Joseph Les

CS 405 Professor Hodde

Buffer Overflow

Upon execution, the program prompts the user to enter a value, utilizing the std::cin.getline function to read the input. This function ensures that the input does not exceed the predefined limit of Max\_Input - 1 characters, thereby preventing overflow within the user\_input buffer. The program then checks if the input operation failed due to excessive input length by evaluating the state of the input stream using std::cin.fail() and std::cin.eof(). If an overflow is detected, the program clears the error flag and ignores the rest of the input buffer, subsequently displaying an error message to inform the user.

If the input is within the acceptable range, the program proceeds to output the entered value. It concludes by displaying the predefined account number and a message indicating the end of the program. This example effectively demonstrates best practices for input handling in C++ by ensuring that user inputs are managed safely, thereby avoiding potential security risks associated with buffer overflows.

A screenshot of a computer program

Description automatically generated