
Lab Report – Week [4] - [Files and Directories Programming Assignment]

[Leo Vergnetti]

[CSCI 112-online] [Spring]

Assignment Analysis and Design

The assignment was to design software that copies a user-specified directory and all of its contents / subdirectories to a separate user specified directory. I created a Class called DirectoryCopier that handles the class specific code. The main method creates an instance of DirectoryCopier, and after two quick Prompt and Captures, passes the source directory and target directory to the DirectoryCopier classes cloneFile() method. This is the public method that the user is permitted to use. First this method calls its own public method validate, which returns TRUE if the target directory is outside of the source directory (very unsatisfied with this method, as it does so using a basic String comparison). This is done to prevent the user from causing an infinite spawn of folders. Assuming all is well, copyFile() is called. I created copyDirectory as a private method, to prevent the user from copying without first validating that the source and target directories are safe. The copyDirectory method makes a copy of the directory in the target directory. An enhanced for statement iterates a File array of the files in the directory. If it comes across a file, it calls the private copyFile method (outlined in chapter 18). If it finds a directory, it recursively calls copyDirectory.

Assignment Code

Source code included in attached file.

Assignment Testing

Testing this was difficult. I created the validate method, however it does not perform adequately enough to prevent all cases of subdirectories. The copy method works well. I copied a few folders and directories on my hard drive, and it worked well. I have not yet tested copying directories to an external device as yet .

Assignment Evaluation

I loved working on this project. I am still unsatisfied with the validate method, but it functions crudely enough. This is done from preventing the user (me) from recursively creating a folders of folders nested so deeply in the hard drive so as to prevent the user from removing it via conventional means. This forces the (careless) user to write a program to VERY CAREFULLY recursively remove the files. Such a careless user may have to divert an hour or two to this nerve racking and tedious task. Otherwise I really enjoyed attempting to restrict access to some of the methods inside of the Class, while attempting to separate the class from the implementation of the interface.