# CS 405 Project Two Script Template

Complete this template by replacing the bracketed text with the relevant information.

[**https://youtu.be/PMcuivH0FfU**](https://youtu.be/PMcuivH0FfU)

| **Slide Number** | **Narrative** |
| --- | --- |
| **1** | Hello, my name is Jillian Ortiz and this is my final presentation for Green Pace. |
| **2** | My policy outlines the security principles with the C/C++ coding language. We are going to discuss the three A’s, authorization, authentication, and auditing. We’ll dive into encryption and explore unit testing as well. These standards will give us a road map to how we apply defense in depth to create a secure system by layering defense across different attack vectors. |
| **3** | This shows our threat matrix and will help guide us. When designing our security, we should prioritize standards that fall into the likely and high-priority boxes. Automation tools like Astree and CPPCheck can be used to complete static code analysis to help us catch and vulnerabilities. |
| **4** | The table shows the ten secure coding principles on the left and on the right we can see the coding standards that apply to the principles. (read the principles).  Many of the principles have multiple standards listed and on the next slide, we’ll look more in-depth into those coding standards. |
| **5** | Here we have all ten of our coding standards listed from the highest priority at the top to the lowest priority at the bottom. This was based on the threat matrix we reviewed earlier along with other factors like remediation costs.  (read standards) |
| **6** | It's important to think about encryption in all its phases. It’s important to not only encrypt data when it's stored and being accessed or used like we normally think about data attacks happening but also encrypt data when it's in transit or in flight.   * Encryption at rest protects stored data, including hard drives, phones, computers, and cloud assets. * Encryption in flight is the process of securing data while it is moving. This usually happens between two devices within a network or while the data is transferring outside of a network. Examples of this are email encryption, DLP (Data Loss Prevention) solutions, and security features like firewalls and authentication. * Encryption in use protects data that is being created, edited, or otherwise in use. To ensure data security, we have to have control and protection measures in place before the data is accessed. |
| **7** | The three A’s:   1. Authentication is verifying you are who you say you are. The system makes sure username and passwords match 2. Authorization controls the level of access a user has to the system. This is usually done by role-based access. A good practice here is to apply the principle of least privileges by only giving a user access based on their role to complete role-given tasks. This pairs well with default deny which denies access to all subsystems unless their role requires it. 3. Accounting means keeping track of activities when interacting with a system. So keeping system logs, timestamps, who is accessing what and allows us to monitor and investigate if we need to. |
| **8** | Is out of range is a unit test we ran regarding memory protection that verifies the standard out\_of\_range exception is thrown when calling the at function with an index that is out of bounds. This test was successful as you can see and passed. |
| **9** | Verify max size verifies that the max size is greater than or equal to the size for x amount of entries, and was successfully completed. |
| **10** | Does size Decrease is a unit test regarding memory protection that verifies resizing decreases the collection when removing from the collection. We can see this test was successful |
| **11** | Does Resize Increase is a unit test regarding Memory Protection: This verifies resizing increases the collection when adding to it. This test successfully passed as shown |
| **12** | This is the DevOps pipeline. DevOps ends up transforming to DevSecOps by integrating security measures into each step of the DevOps pipeline. In the beginning phases, threat modeling and security tool training are added into the “Access and Plan” phase. In both the “Design” and “Build” phases, IDE security concerns are addressed. Automation will use several external tools to check for security vulnerabilities, dependency checks, and policy compliance. |
| **13** | * In traditional software development practices, organizations typically add security measures for applications and infrastructure after the testing phase. In contrast, DevSecOps integrates security into the DevOps pipeline from the very beginning. This means that security is considered and implemented simultaneously as the code is written and the application is developed.   + CPPCheck is a static analysis tool used to check for runtime errors   + Google Test automated unit tests based on user-defined assertions |
| **14** | * When encountering a security threat we can either wait, or act immediately. This is usually decided based on key factors based on threat severity, remediation costs, and manpower or labor needed. * Immediate remedies usually mean containment by either shutting down the affected systems or subsystems, or stopping any unauthorized activity, removing all corrupted/compromised data, updating or resetting user credentials,, and continual monitoring of system entry/exit points.   **Risks of delayed reaction:**   * Open for more attacks * Further data loss * Harming reputation * Losing customer’s trust and business   **Benefits of immediate action:**   * Mitigating widespread damage * Preventing future threats * Maintaining consistent performance * Delivering quality customer service. |
| **15** | Encryption should be everywhere! Unprotected and unencrypted data is too easy to steal. It is like leaving your wallet unattended in a room full of thieves. Lack of encryption is huge mistake made by companies that often ends up costing them in the long run. Secure coding is not a one and done practice, however an ongoing practice of all systems because the threats of today are not always the threats of tomorrow. The cost of protection pales in comparison to the risk of exploitation |
| **16** | Taking the extra steps from the start, implementing Defense in Depth, and incorporating security testing into everyday practice will allow DevOps to be transformed into DevSecOps here at Green Pace. Adopting all standards to improve the security of systems and applications, is a must. |
| **17** | I’d like to thank all the resources who contributed to the development of this project. |