Console Application and Syntax Corrections

Stephan Peters

Colorado State University Global

CSC450-1 23WD: Programming III

Reginald Haseltine

18 February 2024

Console Application and Syntax Corrections

For this assignment, the students were asked to create a console application in C++ that returns general address information for an individual, and correct syntax in two additional applications to compile and run them. This paper will show the source code for all these applications, and screenshots of them running.

Print Information for a Fictional Person

This console application will ask a user for a name and address and store them in variables to output a full address after the information is entered. I was reminded of an old Infocom game, written by Douglas Adams in 1987 named *Bureaucracy* that begins with a terrible and frustrating form that adds random disparaging remarks about the user's name and place of residence as the user fills out the form. The game itself may be played online at myabandonware.com, it begins with the form. I decided to add these comments from the form into my code as the user is asked to enter the information, just to make the project a little less boring. I retrieved the comments from a historical archive of the source code.

Source Code 1

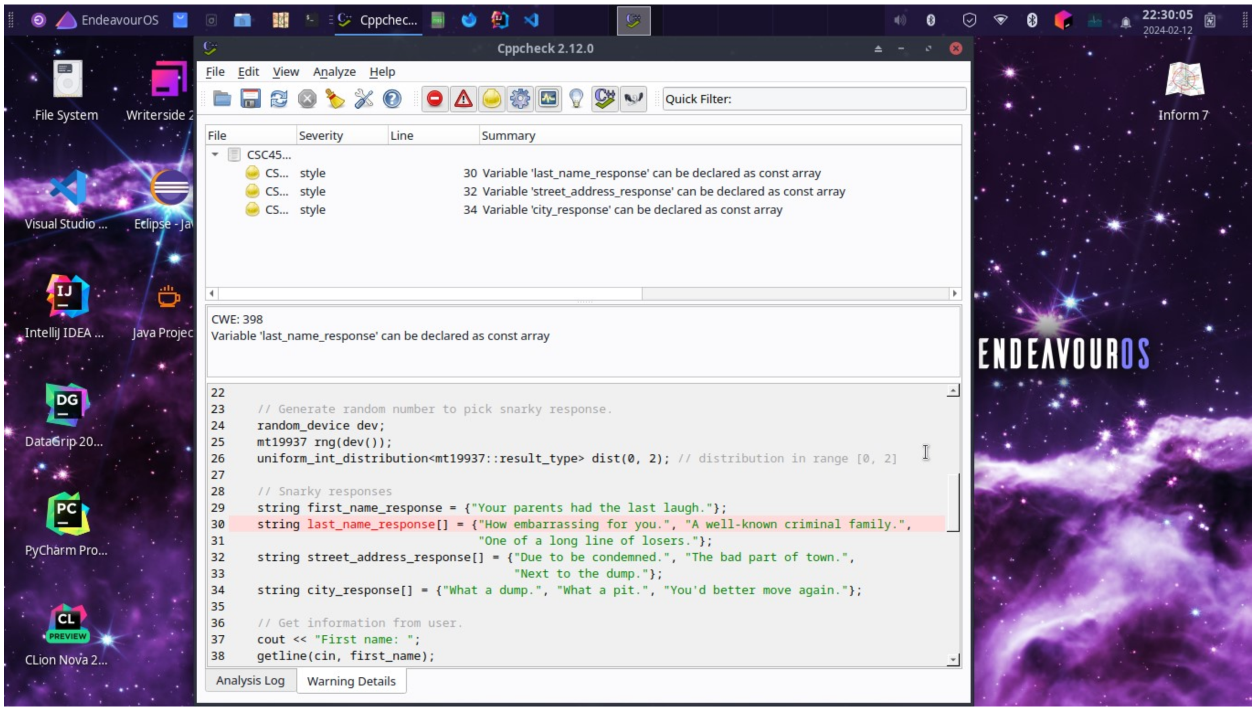
Source code for CSC450_CT1_person.cpp.

```
/*
 * A simple C++ console application that prints information
 * for a fictional person.
 * Returns the snarky responses from the bank form
 * in the Infocom Text Adventure "Bureaucracy".
 * (CSC450_CT1_person.cpp)
 */

#include <iostream>
#include <random>
#include <string>
#include <conio.h>

// Standard namespace declaration
using namespace std;

// Main Function
int main() {
    string first_name;
    string last_name;
    string street_address;
    string city;
    string zip;

    // Generate random number to pick snarky response.
    random_device dev;
    mt19937 rng(dev());
    uniform_int_distribution<mt19937::result_type> dist(0, 2);

    // Snarky responses
    const string first_name_response = {"Your parents had the last laugh."};
    const string last_name_response[] = {"How embarrassing for you.",
                                         "A well-known criminal family.",
                                         "One of a long line of losers."};
    const string street_address_response[] = {"Due to be condemned.",
                                              "The bad part of town.",
                                              "Next to the dump."};
    const string city_response[] = {"What a dump.",
                                    "What a pit.",
                                    "You'd better move again."};

    // Get information from user.
    cout << "First name: ";
    getline(cin, first_name);
    cout << first_name_response << endl << endl;

    cout << "Last name: ";
    getline(cin, last_name);
    cout << last_name_response[dist(rng)] << endl << endl;

    cout << "Street address: ";
    getline(cin, street_address);
    cout << street_address_response[dist(rng)] << endl << endl;

    cout << "City: ";
    getline(cin, city);
    cout << city_response[dist(rng)] << endl << endl;

    cout << "Zip Code: ";
    getline(cin, zip);
    cout << endl << endl;

    // Print information to screen
    cout << first_name << last_name << endl;
    cout << street_address << endl;
    cout << city << ", " << zip << endl << endl;

    cout << "Welcome to Bureaucracy" << endl
         << "A Paranoid Fantasy" << endl;

    getch();

    return 0;
}
```

While I was concerned getline() might introduce buffer overflow security issues as defined in STR52-CPP (Ballman, 2017 pp 205-207), I found no references to this introducing vulnerabilities. I checked the code using cppcheck and found the only complaint about this code was I had not declared the string and string arrays for the snarky responses as constants (see figure 1).

Figure 1

Checking CSC450_CT1_person,cpp for vulnerabilities using cppcheck..



File	Severity	Line	Summary
CSC45...			
CS...	style	30	Variable 'last_name_response' can be declared as const array
CS...	style	32	Variable 'street_address_response' can be declared as const array
CS...	style	34	Variable 'city_response' can be declared as const array

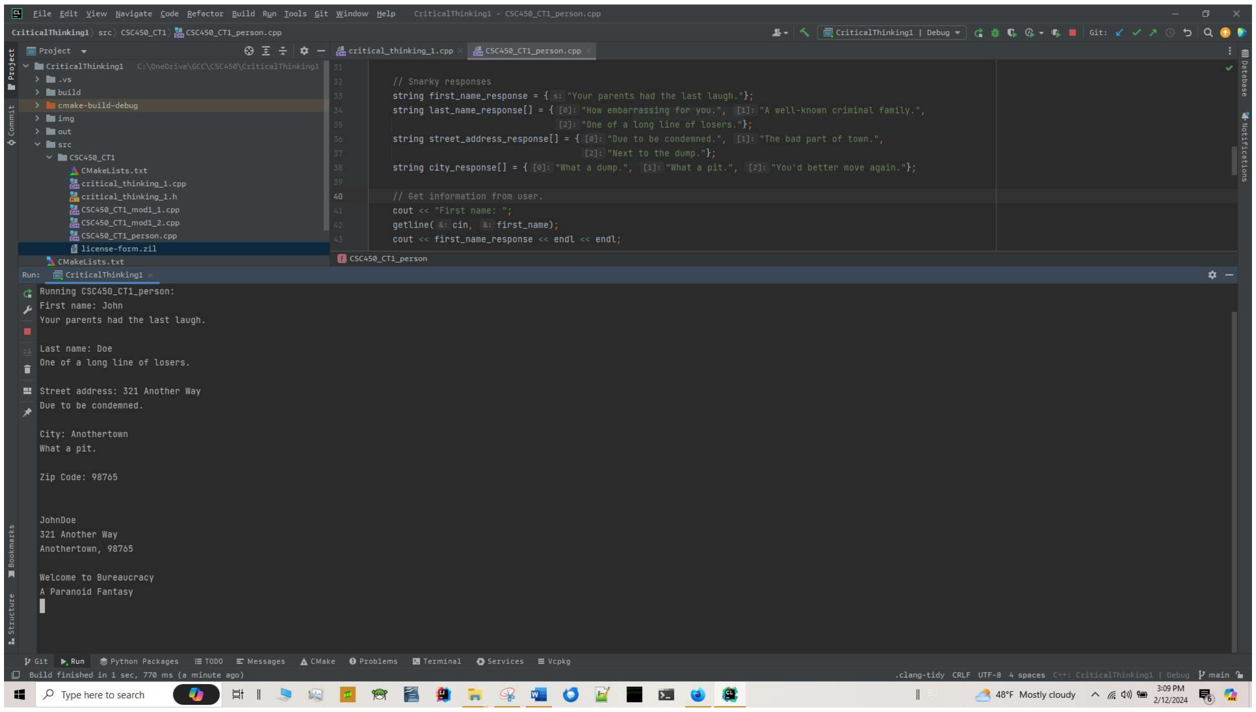
CWE: 398
Variable 'last_name_response' can be declared as const array

```
22
23    // Generate random number to pick snarky response.
24    random_device dev;
25    mt19937 rng(dev());
26    uniform_int_distribution<mt19937::result_type> dist(0, 2); // distribution in range [0, 2]
27
28    // Snarky responses
29    string first_name_response = {"Your parents had the last laugh."};
30    string last_name_response[] = {"How embarrassing for you.", "A well-known criminal family.",
31                                   "One of a long line of losers."};
32    string street_address_response[] = {"Due to be condemned.", "The bad part of town.",
33                                        "Next to the dump."};
34    string city_response[] = {"What a dump.", "What a pit.", "You'd better move again."};
35
36    // Get information from user.
37    cout << "First name: ";
38    getline(cin, first_name);
```

Figure 2 shows a screenshot of the program running in the CLion IDE. A full-resolution image can be found at my CSC450 GitHub Repository.

Figure 2

Running CSC450_CT1_person.exe in the JetBrains CLion IDE.



```cpp
    // Snarky responses
    string first_name_response = { s: "Your parents had the last laugh."};
    string last_name_response[] = { [0]: "How embarrassing for you.", [1]: "A well-known criminal family.",
                                    [2]: "One of a long line of losers."};
    string street_address_response[] = { [0]: "Due to be condemned.", [1]: "The bad part of town.",
                                         [2]: "Next to the dump."};
    string city_response[] = { [0]: "What a dump.", [1]: "What a pit.", [2]: "You'd better move again."};

    // Get information from user.
    cout << "First name: ";
    getline( &: cin, &: first_name);
    cout << first_name_response << endl << endl;
```

```
Running CSC450_CT1_person:
First name: John
Your parents had the last laugh.

Last name: Doe
One of a long line of losers.

Street address: 321 Another Way
Due to be condemned.

City: Anothertown
What a pit.

Zip Code: 98765


JohnDoe
321 Another Way
Anothertown, 98765

Welcome to Bureaucracy
A Paranoid Fantasy
```

```
Build finished in 1 sec, 770 ms (a minute ago)
```

Correct Syntax in a C++ Program

In this exercise, I was asked to correct the syntax on a C++ program in order to compile and run it. There were various errors, such as a missing close to the header comment, missing quotes, missing lines, comments not preceded by a comment symbol, etc. The program in its original, broken form may be viewed in critical_thinking_1.md at my GitHub repository for this course.

Source Code 2

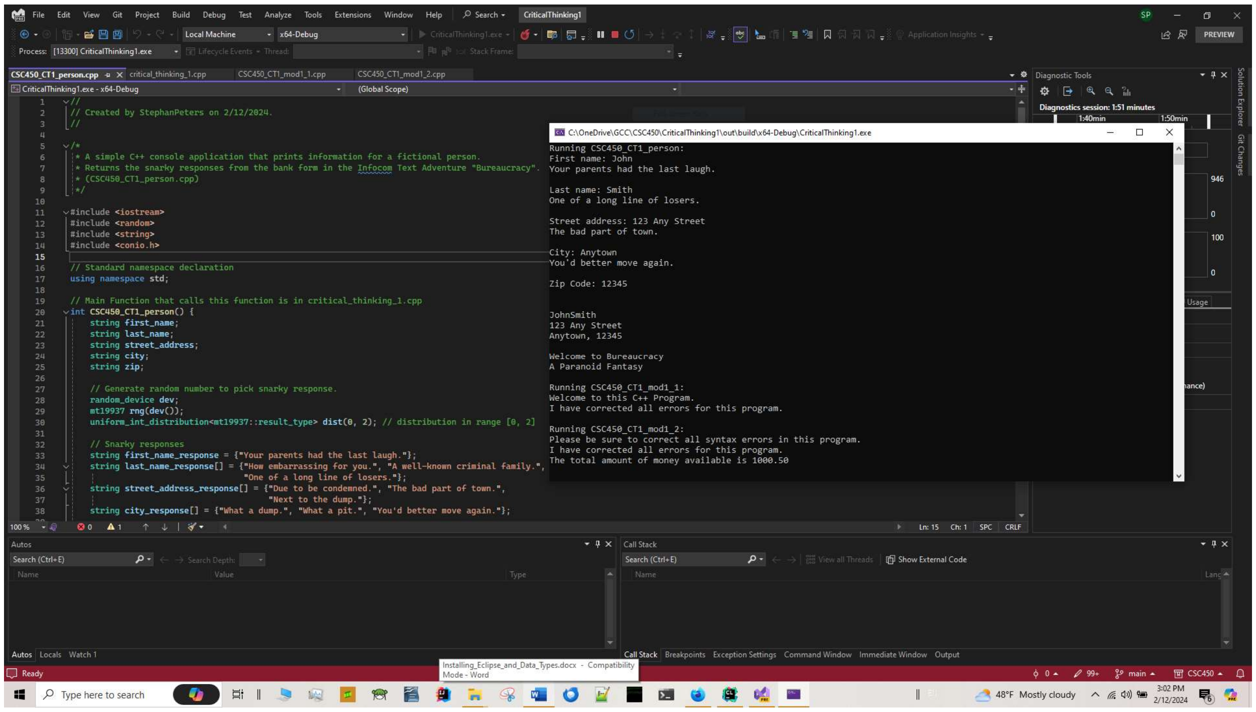
Corrected source code for CSC450_CT1_mod1_1.cpp.

```
/*
 * Simple Program with a few Errors corrected.
 * (CSC450_CT1_mod1_1.cpp)
 */

#include <iostream>
#include <conio.h>

// Standard namespace declaration
using namespace std;

// Main Function
int main() {

    //Standard Output Statement
    cout << "Welcome to this C++ Program." << endl;

    cout << "I have corrected all errors for this program." << endl;

    // Wait For Output Screen
    getch();

    // Function return Statement
    return 0;

}
```

Figure 3 shows the output of this code, as well as the second program corrected in this project running in the Microsoft Visual Studio 2022 Preview. This image may also be viewed in full resolution at my CSC450 GitHub Repository.

Figure 3

Running CSC450_CT1_mod1_1.cpp and CSC450_CT1_mod1_2.cpp in the Microsoft Visual Studio 2022 Preview IDE.



CSC450_CT1_person.cpp
critical_thinking_1.cpp
CSC450_CT1_mod1_1.cpp
CSC450_CT1_mod1_2.cpp
CriticalThinking1.exe - x64-Debug
(Global Scope)
Diagnostic Tools
Diagnostics session: 1:51 minutes
C:\OneDrive\GCC\CSC450\CriticalThinking1\out\build\x64-Debug\CriticalThinking1.exe

```cpp
//
// Created by StephanPeters on 2/12/2024.
//

/*
 * A simple C++ console application that prints information for a fictional person.
 * Returns the snarky responses from the bank form in the Infocom Text Adventure "Bureaucracy".
 * (CSC450_CT1_person.cpp)
 */

#include <iostream>
#include <random>
#include <string>
#include <conio.h>

// Standard namespace declaration
using namespace std;

// Main Function that calls this function is in critical_thinking_1.cpp
int CSC450_CT1_person() {
    string first_name;
    string last_name;
    string street_address;
    string city;
    string zip;

    // Generate random number to pick snarky response.
    random_device dev;
    mt19937 rng(dev());
    uniform_int_distribution<mt19937::result_type> dist(0, 2); // distribution in range [0, 2]

    // Snarky responses
    string first_name_response = {"Your parents had the last laugh."};
    string last_name_response[] = {"How embarrassing for you.", "A well-known criminal family.",
                                   "One of a long line of losers."};
    string street_address_response[] = {"Due to be condemned.", "The bad part of town.",
                                        "Next to the dump."};
    string city_response[] = {"What a dump.", "What a pit.", "You'd better move again."};
```

```
Running CSC450_CT1_person:
First name: John
Your parents had the last laugh.

Last name: Smith
One of a long line of losers.

Street address: 123 Any Street
The bad part of town.

City: Anytown
You'd better move again.

Zip Code: 12345


JohnSmith
123 Any Street
Anytown, 12345

Welcome to Bureaucracy
A Paranoid Fantasy

Running CSC450_CT1_mod1_1:
Welcome to this C++ Program.
I have corrected all errors for this program.

Running CSC450_CT1_mod1_2:
Please be sure to correct all syntax errors in this program.
I have corrected all errors for this program.
The total amount of money available is 1000.50
```

Correct Syntax in a Second C++ Program

For this exercise, I was asked to correct the syntax on an additional C++ program in order to compile and run it. This program in its original, broken form may also be viewed in critical_thinking_1.md at my GitHub repository for this course.

Source Code 3

Corrected source code for CSC450_CT1_mod1_2.cpp.

```
/*
 * Simple Program with a few Errors corrected.
 * (CSC450_CT1_mod1_2.cpp)
 */

#include <iostream>
#include <iomanip>
#include <conio.h>

// Standard namespace declaration
using namespace std;

// Main Function
int main() {

    //this should be printed out
    double myMoney = 1000.50;


    // Standard Output Statement
    cout << "Please be sure to correct all syntax errors in this program."
<< endl;

    cout << "I have corrected all errors for this program." << endl;

    cout << "The total amount of money available is ";
    cout << setprecision(2) << fixed << "$" << myMoney << endl;

    // Wait for Output Screen
    getch();

    // Function return Statement
    return 0;
}
```

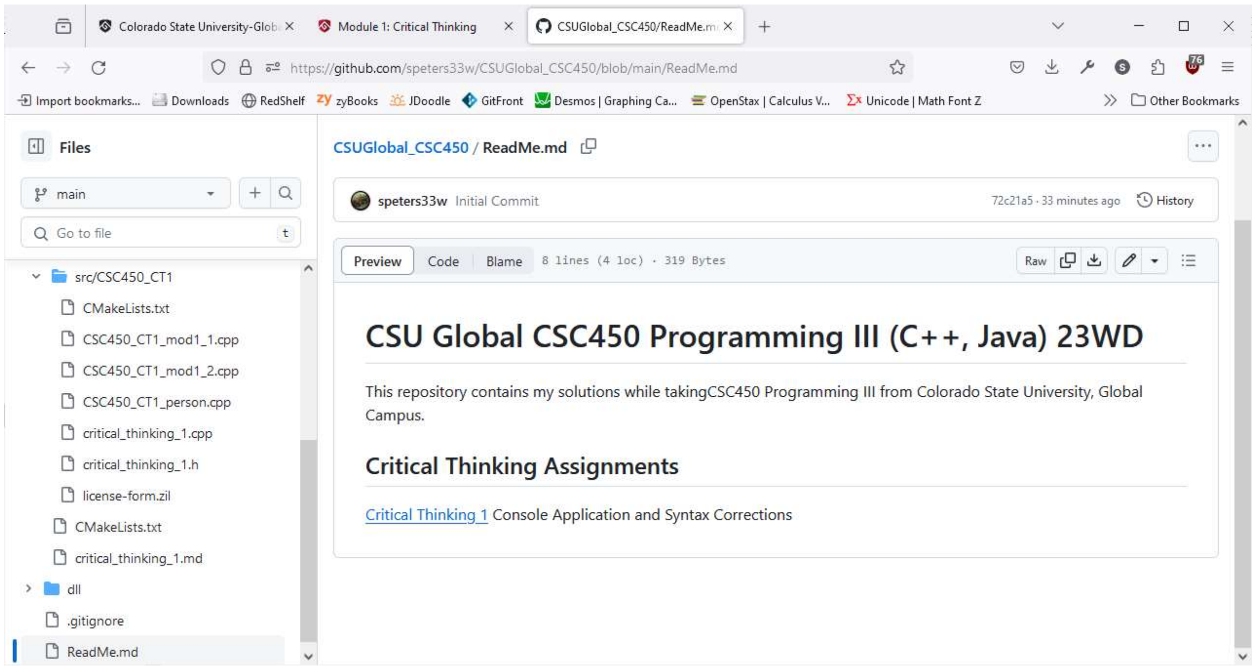
GitHub Repository

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/CriticalThinking1

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


Colorado State University-Glob
Module 1: Critical Thinking
CSUGlobal_CSC450/ReadMe.m
https://github.com/speters33w/CSUGlobal_CSC450/blob/main/ReadMe.md
Import bookmarks...
Downloads
RedShelf
zyBooks
JDoodle
GitFront
Desmos | Graphing Ca...
OpenStax | Calculus V...
Unicode | Math Font Z
Other Bookmarks
Files
main
Go to file
t
CSUGlobal_CSC450 / ReadMe.md
speters33w Initial Commit
72c21a5 · 33 minutes ago
History
Preview
Code
Blame
8 lines (4 loc) · 319 Bytes
Raw

```
src/CSC450_CT1
    CMakeLists.txt
    CSC450_CT1_mod1_1.cpp
    CSC450_CT1_mod1_2.cpp
    CSC450_CT1_person.cpp
    critical_thinking_1.cpp
    critical_thinking_1.h
    license-form.zil
  CMakeLists.txt
  critical_thinking_1.md
dll
.gitignore
ReadMe.md
```

CSU Global CSC450 Programming III (C++, Java) 23WD

This repository contains my solutions while takingCSC450 Programming III from Colorado State University, Global Campus.

Critical Thinking Assignments

Critical Thinking 1 Console Application and Syntax Corrections

Conclusion

As in other programming languages, precise syntax is required to run or compile an application from code. I found IDEs with preconfigured CMake build environments (CLion, Visual Studio) are much more user friendly than those which need manual configuration, and these manual configuration requirements change frequently from version to version. I found writing the *person* application using the comments from Bureaucracy made this project much more enjoyable. I think I'll break out Windows Frotz and see if my blood pressure goes up.

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

Cerfeuil. (2024, February 5). *Bureaucracy*. IFDB. https://ifdb.org/viewgame?id=zjyxds3s57pgis3x

Cppcheck team. (n.d.). Cppcheck manual. In *SourceForge.io*. Retrieved February 12, 2024, from https://cppcheck.sourceforge.io/manual.pdf

Kinder, D. (n.d.). *Windows Frotz*. Davidkinder.co.uk. Retrieved February 13, 2024, from https://davidkinder.co.uk/frotz.html

MyAbandonware. (n.d.). *Bureaucracy*. My Abandonware. Retrieved February 13, 2024, from https://www.myabandonware.com/game/bureaucracy-a9/play-a9

Proudfoot, A. (2019, June 22). *bureaucracy/forms.zil at develop · the-infocom-files/bureaucracy*. GitHub. https://github.com/the-infocom-files/bureaucracy/blob/develop/forms.zil