsoft Office 365 27 150

Visual Studio 2017 19 700

**#Deliverables**

1. Your complete source code files in a **single** zip file. This includes all the header files, cpp files etc. and your software.dat file that you used to test your code.

**#Useful Resources**

* File Handling:
  + <https://www.geeksforgeeks.org/file-handling-c-classes/>
  + <https://csegeek.com/csegeek/view/tutorials/cpp_lang/cpp_file.php> (binary files)
  + <https://www.includehelp.com/code-snippets/cpp-program-to-write-and-read-an-object-in-from-a-binary-file.aspx> (binary files)
* C++ String Class
  + <http://www.cplusplus.com/reference/string/string/>
* XCode File Handling issues (for people who use Mac):
  + <https://stackoverflow.com/questions/16779149/c-program-in-xcode-not-outputting-simple-text-file-using-outfile>