CSC 385
Module 2 Homework

One possible solution

Here is one possible solution to the DirectoryLister assignment, showing only the code for the two methods you needed to implement: `showDirectoryContents` and `enumerateDirectory`.

ShowDirectoryContents

```
/**
  * Show the directory listing.
  * An error message is displayed if basePath does not represent a valid directory.
  *
  * @param basePath the absolute path of a directory in the file system.
  */
public void showDirectoryContents(String basePath)
{
    try
    {
        // create File object for the selected directory
        File source = new File(basePath);

        // validity check
        if (source.exists() && source.isDirectory())
        {
            // valid directory path
            enumerateDirectory(source);
        }
        else
        {
            // invalid directory path
            // can happen if the user clicks the Cancel button on the JFileChooser
dialog,
            // which will return a null value for basePath
            throw new Exception();
        }
    }
    catch(Exception e)
    {
        // display stack trace for debug purposes
        e.printStackTrace();

        // display error message to user
        if (basePath == null)
        {
            JOptionPane.showMessageDialog(null, "No directory chosen!", "Error",
            JOptionPane.ERROR_MESSAGE);
        }
        else
        {
            JOptionPane.showMessageDialog(null, "Exception - " + basePath + " is not a
valid file or directory!", "Error", JOptionPane.ERROR_MESSAGE);
        }
    }
}
```

enumerateDirectory

```
/**
  * Recursive method to enumerate the contents of a directory.
  *
```

```
     * @param f directory to enumerate
     */
    private void enumerateDirectory(File f)
    {
        // update the GUI with current file/folder information
        gui.updateListing(f.getAbsolutePath(), (f.isDirectory() ? "" :
    getSizeString(f.length())),
                          (f.isDirectory() ? "Folder" : "File"),
    formattedDateString(f.lastModified()));

        // do recursion if f is a subfolder
        if (f.isDirectory())
        {
            // get files/folders in f
            File[] fileList = f.listFiles();

            // recursively process subdirectory
            for (int i = 0; i < fileList.length; i++)
            {
                enumerateDirectory(fileList[i]);
            }
        }
    }
```