**Logo, company name

Description automatically generated**

Template based on the Centers for Medicare & Medicaid Services, Information Security & Privacy Management’s Assessment

**Security Assessment Report**

Version 5.0

April 30, 2023

Table of Contents

[1. Summary 3](#_Toc133754398)

[1. Assessment Scope 3](#_Toc133754399)

[2. Summary of Findings 3](#_Toc133754400)

[3. Summary of Recommendations 5](#_Toc133754401)

[2. Goals, Findings, and Recommendations 6](#_Toc133754402)

[1. Assessment Goals 6](#_Toc133754403)

[2. Detailed Findings 6](#_Toc133754404)

[3. Recommendations 6](#_Toc133754405)

[3. Methodology for the Security Control Assessment 7](#_Toc133754406)

[4. Figures and Code 9](#_Toc133754407)

[4.1.1 Process or Data flow of System. 9](#_Toc133754408)

[4.1.2 Other figure of code 11](#_Toc133754409)

[5. Works Cited 12](#_Toc133754410)

# Summary

As the project was hastily made there are many faults in the project that could cause issues later. To that end this security assessment was made to showcase these issues and provide solutions to some of them.

## Assessment Scope

* GitHub: GitHub is an online repository that I used to help store and organize the project.
* Clion: Clion is the IDE that was used to code the project.
* Draw.io: Draw.io is a diagram making application that was used to make the diagram of the project.
* Manpower limitation: As only one person is coding the project fixes and new iterations of the project take longer then with a normal project.

## Summary of Findings

Of the findings discovered during our assessment, 1 was considered High risks, 2 Moderate risks, 1 Low, and 0 Informational risks. The SWOT used for planning the assessment are broken down as shown in Figure 2.

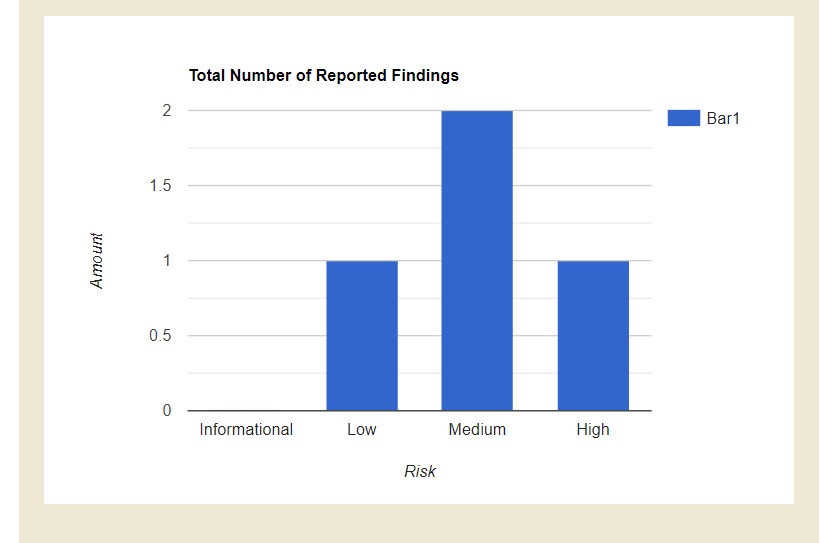


Figure 1. Findings by Risk Level

A picture containing table

Description automatically generated

Figure 2. SWOT

Over the course of the Assessment, I discovered four main security issues with the project. The first being that the project had no way to check user input causing it to crash if an incorrect input was entered. The second being that the project never saved user input after it stopped running causing the user to lose all data previously calculated. Thirdly the code was inefficient and needed to be condensed. Finally, no unit testing was used in the project, so any new functions were immediately entered into the project, sometimes causing issues within the project.

## Summary of Recommendations

The code can now check for user input without crashing and a sperate unit testing program has been made to test new functions before they are entered into the main program. While good progress has been made there are still some glaring issues that need to be fixed. Mainly the program does not save a user’s input and many parts of the code could be condensed to make the project more efficient.

# Goals, Findings, and Recommendations

## Assessment Goals

The purpose of this assessment was to do the following:

* Ensure that the system follows the initial assignment’s instructions.
* Determine if the project was securely stored.
* Find if the project had any use outside of its respective assignment.
* Clarify coding issues the project has.
* Create solutions for the coding issues mentioned earlier.

## Detailed Findings

[Security assessment link](#_Process_or_Data)

* Vulnerability 1: Data is lost from the program every time the project is stopped making it impossible to see old calculations.
* Vulnerability 2: The program will crash if a letter is typed into any shape measurement such as width.
* Vulnerability 3: If too large of a number is entered into the project at any point it can crash as it will be unable to interpret the data.
* Vulnerability 4: The project would be easy to take as it is housed in one file.
* Vulnerability 5: As the project is only housed on one device everything could be lost if the device’s hard drive is damaged.
* Vulnerability 6: The code can be lost in the files as it is vaguely named shape project.
* Vulnerability 7: The project could be edited by any user resulting in potential crashes and errors.
* Vulnerability 8: The comments in the project let intruders understand the projects functions and how to modify them.

## Recommendations

[Security assessment link](#_Process_or_Data)

* Vulnerability fix for 1, 2, 3: A user input function was created that checks if the user’s input is correct and prevents the user from inputting an incorrect value into the project.
* Vulnerability fix for 4, 5, 6: The projects files were renamed shape project files and three copies were made 1 on a flash drive, 1 on my computer and 1 on GitHub.
* Vulnerability fix for 7, 8: the files are now password locked and the comments were edited so that they explain less why the code works and more how the code works to keep the overall descriptions of functions vague.

# Methodology for the Security Control Assessment

**3.1.1 Risk Level Assessment (delete this text: you don’t have to change 3.1.1)**

Each Business Risk has been assigned a Risk Level value of High, Moderate, or Low. The rating is an assessment of the priority with which each Business Risk will be viewed. The definitions in Table 1 apply to risk level assessment values (based on probability and severity of risk). While Table 2 describes the estimation values used for a risk’s “ease-of-fix”.

Table - Risk Values

| Rating | Definition of Risk Rating |
| --- | --- |
| High Risk | Exploitation of technical or procedural vulnerability will cause substantial harm to the business processes. Significant political, financial, and legal damage is likely to result |
| Moderate Risk | Exploitation of the technical or procedural vulnerability will significantly impact the confidentiality, integrity and/or availability of the system, or data. Exploitation of the vulnerability may cause moderate financial loss or public embarrassment to organization. |
| Low Risk | Exploitation of the technical or procedural vulnerability will cause minimal impact to operations. The confidentiality, integrity and availability of sensitive information are not at risk of compromise. Exploitation of the vulnerability may cause slight financial loss or public embarrassment |
| Informational | An “Informational” finding, is a risk that has been identified during this assessment which is reassigned to another Major Application (MA) or General Support System (GSS). As these already exist or are handled by a different department, the informational finding will simply be noted as it is not the responsibility of this group to create a Corrective Action Plan. |
| Observations | An observation risk will need to be “watched” as it may arise because of various changes raising it to a higher risk category. However, until and unless the change happens it remains a low risk. |

Table - Ease of Fix Definitions

| Rating | Definition of Risk Rating |
| --- | --- |
| Easy | The corrective action(s) can be completed quickly with minimal resources, and without causing disruption to the system or data |
| Moderately Difficult | Remediation efforts will likely cause a noticeable service disruption.   * A vendor patch or major configuration change may be required to close the vulnerability. * An upgrade to a different version of the software may be required to address the impact severity. * The system may require reconfiguration to mitigate the threat exposure. * Corrective action may require construction or significant alterations to the way business is undertaken |
| Very Difficult | The high risk of substantial service disruption makes it impractical to complete the corrective action for mission critical systems without careful scheduling.   * An obscure, hard-to-find vendor patch may be required to close the vulnerability. * Significant, time-consuming configuration changes may be required to address the threat exposure or impact severity. * Corrective action requires major construction or redesign of an entire business process |
| No Known Fix | No known solution to the problem currently exists. The Risk may require the Business Owner to:   * Discontinue use of the software or protocol * Isolate the information system within the enterprise, thereby eliminating reliance on the system.   In some cases, the vulnerability is due to a design-level flaw that cannot be resolved through the application of vendor patches or the reconfiguration of the system. If the system is critical and must be used to support on-going business functions, no less than quarterly monitoring shall be conducted by the Business Owner, and reviewed by IS Management, to validate that security incidents have not occurred |

[Security assessment link](#_Process_or_Data)

All the risks identified were accurate to the project but were more unlikely to occur such as the code being converted into another language or someone illegally gaining access to the project. For the first one there is no reason to convert it to another language unless access is lost to all C++ interpreters and as the project does not store data there would be nothing for an intruder to steal except the project itself which on its own has little value.

**3.1.2 Tests and Analyses**

Black and White box testing were used to test the project,

White box testing is when a tester has full or partial access to the source code of what their testing. The test usually also has some knowledge of the code and is testing the inner workings of the code.

Black box testing is when the tester has no access to the code and usually no knowledge of the code’s inner workings, this is done to test the code’s functionality.

**3.1.3 Tools**

Clion and GitHub were used to test and create the project,

Clion is the C++ interpreter that was used to create, run, and test the project.

GitHub is an online repository that was used to store old versions of the project in case of an emergency.

Draw.io is a diagram application I used to make diagrams for the project.

# Figures and Code

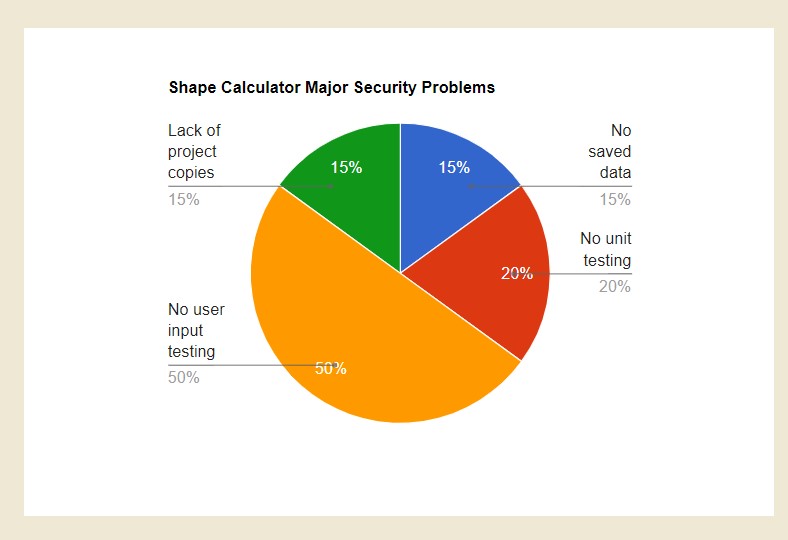
### Process or Data flow of System.

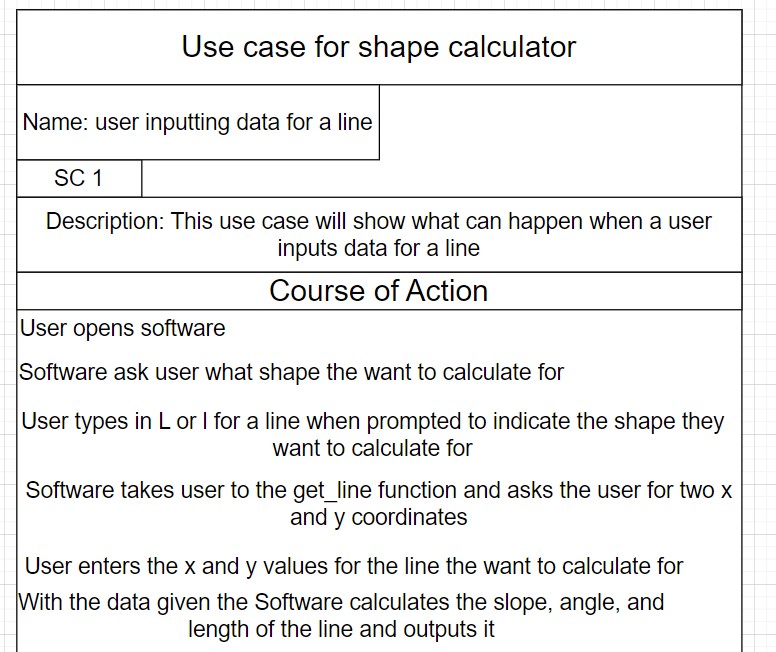
Graphical user interface, application

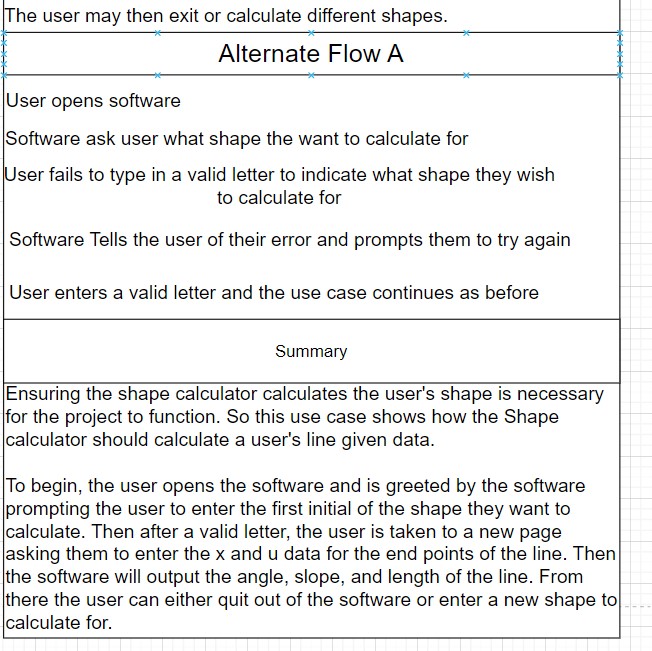
Description automatically generated

A picture containing table

Description automatically generated

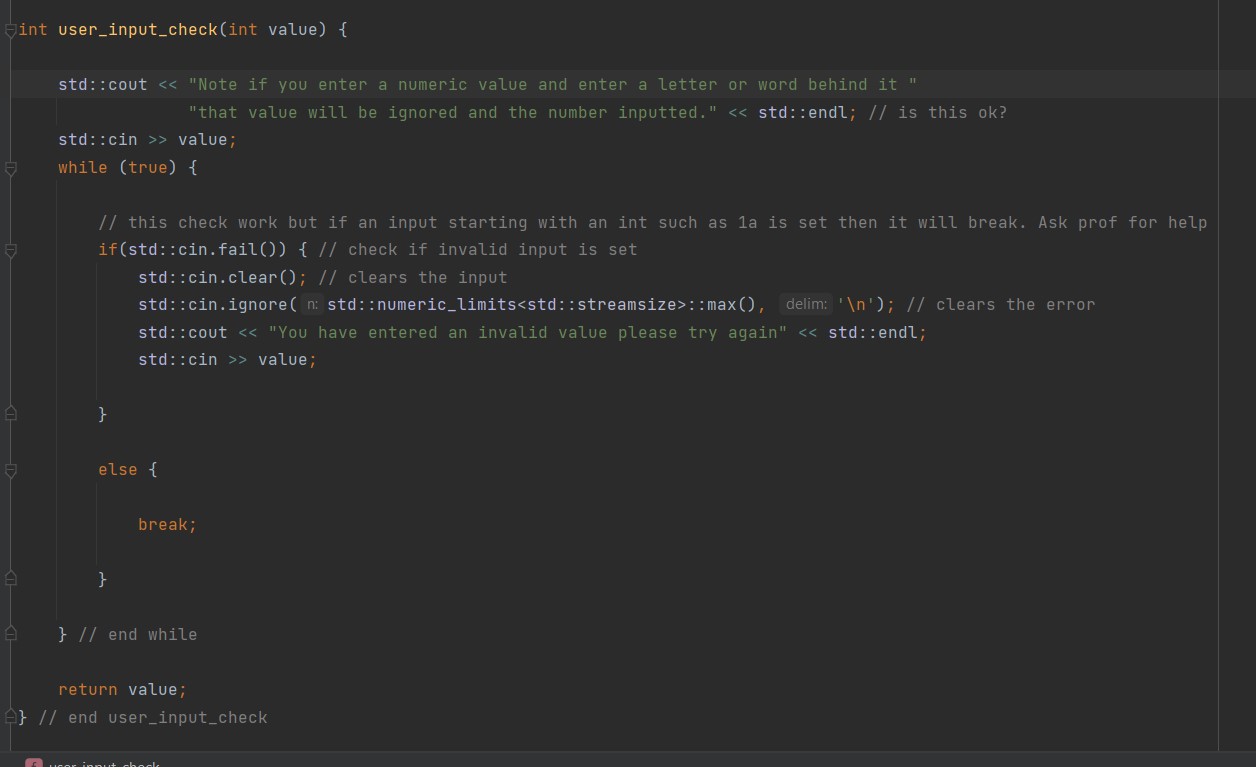


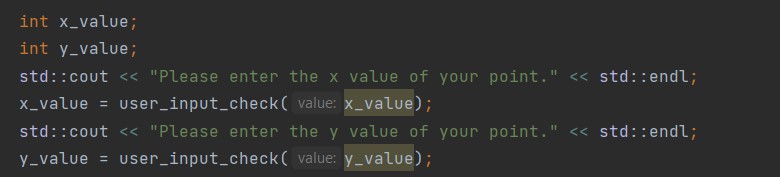




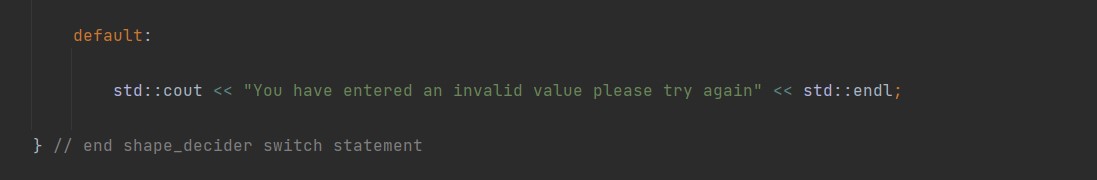
### Other figure of code

User input function:

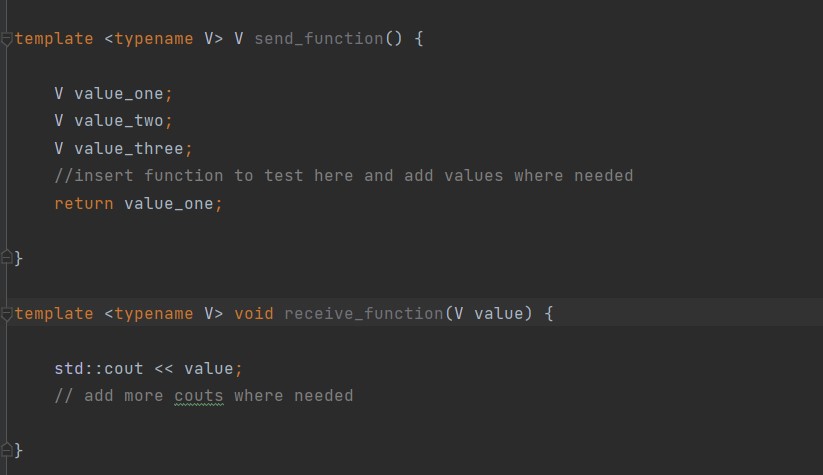


User input function called same across all functions

Default value in switch statement



The Unit testing functions:



# Works Cited

Swebok edited by Bourque Pierre and Dick E. Richard

Lesson on White and black box testing by Josiah Greenwell for course CEN 3078

Lesson on vulnerability management by Josiah Greenwell for course CEN 3078

TryHackMe lesson 2 “Active Reconnaissance” <https://tryhackme.com/room/activerecon>.

Lesson on Security Basics-Risk by Josiah Greenwell for course CEN 3078

Access control Discussion on Canvas for course CEN 3078