String Input Console Application

Stephan Peters

Colorado State University Global

CSC450-1 23WD: Programming III

Reginald Haseltine

25 February 2024

String Input Console Application

For this assignment, the students were asked to create a console application in C++ that gets two strings from the user, concatenates them, and prints the output to the console. This process should be repeated three times.

The Console Application

The program main() program repeats the concatenate() function three times, then exits. The concatenate() function declares two strings for user input, uses getln(cin, *string*) to store the user input, then uses string.append() to concatenate with a space delimiter, then uses printf() to output the concatenated string to the console.

Source Code 1

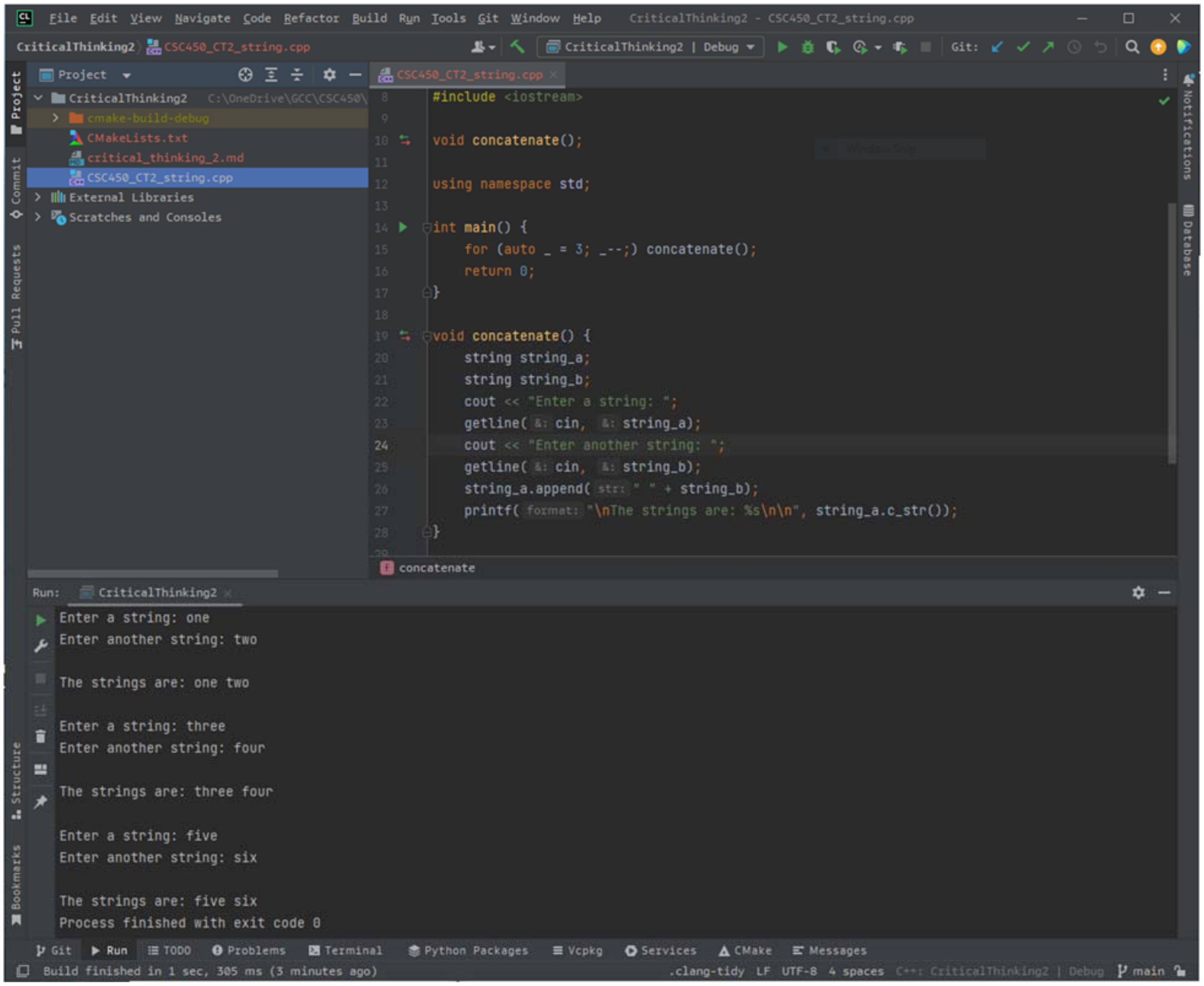
Source code for CSC450_CT2_string.cpp.

```cpp
/*
 * A simple C++ program that takes two string inputs from a user,
 * concatenates the two strings,
 * then prints the resulting output to the screen.
 * (CSC450_CT2_string.cpp)
 */

#include <iostream>

void concatenate();

using namespace std;

int main() {
    for (auto _ = 3; _--;) concatenate();
    return 0;
}

void concatenate() {
    string string_a;
    string string_b;
    cout << "Enter a string: ";
    getline(cin, string_a);
    cout << "Enter another string: ";
    getline(cin, string_b);
    string_a.append(" " + string_b);
    printf("\nThe strings are: %s\n\n", string_a.c_str());
}
```

Figure 1 shows the console application running after compilation in the CLion IDE.

Figure 1

Running the CSC450_CT2_strings.cpp program



```cpp
#include <iostream>

void concatenate();

using namespace std;

int main() {
    for (auto _ = 3; _--;) concatenate();
    return 0;
}

void concatenate() {
    string string_a;
    string string_b;
    cout << "Enter a string: ";
    getline( &: cin,  &: string_a);
    cout << "Enter another string: ";
    getline( &: cin,  &: string_b);
    string_a.append( str: " " + string_b);
    printf( format: "\nThe strings are: %s\n\n", string_a.c_str());
}
```

```
Enter a string: one
Enter another string: two

The strings are: one two

Enter a string: three
Enter another string: four

The strings are: three four

Enter a string: five
Enter another string: six

The strings are: five six
Process finished with exit code 0
```

Build finished in 1 sec, 305 ms (3 minutes ago)

Vulnerability Assessment

This console app uses strings as variables, not character arrays so it should not be vulnerable to common buffer overflow attacks. By not restricting the input length, a malicious user *could* enter a string so large it would consume the entire memory allocated to the execution of the program, in essence forcing a Denial of Service (DoS) for concurrent users of the software or machine, an assessment of this would need to take place before deployment. This program also uses printf, which can be incorrectly coded in a way that introduces the Format String Vulnerability (CWE 134) (Du, 2014), but the printf function in this code is written correctly, and is compliant with security standards.

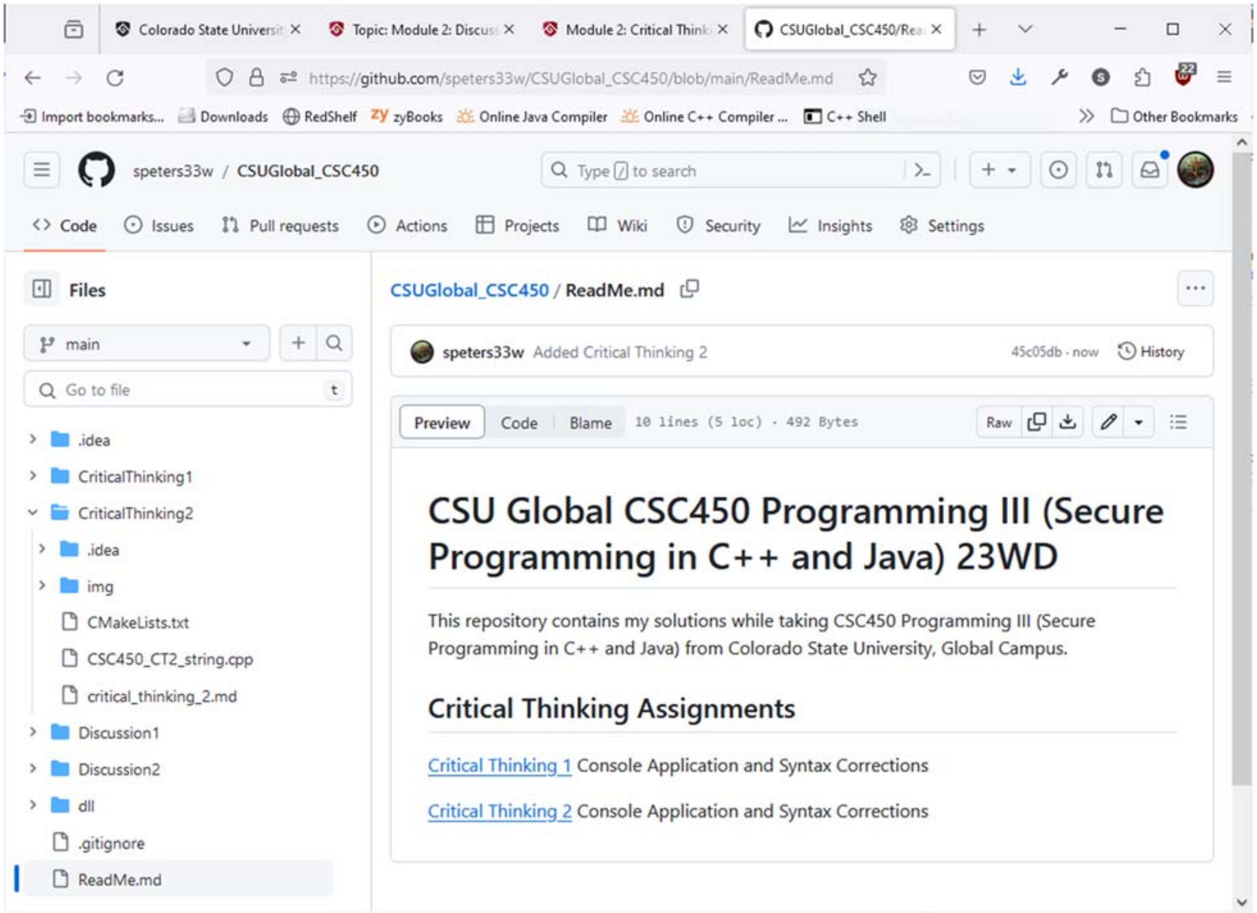
GitHub Repository

In addition to the solutions above, I was tasked to create a GitHub repository for the project. This repository is located at https://github.com/speters33w/CSUGlobal_CSC450/tree/main/CriticalThinking2

Figure 4 shows a screenshot of the main page of this repository.

Figure 4

Image of the main page of my CSC 450 GitHub repository



CSU Global CSC450 Programming III (Secure Programming in C++ and Java) 23WD

This repository contains my solutions while taking CSC450 Programming III (Secure Programming in C++ and Java) from Colorado State University, Global Campus.

Critical Thinking Assignments

Critical Thinking 1 Console Application and Syntax Corrections

Critical Thinking 2 Console Application and Syntax Corrections

References

Ballman, A. (2017). *SEI CERT C++ Coding Standard Rules for Developing Safe, Reliable, and Secure Systems in C++ 2016 Edition* (p. 205). https://resources.sei.cmu.edu/downloads/secure-coding/assets/sei-cert-cpp-coding-standard-2016-v01.pdf

Du, W. (Kevin). (2014). Format String Vulnerability printf (user input). In *Department of Electrical Engineering and Computer Science, Syracuse University*. https://web.ecs.syr.edu/~wedu/Teaching/cis643/LectureNotes_New/Format_String.pdf

Mitre. (2023, June 29). *CWE - CWE-134: Use of Externally-Controlled Format String (4.6)*. Cwe.mitre.org. https://cwe.mitre.org/data/definitions/134.html