EXPLORER

MODUL...

- .vscode
- CSC450_CT5_mod5.txt
- userInput.cpp
- userInput.exe

OUTLINE

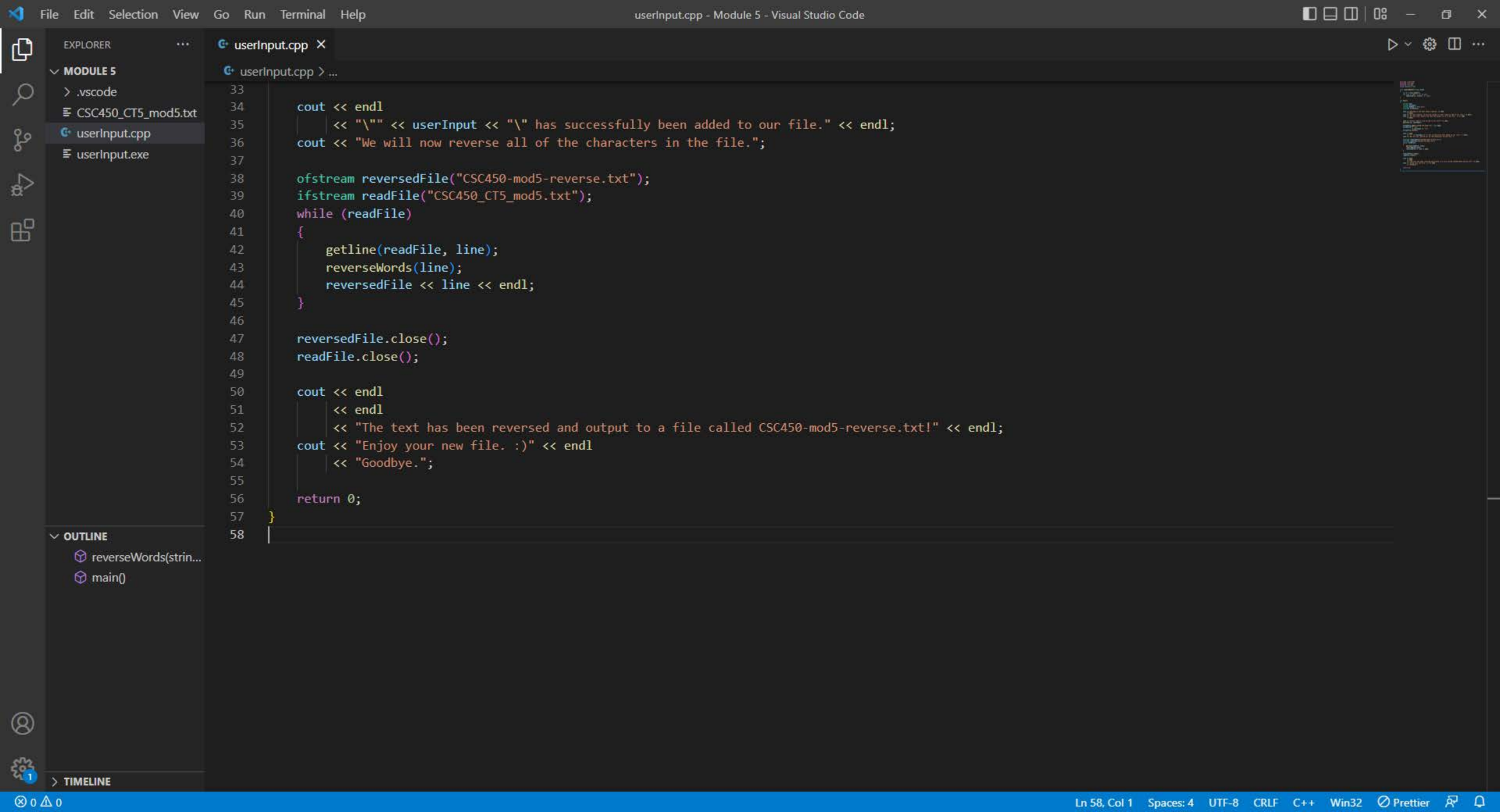
- reverseWords(strin...
- main()

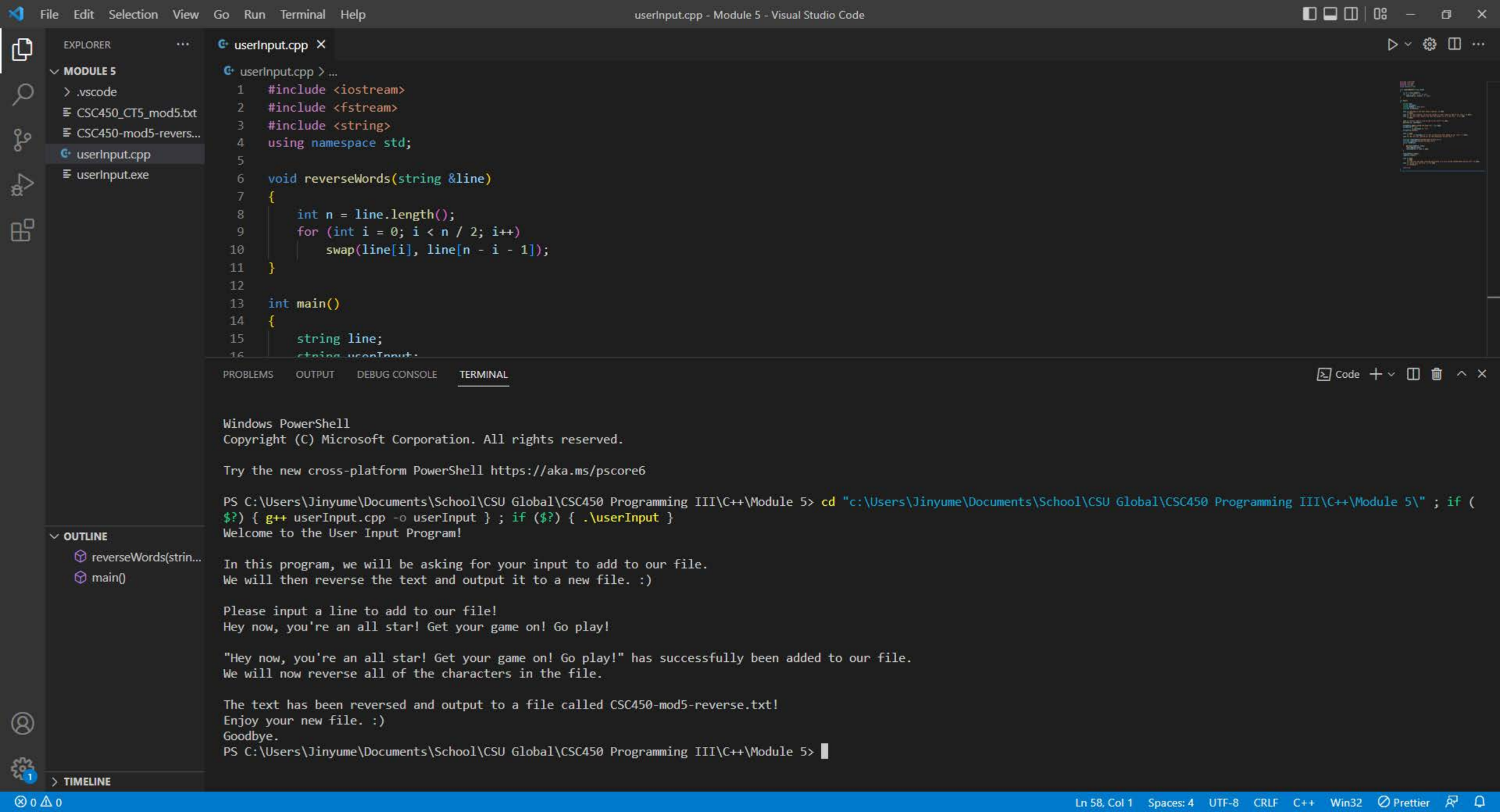
TIMELINE

userInput.cpp

userInput.cpp > main()

```cpp
#include <iostream>
#include <fstream>
#include <string>
using namespace std;

void reverseWords(string &line)
{
    int n = line.length();
    for (int i = 0; i < n / 2; i++)
        swap(line[i], line[n - i - 1]);
}

int main()
{
    string line;
    string userInput;
    string theName = "Clarissa";
    ofstream write2File;

    cout << "Welcome to the User Input Program!" << endl
         << endl;
    cout << "In this program, we will be asking for your input to add to our file." << endl;
    cout << "We will then reverse the text and output it to a new file. :)" << endl
         << endl;

    cout << "Please input a line to add to our file!" << endl;
    getline(cin, userInput);

    write2File.open("CSC450_CT5_mod5.txt", ios::app);
    write2File << "\n"
               << userInput << "\n";
    write2File.close();

    cout << endl
         << "\"" << userInput << "\" has successfully been added to our file." << endl;
    cout << "We will now reverse all of the characters in the file.";

    ofstream reversedFile("CSC450-mod5-reverse.txt");
    ifstream readFile("CSC450_CT5_mod5.txt");
    while (readFile)
```


File Edit Selection View Go Run Terminal Help
userInput.cpp - Module 5 - Visual Studio Code
EXPLORER
MODULE 5
.vscode
CSC450_CT5_mod5.txt
userInput.cpp
userInput.exe
userInput.cpp
userInput.cpp > ...
33
34 cout << endl
35 << "\"" << userInput << "\" has successfully been added to our file." << endl;
36 cout << "We will now reverse all of the characters in the file.";
37
38 ofstream reversedFile("CSC450-mod5-reverse.txt");
39 ifstream readFile("CSC450_CT5_mod5.txt");
40 while (readFile)
41 {
42 getline(readFile, line);
43 reverseWords(line);
44 reversedFile << line << endl;
45 }
46
47 reversedFile.close();
48 readFile.close();
49
50 cout << endl
51 << endl
52 << "The text has been reversed and output to a file called CSC450-mod5-reverse.txt!" << endl;
53 cout << "Enjoy your new file. :)" << endl
54 << "Goodbye.";
55
56 return 0;
57 }
58
OUTLINE
reverseWords(strin...
main()
TIMELINE
Ln 58, Col 1
Spaces: 4
UTF-8
CRLF
C++
Win32
Prettier


File Edit Selection View Go Run Terminal Help
userInput.cpp - Module 5 - Visual Studio Code
EXPLORER
MODULE 5
.vscode
CSC450_CT5_mod5.txt
CSC450-mod5-revers...
userInput.cpp
userInput.exe
userInput.cpp
userInput.cpp > ...
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
void reverseWords(string &line)
{
int n = line.length();
for (int i = 0; i < n / 2; i++)
swap(line[i], line[n - i - 1]);
}
int main()
{
string line;
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell https://aka.ms/pscore6
PS C:\Users\Jinyume\Documents\School\CSU Global\CSC450 Programming III\C++\Module 5> cd "c:\Users\Jinyume\Documents\School\CSU Global\CSC450 Programming III\C++\Module 5\" ; if ($?) { g++ userInput.cpp -o userInput } ; if ($?) { .\userInput }
Welcome to the User Input Program!
In this program, we will be asking for your input to add to our file.
We will then reverse the text and output it to a new file. :)
Please input a line to add to our file!
Hey now, you're an all star! Get your game on! Go play!
"Hey now, you're an all star! Get your game on! Go play!" has successfully been added to our file.
We will now reverse all of the characters in the file.
The text has been reversed and output to a file called CSC450-mod5-reverse.txt!
Enjoy your new file. :)
Goodbye.
PS C:\Users\Jinyume\Documents\School\CSU Global\CSC450 Programming III\C++\Module 5>
OUTLINE
reverseWords(strin...
main()
TIMELINE
Ln 58, Col 1 Spaces: 4 UTF-8 CRLF C++ Win32 Prettier

CSC450_CT5_mod5.txt - Notepad

File Edit Format View Help

Please be sure to append your data to this text file.

If these first three lines are deleted, then your program is not functioning as expected.

Hey now, you're an all star! Get your game on! Go play!

Ln 1, Col 1 100% Windows (CRLF) UTF-8

CSC450-mod5-reverse.txt - Notepad

File Edit Format View Help

```
.elif txet siht ot atad ruoy dneppa ot erus eb esaelP

.detcepxe sa gninoitcnuf ton si margorp ruoy neht ,deteled era senil eerht tsrif eseht fI

!yalp oG !no emag ruoy teG !rats lla na er'uoy ,won yeH
```

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

Clarissa Kuter

CSC450 Programming III

Reginald Haseltine

17 July 2022

Employee Salary Calculator Analysis

For the critical thinking assignment due this week, I was assigned with creating a C++ program that would once again obtain input from a user. The input would then be appended to the end of a file provided by the professor without altering the text already in it. In this case, the file was CSC450_CT5_mod5.txt. From here, there were two options I could have tried. The first was to then create a reversal method that would have the program read the text in the file and then reverse all of it. It would then output the reversed text to a new file titled CSC450_mod5_reverse.txt. The other option was to instead write a method that would search the file for my name. If it was found, the program would return the name, but if it wasn't, the program would instead return the message "name not found" to a text file titled CSC450-not-found.txt.

When I first pursued this assignment, I tried tackling Option #2; however, I was unable to get my program to perform the second task properly. While I could get it to search the file and return the name when found, I could not get it return the other message when the name was not found. Instead, if the name was not in the file, the program would return both the name and the name not found responses, or, more commonly, it would simply not return a message and then

the program would cease to terminate. Despite hours of tinkering, I was unable to find the solution to this crucial error I kept stumbling into; as such I eventually reverted to Option #1.

This error was its own lesson on the importance of file input and output safety. Going into the assignment, I knew that it was important to always ensure I close any file I open after its task has been performed. Failure to close a file could allow an attacker to exhaust the system's resources; it could also allow data written into in-memory file buffers to not be cleared if an abnormal program termination occurred. In my Option #2 attempt, I wrote for the file to be closed after the name was found or not found; however, the program was unable to reach that statement whenever the latter option occurred. That opened my first attempt of the program to many security errors. For some cases, it seemed as though the program never reached the end of file while reading the lines in it, and if left eternally reading the file, that could also open the program up to security vulnerabilities.

While I didn't toy with it too much in either rendition of my program, another thing to be wary of when outputting strings to or from a file is the habit of not making a call to `std::basic_filebuf::seekoff()` either directly following an output to stream or when outputting to stream right after receiving input from stream. Failure to do so can result in undefined behavior from the program; therefore, learning to utilize that call is important when mitigating security risks in one's program.