**Buffer Overflow Prevention Summary**

In this project, the goal was to fix a buffer overflow problem in a C++ program. A buffer overflow happens when too much data is entered, causing it to overwrite other important information in memory. In our case, this could lead to the account number being changed when the user enters more than 19 characters. The task was to prevent this from happening and to notify the user if they enter too many characters.

Originally, the program allowed users to enter more than 20 characters, which caused the account number to be changed. To fix this, we needed to limit the input to a safe number of characters and notify the user if they entered too much.

**Code Changes and Explanation:**

1. **Changed the input method:** Instead of using std::cin >>, which doesn't protect against buffer overflow, we switched to std::cin.get(). This method ensures that only up to 19 characters are taken in, leaving room for the necessary null terminator that ends the string.
2. **Checked for input overflow:** After the input is gathered, the program checks if the user entered more than 19 characters. If they did, the extra characters are cleared from the input buffer, and a warning is shown to the user.

**Updated Code:**

#include <iomanip>

#include <iostream>

#include <cstring> // For safer string manipulation

int main()

{

std::cout << "Buffer Overflow Example" << std::endl;

// Constant account number that should not be modified

const std::string account\_number = "CharlieBrown42";

// Create a buffer for user input

char user\_input[20];

// Notify the user and request input

std::cout << "Enter a value (max 19 characters): ";

// Get user input safely using 'std::cin.get()' to prevent buffer overflow

std::cin.get(user\_input, sizeof(user\_input)); // Safely get up to 19 characters

// Check if the user entered too many characters (if input buffer is full)

if (std::cin.gcount() == sizeof(user\_input) - 1 && std::cin.peek() != '\n') {

std::cin.ignore(10000, '\n'); // Clear the input buffer

std::cout << "Warning: You entered too many characters! Input truncated." << std::endl;

}

// Output the user input and the account number

std::cout << "You entered: " << user\_input << std::endl;

std::cout << "Account Number = " << account\_number << std::endl;

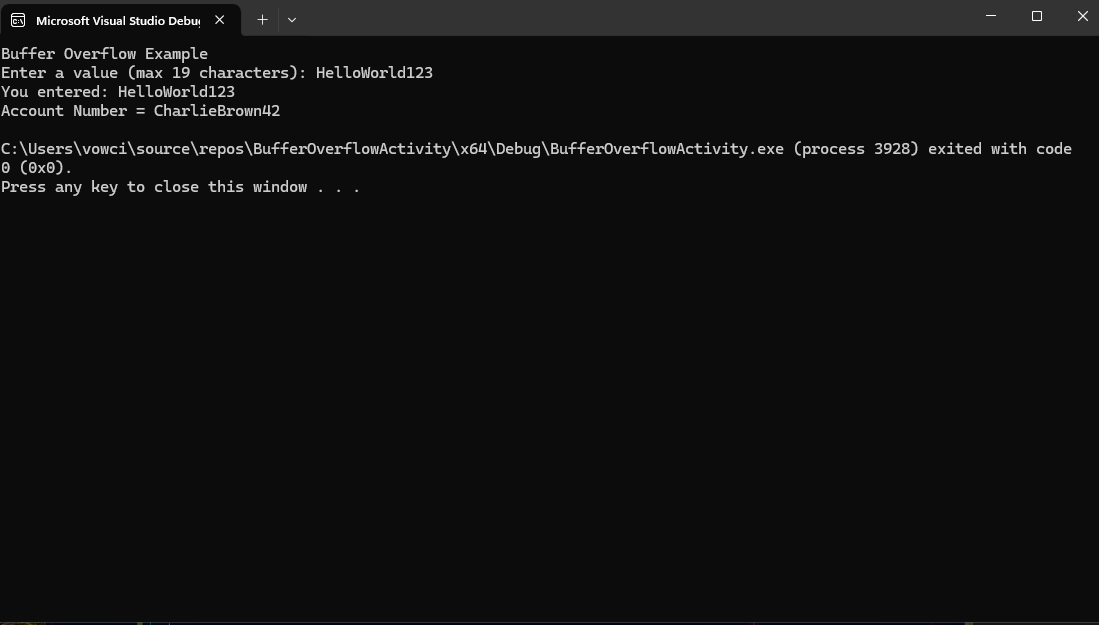
}

**What the Code Does:**

* It gets input from the user and limits it to 19 characters.
* If the user enters more than 19 characters, the extra ones are thrown away, and the program shows a warning.
* The account number is printed, and it stays safe from being changed.

**Screenshot Explanations:**

1. **Screenshot 1:**
   * Shows the output when the user enters "HelloWorld123", which is less than 19 characters. The input is displayed correctly, and the account number stays the same.



1. **Screenshot 2:**
   * Shows what happens when the user enters "Helloworld12345678", which is exactly 19 characters. It fits within the limit, and the program works without any issues.

A screenshot of a computer

Description automatically generated

1. **Screenshot 3:**
   * Shows a successful build process.

A screenshot of a computer program

Description automatically generated