**BOLDED TEXT – finished implementing**

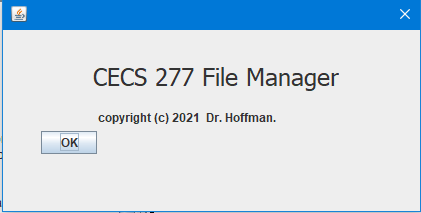
1. **Main frame: Main feature presentation of multiple frames containing a split-pane.**
2. **Secondary frames: holds a split-pane, both sides of which contain scrollpanes. (frames are independent of each other)**
   1. **Left side of scrollpane contains a tree structure representing the entire folder structure of the entire drive**
      1. **The root of the tree in the left side is always a drive**
      2. **The tree root is always expanded**
      3. **Each entry in the tree is a directory (folder). No files**.
      4. **Directories have an icon with a ‘+’ to indicate it has subdirectories (right side does not need to put + on the folders**
      5. **Don’t read entire drive into tree at once. It is necessary to read at least one level further into a directory to determine if it has subdirectories unless there’s a package that does that for us**
      6. **Double clicking the folder in the tree “expands” that branch (only that branch, not the subdirectories)**
      7. **Each folder must be double clicked to expand it. (Except the root)**
      8. **Clicking a tree branch (single or double) will display its corresponding content in the right scrollpane**
   2. **Right side of the scrollpane contains a list of all the folders and files in the folder** **which is currently selected in the left scrollpane**
      1. **Files are presented with different icons and additional information.**
         1. **File icon is similar to that of a windows .html or .txt file**
         2. **Additional info is the file’s last modified date and its size in bytes**
   3. **Selecting a different folder in the left scrollpane causes the right scrollpane to redraw to show the folders and files that belong to the newly selected folder**
   4. **Each frame can be closed, minimized (iconified), maximized (de-iconified), and moved within the main frame**
3. Secondary features (assumes a file has been selected by clicking)
   1. Single click
      1. Renaming a file (The selected file can be renamed in its directory)
      2. Copying a file (The selected file can be copied to its directory (by renaming or copying) or copied to a different directory.
      3. Deleting a file (The selected file will be deleted from its directory)
         1. How to rename, copy, and delete? 2 ways to do the same thing
            1. File menu entries for rename, copy, and delete
            2. A popup menu that opens over the file

An action listener can be written to respond to the commands. For rename and copy, you will just need a small dialog box to get information. For copying and renaming, dialog requires the user to provide a complete file path and file name for the “To:” textbox. The result is that the “From:” file is renamed to the current directory or copied as it is named to the new directory name provided by the user

For delete, you will need to use a JOptionPane to confirm the deletion. Starts the same way by clicking and then selecting delete, but doesn’t require additional information, just confirmation

* 1. Double click
     1. Execute the file if it is an executable
     2. Open the registered application for that type of file and load the file.
        1. Ex. If it is a .doc file, the program registered with the OS for that file (Word) will be launched and the file opened in Word. (All it takes to execute the file is one declaration and one method call!)
        2. If there is no default program, the OS will ask the user what to use (takes one line of code to execute any file)
  2. **Copying by dragging and dropping (DnD)**
     1. **Must be capable of receiving files from sources other than the default Windows Folder Manager (external drag and drop)**
        1. **Drop external files to the right scrollpane to add it to that directory (All this operation requires is a DropTarget class in your project)**
     2. **Must also be capable of copying by DnD from one frame to another (internal drag and drop)**
        1. **To do internal DnD, the project must add additional classes for drag gesture recognition and transfer file preparation**

1. Miscellaneous features
   1. **Menu toolbar for main frame**
      1. **File: Rename, copy, delete, run, exit**
         1. Rename, copy, delete, and run all share code from the popup menu
      2. **Tree: Expand branch, Collapse branch**
         1. **If a branch in the directory tree is selected and it has subdirectories and Expand Branch is selected, it expands only one level**
         2. **If a branch is selected and Collapse Branch is clicked, that branch is collapsed.**
            1. **If a Subbranch of the collapsed branch are expanded, then upon expanding the collapsed branch, the subbranches are expanded too**
      3. **Window: New, Cascade**
         1. **New creates a new internal frame and places it at location 0,100 within the desktop pane, Defaults to C:, Any number of new frames can be created**
         2. **Cascade: stacks the open frames in a nice, neat overlapping style**
      4. **Help: Help, About**
         1. Left to your imagination
         2. **About shows**



1. **Toolbar (at the top of the frame, below Menu toolbar, but above the desktop)**
   1. **Combobox for drive selection (Every drive on the computer must be found and listed, even USB drives that can be disconnected, jump drives, cd, dvd, fixed hard drives, every drive) However, the program does not need to recognize if drives change dynamically.**
   2. **Simple button**
      1. **Clicking this will redraw the right scrollpane without file sizes and dates**
   3. **Details button**
      1. **Clicking this will restores the file sizes and dates**
2. **Statusbar (at the very bottom of the application)**
   1. **A small string that provides the drive information.**
      1. **“Current Drive: C:\ Free Space: 567GB Used Space: 373GB Total Space: 952GB”**
   2. **Must reflect which drive is currently displayed in the frame that has the focus of input**

How to start:

1. Start with the main frame. I will provide some videos on how to use Java Swing classes.
2. Add the menu system (with no operations linked), the toolbar, statusbar.
3. Develop the desktop pane. I will do a video.
4. Develop the tree side first (left). Then do the file list(right).
5. Figure out the multiple frame concepts. I will do a video.
6. Add the popup menu. Code the separate functions in their own handler classes.
7. Link the handler classes to the menu system.
8. Finish off with expand/collapse, cascading.

