**Module 5 Critical Thinking Assignment**

Alex Stampfl

Colorado State University Global

CSC450 Programming III

Farhad Bari

February 15th, 2025

**Pseudocode**

1. Create an output file stream (outputFile) with file path
   1. Use append mode (std::ios:app) so new content is added instead of overwriting.
2. If outputFile opens successfully:
   1. Initialize userInput variable
   2. Display “Enter text” prompt to the user in the terminal
3. Use a do-while loop to collect user input:
   1. Read input (getline(cin, userInput) )
   2. If userInput is not “done”, write it to outputFile
   3. Repeat until user types “done”
4. Close output file
5. Invoke reverse() method
6. Inside reverse() method:
   1. Open input file (inputFile) for reading
   2. If inputFile doesn’t exist, show error message and exit function
   3. Read entire file content into a string (fileContent)
   4. Close inputFile
7. If `fileContent` is empty, display a warning message and exit function
8. Reverse the file content using `std::reverse` (#include <algorithm>)
9. Open a new output file (CSC450-mod5-reverse.txt) for writing (overwrite mode)
10. Write reversed content to `CSC450-mod5-reverse.txt`
11. Close file

**Source Code**

#include <iostream>

#include <string>

#include <fstream>

#include <algorithm> // for reverse

void reverse();

int main() {

    // create i/o file variable

    std::ofstream outputFile("CSC450\_CT5\_mod5.txt", std::ios::app); // oepn file in 'append' mode via flag `std::ios::app`

    // check if open

    if (outputFile.is\_open()) {

        std::string userInput;

        std::cout << "Enter text to append to file (type 'done' to finish):\n";

        // write to file while userInput is not done

        do {

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

            if (userInput != "done") {

                outputFile << userInput << std::endl;

            }

        } while (userInput != "done");

        outputFile.close();

        std::cout << "Data written successfully to file.\n";

    } else {

        std::cerr << "Unable to open file for writing.\n";

        return 1;

    }

    reverse();

    return 0; // end program

}

// Reverse characters in external text file

void reverse() {

    std::ifstream inputFile("CSC450\_CT5\_mod5.txt");

    if (!inputFile) {

        std::cerr << "Error: Unable to open file.\n";

        return;

    }

    // Read entire file into string

    std::string fileContent, line;

    while (std::getline(inputFile, line)) {

        fileContent += line + "\n";

    }

    inputFile.close();

// Check if file is empty

    if (fileContent.empty()) {

        std::cerr << "Warning: Input file is empty. Nothing to reverse.\n";

        return;

    }

    // reverse string

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

    // Open output file for writing

    std::ofstream outputFile("CSC450-mod5-reverse.txt"); // Create new reverse file

    if (!outputFile) {

        std::cerr << "Error: Unable to open output file.\n";

        return;

    }

    // Write reverse content to output file

    outputFile << fileContent;

    outputFile.close();

    std::cout << "Reversed content appended to output file.";

}

**Screenshots (output)**

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A black screen with white text

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Git Repository**

A screenshot of a computer

AI-generated content may be incorrect.