Module 5 Critical Thinking – User Input Program

Malyk Parker

Colorado State University Global

CSC450-1 Programming III

Reginald Haseltine

May 18, 2025

Contents

[Git Repository Link 3](#_Toc198461198)

[Source Code Screenshot 3](#_Toc198461199)

[Output Screenshot 4](#_Toc198461200)

[Pseudocode 5](#_Toc198461201)

# Git Repository Link

<https://github.com/Malypar/MOD5-USERINPUT.git>

# Source Code Screenshot

#include <iostream>

#include <string>

#include <fstream>

#include <algorithm>

#include <mutex>

#include <thread>

std::mutex file\_mutex;

void appendToFile(const std::string& filename, const std::string& input) {

    std::lock\_guard<std::mutex> lock(file\_mutex);

    std::ofstream file(filename, std::ios::app);

    if (file.is\_open()) {

        file << input << std::endl;

        file.close();

    } else {

        std::cerr << "Unable To Open File For Appending." << std::endl;

    }

}

void reverseFile(const std::string& inputFile, const std::string& outputFile) {

    std::lock\_guard<std::mutex> lock(file\_mutex);

    std::ifstream inFile(inputFile);

    std::ofstream outFile(outputFile);

    if (inFile.is\_open() && outFile.is\_open()) {

        std::string totalContent((std::istreambuf\_iterator<char>(inFile)), std::istreambuf\_iterator<char>());

        std::reverse(totalContent.begin(), totalContent.end());

        outFile << totalContent;

        inFile.close();

        outFile.close();

    } else {

        std::cerr << "There Was An Error Opening The Files." << std::endl;

    }

}

int main() {

    std::string userInput;

    std::cout << "Enter Text To Append To File: \n";

    std::getline(std::cin, userInput);

    const std::string origianlFile = "CSC450\_CT5\_mod5.txt";

    const std::string reversedFile = "CSC450-mod5-reverse.txt";

    std::thread t1(appendToFile, origianlFile, userInput);

    std::thread t2(reverseFile, origianlFile, reversedFile);

    t1.join();

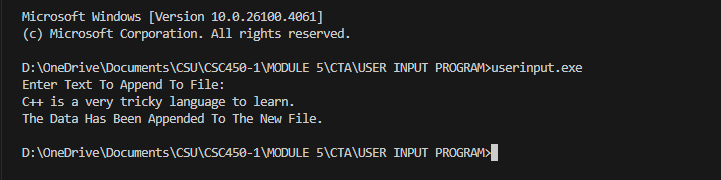
    t2.join();

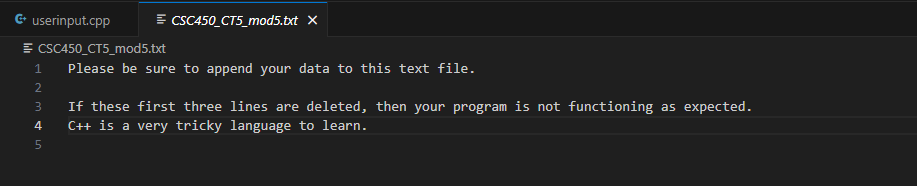
    std::cout << "The Data Has Been Appended To The New File." << std::endl;

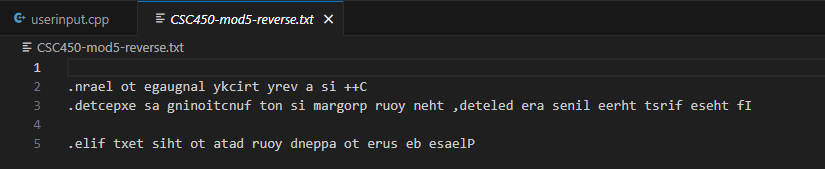
    return 0;

}

# Output Screenshot







# Pseudocode

BEGIN

Define a shared file lock (mutex)

FUNCTION appendToFile(fileName, userInput)

Acquire file lock

Open the file in append mode

IF file is open THEN

Write userInput to the file

Close the file

ELSE

Print "Unable to open file for appending"

ENDIF

Release file lock

END FUNCTION

FUNCTION reverseFile(inputFileName, outputFileName)

Acquire file lock

Open the input file for reading

Open the output file for writing

IF both files are open THEN

Read the entire content of the input file

Reverse all characters in the content

Write the reversed content to the output file

Close both files

ELSE

Print "Error opening files"

ENDIF

Release file lock

END FUNCTION

MAIN PROGRAM

PROMPT user to "Enter text to append to the file"

Read user input

Set originalFileName = "CSC450\_CT5\_mod5.txt"

Set reversedFileName = "CSC450-mod5-reverse.txt"

Create a thread to run appendToFile(originalFileName, userInput)

Create another thread to run reverseFile(originalFileName, reversedFileName)

Wait for both threads to finish

Print "Data appended and reversed file created."

END MAIN

END