# CS 405 Project Two Script Template

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

| **Slide Number** | **Narrative** |
| --- | --- |
| **1** | Hi there, thanks everyone for joining me today. My name is Michael and I’ll be going over the new Security Policy for Green Pace today. |
| **2** | From what the research for this policy has shown is that one layer of defense is never enough for a full security ecosystem. As an example, monitoring of server interactions should always be taking place to watch for unexpected behavior, as well access controls being implemented to prevent unauthorized access. Stacking layers of defense helps ensure that the best effort to software security is being given. |
| **3** | The threats matrix identifies how the coding standards rank in terms of severity and probability. The description on the left further defines what each of these categories are. Further explanation of what these coding standards are will be discussed in the following slides. |
| **4** | The 10 security principles that this security policy is centered around can be seen on this slide. Their corresponding coding standards are highlighted beside them, note how some coding standards apply to multiple principles. |
| **5** | The name associated with each coding standard can be seen here. Within the security policy, a detailed explanation of what threats these standards bring can be found. |
| **6** | Encryption of user data should occur at all times. Rather it be at rest, in flight or in use, a user’s data should be protected. The use of hashing algorithms are used to obscure this data. |
| **7** | Aligning with some of the security principles located in this policy, a Triple-A approach should be utilized when tracking client-server interactions. By ensuring that we follow the principle of Least Privilege, many attacks can be mitigated at the user level. Every interaction with our software should be authenticated, authorized, and accounted for. |
| **8** | As the text suggests, the following four slides show unit tests being conducted based on a couple of the coding standards in this policy. The first unit test slide is of a positive test that checks user input type. The next unit test checks for a correct size of a container based on a user attempting to increase the size by 5. The third test is a negative one that ensures an exception is thrown when an out of range index is accessed. The fourth and final unit test slide is a negative test that ensures a length error exception is thrown when trying to resize larger than what is possible. |
| **9** | In this slide we can see the DevSecOps cycle. I recommend that we implement automation in the ‘build’ and ‘verify and test’ phases. The use of assertions should be toggled within production when not in use, as listed in the security policy. Dependency checkers should be utilized to make sure that all of our libraries are up to date, and that we are not unknowingly introducing vulnerabilities due to dependencies. |
| **10** | The two main tools I recommend to jumpstart the automation process is Google Test and Cppcheck. Google Test allows for togglable assertions that can easily be turned off for production code and Cppcheck can identify many more bugs in the code than a standard compiler can. |
| **11** | One could argue that the time it takes implementing unit tests could be better served implementing new features to software in a timely manner. However, I am of the belief that having a more secure program will entice customers over bug ridden programs. |
| **12** | My recommendations here detail my thought process moving forward now that the security policy has been established. Training and implementation are paramount to a successful revitalization of security within our company. |
| **13** | In conclusion, we should re-explore our legacy code and determine if vulnerabilities exist based on this new security policy. Integrate the security policy with all upcoming software and revise the coding standards as seen fit. |
| **14** | Thank you for your time. |