**Pseudocode**

***FILE EXPLORER.H***

CLASS FileExplorer

PUBLIC METHODS:

CONSTRUCTOR(FileExplorer, Logger& logger)

METHOD listFiles(directory: STRING)

METHOD copyFile(src: STRING, dest: STRING) -> BOOLEAN

METHOD moveFile(src: STRING, dest: STRING) -> BOOLEAN

METHOD deleteFile(path: STRING) -> BOOLEAN

METHOD createFile(path: STRING) -> BOOLEAN

METHOD searchFile(filename: STRING) -> BOOLEAN

METHOD setPermissions(path: STRING, permissions: INTEGER) -> BOOLEAN

METHOD changeDirectory(path: STRING)

METHOD goUpOneDirectory()

PRIVATE ATTRIBUTES:

Logger logger\_

END CLASS

***FILE EXPLORER.CPP***

IMPLEMENTATION FileExplorer

CONSTRUCTOR(FileExplorer, Logger& logger)

INITIALIZE logger\_ with logger

METHOD listFiles(directory: STRING)

LOG "Listing files in directory: " + directory

TRY

FOR each entry in directory

PRINT entry path

CATCH exception

LOG "Listing files failed: " + exception message

METHOD copyFile(src: STRING, dest: STRING) -> BOOLEAN

LOG "Copying file from " + src + " to " + dest

TRY

COPY src to dest

LOG "File copied successfully."

RETURN TRUE

CATCH exception

LOG "Copy failed: " + exception message

RETURN FALSE

METHOD moveFile(src: STRING, dest: STRING) -> BOOLEAN

LOG "Moving file from " + src + " to " + dest

TRY

RENAME src to dest

LOG "File moved successfully."

RETURN TRUE

CATCH exception

LOG "Move failed: " + exception message

RETURN FALSE

METHOD deleteFile(path: STRING) -> BOOLEAN

LOG "Deleting file: " + path

TRY

REMOVE path

LOG "File deleted successfully."

RETURN TRUE

CATCH exception

LOG "Delete failed: " + exception message

RETURN FALSE

METHOD createFile(path: STRING) -> BOOLEAN

LOG "Creating file: " + path

TRY

OPEN file at path

CLOSE file

LOG "File created successfully."

RETURN TRUE

CATCH exception

LOG "File creation failed."

RETURN FALSE

METHOD searchFile(filename: STRING) -> BOOLEAN

LOG "Searching for file: " + filename

TRY

FOR each entry in recursive directory iterator

IF entry filename matches filename

LOG "File found: " + entry path

RETURN TRUE

LOG "File not found."

RETURN FALSE

CATCH exception

LOG "Search failed: " + exception message

RETURN FALSE

METHOD setPermissions(path: STRING, permissions: INTEGER) -> BOOLEAN

LOG "Setting permissions for file: " + path

TRY

SET permissions of path to permissions

LOG "Permissions set successfully."

RETURN TRUE

CATCH exception

LOG "Setting permissions failed: " + exception message

RETURN FALSE

METHOD changeDirectory(path: STRING)

LOG "Changing directory to: " + path

TRY

CHANGE directory to path

LOG "Directory changed successfully."

CATCH exception

LOG "Change directory failed: " + exception message

METHOD goUpOneDirectory()

LOG "Going up one directory level."

TRY

CHANGE directory to parent directory

LOG "Directory changed successfully."

CATCH exception

LOG "Change directory failed: " + exception message

END IMPLEMENTATION

***LOGGER.H***

CLASS Logger

PUBLIC METHODS:

CONSTRUCTOR(Logger, filename: STRING)

METHOD log(message: STRING)

PRIVATE ATTRIBUTES:

FILE logFile\_

END CLASS

***LOGGER.CPP***

IMPLEMENTATION Logger

CONSTRUCTOR(Logger, filename: STRING)

OPEN logFile with filename in append mode

METHOD log(message: STRING)

IF logFile is open

WRITE message to logFile

ELSE

PRINT "Failed to open log file."

END IMPLEMENTATION

***MAIN.CPP***

PROGRAM main

INITIALIZE Logger with "file\_explorer.log"

INITIALIZE FileExplorer with Logger

REPEAT UNTIL user chooses to exit

PRINT current directory

PRINT menu options

GET user choice

SWITCH choice:

CASE 1:

PROMPT for directory path

CALL explorer.listFiles with path

CASE 2:

PROMPT for source and destination file names

CALL explorer.copyFile with src and dest

CASE 3:

PROMPT for source and destination file names

CALL explorer.moveFile with src and dest

CASE 4:

PROMPT for file name to delete

CALL explorer.deleteFile with path

CASE 5:

PROMPT for file name to create

CALL explorer.createFile with path

CASE 6:

PROMPT for file name to search

CALL explorer.searchFile with filename

CASE 7:

PROMPT for file name and permissions

CALL explorer.setPermissions with path and permissions

CASE 8:

PROMPT for directory to navigate to

CALL explorer.changeDirectory with path

CASE 9:

CALL explorer.goUpOneDirectory

CASE 10:

PRINT "Exiting..."

DEFAULT:

PRINT "Invalid choice. Please try again."

END PROGRAM