**Critical Thinking Assignment Module One Assignment**

**Console Application String Input**

Victor Enogwe

Computer Science, Colorado State University - Global Campus

CSC450-1: Programming III

Reginald Haseltine

April 25, 2024

In this module, we’ve looked at the string data type in C++ and learned about the vulnerabilities associated with manipulating strings in C++ and the rules and recommendations in the SEI CERT C++ coding standards.

I created a simple C++ console program that collects a set of user inputs (2) thrice, concatenates each set and prints the result.

**Program Pseudocode**:

PROGRAM User String Input Collector  
- This program collects two string user inputs(thrice) from a user, concatenates them and displays the concatenation.  
  
BEGIN  
 BEGIN  
 create a main.cpp file.  
  
 declare a "str\_tolower" function that takes a string and returns the lowercase version. Make sure the initial string is not mutated.  
 declare a "str\_trim" function that takes a string and returns the string with white space removed from both ends. Make sure the initial string is not mutated.  
 declare a "get\_string\_input" function that prompts a user to enter a string and returns the string.  
 The function takes two arguments: a string "instruction" to tell the user what to enter, and an integer "max\_length" to restrict the input length.  
 Use the "str\_trim" function to trim the string before return.  
 Validate the input string against the maximum length parameter.  
 Ensure that the trimmed input is not empty.  
 declare a "get\_two\_string\_input" function that uses the "get\_string\_input" function to collect user input twice.  
 The function should take a "max\_length" parameter to limit the allowable max input length.  
 END  
  
 declare a main function to run the program.  
 inside the main function using the "get\_two\_string\_input" function, collect the user input thrice.  
 Print out each collected input.  
END

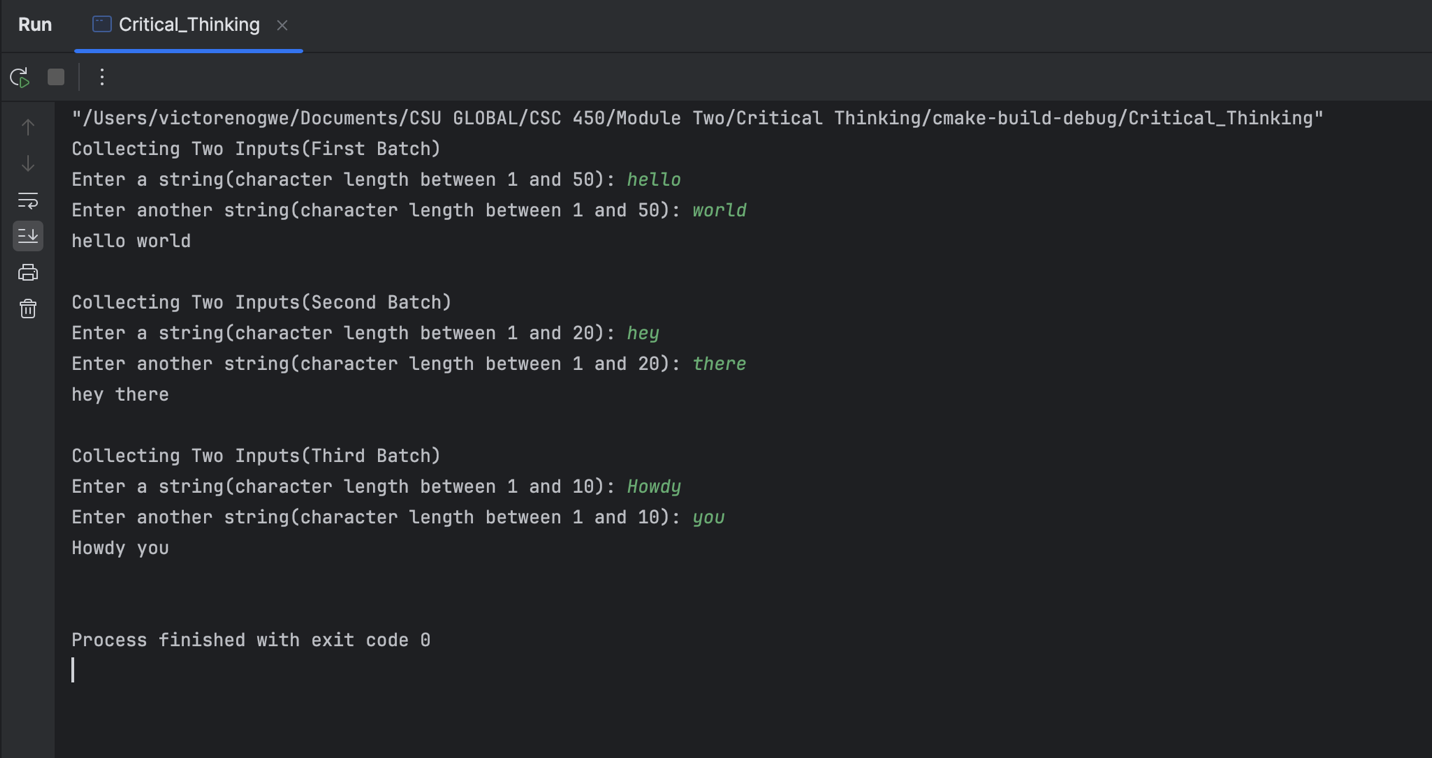
**Main.cpp:**

/\*  
 \* Program: Create a simple C++ console application  
 \* that will write a program that will take two string inputs from a user.  
 \* Your program should concatenate the two strings and then print the resulting output to the screen.  
 \* Take the two string inputs from the user 3 times for varying string lengths.  
 \*/  
  
#include <iostream>  
#include <string>  
#include <regex>  
  
using namespace std::regex\_constants;  
  
using std::cout;  
using std::cin;  
using std::getline;  
using std::endl;  
using std::string;  
using std::to\_string;  
using std::tolower;  
using std::transform;  
using std::regex;  
using std::regex\_replace;  
using std::regex\_match;  
using std::runtime\_error;  
  
/\*  
 \* str\_tolower  
 \* converts a string to lowercase  
 \* avoids mutating original string  
 \* returns the lowercase string  
 \*/  
string str\_tolower(const string& value)  
{  
 string s;  
  
 s.assign(value);  
  
 std::transform(s.begin(), s.end(), s.begin(),  
 [](unsigned char c){ return std::tolower(c); }  
 );  
  
 return s;  
}  
  
/\*  
 \* str\_trim  
 \* trims the leading and trailing whitespaces in a string  
 \* avoids mutating original string  
 \* returns the trimmed string  
 \*/  
string str\_trim(const string& value) {  
 string s;  
  
 s.assign(value);  
  
 return regex\_replace(  
 regex\_replace(s, regex( "^\\s+$" ), ""),  
 regex( "\\s+$" ),  
 ""  
 );  
}  
  
/\*  
 \* get\_string\_input  
 \* asks a user to enter a string  
 \* recursive  
 \*  
 \* returns the input string  
 \*/  
string get\_string\_input(const string& instruction, int max\_length = 50) {  
 try {  
 cout << instruction;  
  
 string input;  
  
 getline(cin, input);  
  
 string trimmed\_input = str\_trim(input);  
  
 if (trimmed\_input.empty()) {  
 throw runtime\_error(string("Input length must be greater than 1\n"));  
 } else if (trimmed\_input.length() > max\_length) {  
 throw runtime\_error(string("Input length must be less than " + to\_string(max\_length) + "\n"));  
 }  
  
 return trimmed\_input;  
 } catch (const runtime\_error& error) {  
 cout << error.what();  
  
 return get\_string\_input(instruction);  
 }  
}  
  
/\*  
 \* get\_two\_string\_input  
 \* collects a string input twice using the get\_string\_input method  
 \*  
 \* returns a space separated concatenation of the two user input  
 \*/  
string get\_two\_string\_input(int max\_length = 50) {  
 string max\_length\_string = to\_string(max\_length);  
 string first\_input(get\_string\_input("Enter a string(character length between 1 and " + max\_length\_string + "): ", max\_length));  
 string second\_input(get\_string\_input("Enter another string(character length between 1 and " + max\_length\_string + "): ", max\_length));  
  
 return first\_input + " " + second\_input + "\n";  
}  
  
int main() {  
 cout << "Collecting Two Inputs(First Batch)" << endl;  
  
 string first\_batch = get\_two\_string\_input();  
  
 cout << first\_batch << endl;  
  
 cout << "Collecting Two Inputs(Second Batch)" << endl;  
  
 string second\_batch = get\_two\_string\_input(20);  
  
 cout << second\_batch << endl;  
  
 cout << "Collecting Two Inputs(Third Batch)" << endl;  
  
 string third\_batch = get\_two\_string\_input(10);  
  
 cout << third\_batch << endl;  
  
 return 0;  
}

**Git Repository Image: Git Branch = Main**

**https://github.com/victor-csu/****CSC450-Module-Two-Critical-Thinking/tree/main**

**Happy Path Execution Screenshot – Fictional Person - CSC450\_CT1\_mod1-0-execution-output:**



References