CPSC 210
FileSystem Lecture/Lab
Composite Design Pattern

1. Check out the `CompositeFileSystem` project from the lectures repository.

2. Apply the Composite Design Pattern to the existing classes defined in the project so that a Folder can be treated similarly to a File. Make use of all of the classes that you have been given.

3. Implement your design.

4. Make `Driver.java` compile by adding appropriate code to the classes in the project.

5. Implement the `print` functionality so that running `Driver` as a Java application prints the following to the screen:

   Folder: dirA
   File: fileInA
   Folder: dirB-in-dirA
   File: fileInB1
   File: fileInB2
   Folder: dirC-in-dirA
   File: fileInC
   Folder: dirD-in-dirC

   Note: Files must have unique names within a Folder. Similarly, Folders must have unique names within a Folder. That is, adding a File named x to a Folder that already contains a File named x should do nothing. However, adding a File named x to a Folder that already contains a Folder named x, will add the File to the Folder.

6. If you are finished everything else, modify your print method (in FileSystemResource and all of its subclasses) so that the nesting level of FileSystemResources is shown. The previous example should now print as:

   Folder: dirA
     File: fileInA
     Folder: dirB-in-dirA
       File: fileInB1
       File: fileInB2
     Folder: dirC-in-dirA
       File: fileInC
       Folder: dirD-in-dirC