**Pseudocode for File Moving App**

Start

Create current Directory as “none”

Create copy Directory as “none”

Create an empty list for file types

Create an empty list for selected files

Create a window, resize it and name it

Set var to stringvar

Create a label for file name. Color it, change font and width, as well giving it the text variable with the string var

Set var2 to stringvar

Create another label for the file count and give it the stringvar

Place it appropriately on the window

Set var3 to stringvar

Create another label for the copy location and give it the stringvar

Place it appropriately on the window

Set var4 to stringvar

Create another label for the created file and give it the stringvar

Place it appropriately on the window

Define Load Folder

Get current directory as a global

Clear the option menu

set file types to empty

set list box to empty

Create an empty list called list

Path = filedialog.askdirectory

Make current directory as path and print it

For files in os.listdir(path)

if the os path is file, make them join path and files

add files to list

For item in list:

insert item in listbox

For item in ext:

Fname, fExtensions = os.path.splitext(ext)

add fExtensions to FileTypes

Display: FileTypes

Create dictionary called file\_type\_dictionary

Clear dictionary

Make the dictionary = x:0 for x in FileTypes

For I in file\_type\_dictionary:

make file\_type\_dictionary[I] into IntVar

Create empty list called fileTypeList

Get files from file\_type\_dictionary and add it to the empty list

Set the list to default value

Make total be the length of fileTypeList and convert to string

val = total + “Files”

Set var2 as val

Define getList(dict):

Return list itemgetter(0) and dict.items()

Define Select\_File\_Extensions:

Set I to 0

Global SelectedFiles

Clear SelectedFiles

Choice = get variable

For file in list directory of currentDirectory

if file.endswith(choice):

i += 1

file\_location = join currentDirectory and file

add file to the SelectedFile list

print the path join of currentDirectory and file

Set val as str(I) + “Files”

Set var2 as val

Display: “Selected Files: “ SelectedFiles

Define ClearOptionMenu

For widgets in the window and any children of the widgets

if widget is in the window

destroy widget

Display: “CLEARED”

Define moveFiles

If currentDirectory = “none”

message box to show information that they need to select a folder first

Else

bring copyDirectory

make a path to store the copied files while active

copyDirectory = path

set the var3 as copydirectory

bring SelectedFiles

for file in SelectedFiles

try

move currentDirectory+name of file, copyDirectory+name of file

Continue

except

DISPLAY:”Error: File not found”

Break

Except

DISPLAY:”Error: File(s) may be open, close it first then try again”

Break

Create Load folder button

Create Move Button

Create listbox

Define clickEvent(self):

If the size of the listbox > 0

Value = get listbox and return indices of currently selected items

Set var1 as value

Else

DISPLAY:”No files were loaded”

Create a click event for the list box select

Create scrollbar

Set scrollbar to window

Define CreateNewFolder

Get the message that’s in the entry box and define it as FolderName

FolderPath = filedialog.askdirectory

Join the FolderName and FolderPath

Try

Create a folder

Except

Display(“Enter a Folder name first”)

Set var4 as ‘FolderName’ was created

Create button to Create New Folder

Create an entry box

Define deleteButton

Bring currentDirectory

Idx = find file from listbox and its index from var1

Delete the idx from listbox

Try

Remove currentDirectory + ‘/’ + var1.get

Except

DISPLAY:”File not found”

Set var5 as ‘var1.get’ + “has been deleted

Set var5 as stringvar

Create a label to use for var5

Create a button for the delete files

Run mainloop