# CS 405 Project Two Script Template

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

| **Slide Number** | **Narrative** |
| --- | --- |
| **1** | This is my security policy presentation for Green Pace. My name is Josh Millan. |
| **2** | This model serves as an overview of the in-depth protection strategies we employ in our work to uphold a reliable standard for secure coding. Our goal is to consistently provide the best services possible in secure coding. |
| **3** | Standards for secure coding include varying degrees of vulnerability so that the effects of the standard can be assessed. Certain dangers are more likely to materialize in most circumstances than others, but they still need to be taken seriously and should receive the appropriate attention. There are low and high priority levels for both probable and improbable cases. |
| **4** | Secure coding is based on ten fundamental concepts. Never forget to double-check your input data, and pay attention to any compiler warnings you see when developing or troubleshooting. When in development, build and design with security in mind. Always make things straightforward but efficient. As a standard, deny access to systems by default. Follow the least privilege principle. To ensure the security of connections and transfers between other systems, sanitize data that is transmitted to them. You can never play defense too well, so always rehearse thoroughly. Additionally, keep in mind to consistently employ successful quality assurance strategies and make sure you choose an efficient and effective safe coding standard. |
| **5** | We comply with a few important coding standards here a Green Pace our top ten standards are:  1. Do not cast to an out-of-range enumeration value  2. Use valid references, pointers, and iterators to reference elements of a container  3. Do not attempt to create a std::string from a null pointer  4. Do not store already-owned pointer value in an unrelated smart pointer  5. Properly deallocate dynamically allocated resources  6. Use a static assertion to test the value of a constant expression  7. Handle all exceptions thrown before main() begins executing  8. Do not alternately input and output from a file stream without an intervening positioning call  9. Do not invoke virtual functions from constructors or destructors  10. Value returning functions must return a value from all exit paths |
| **6** | Here at Green Pace, our encryption rules are also a top priority. Because the data would be encrypted on disk, encryption at rest is intended to prevent hackers from obtaining unencrypted data. The process of encrypting data while it is traveling through the system is known as encryption in flight. Data that is both in motion and at rest is encrypted. |
| **7** | Our Triple-A policies are likewise held to a high standard. The process of determining and confirming a user's identification is known as authentication, and it starts with an A. After the user's identity has been verified, the second A is for authorization. Once a user has been verified and the system has been accessed, this establishes their degree of access. This can involve the user's ability to read, add, remove, or alter people, files, and databases on the system. Accounting receives the last A. This just records every action a user takes within the system so that they can be held responsible for everything they do when logged in. |
| **8** | In order to make sure that our code is safe and functional, we implement unit testing procedures early and frequently during the development process. For example, limiting the character count in a user-input string to avoid buffer overflow. |
| **9** | The DevSecOps pipeline is a secure development technique with an infrastructure built on effectively maintaining code security. It uses a full circle approach to enforce policy. |
| **10** | This system's structure is sound; my only advice would be to always practice defense in depth and to test frequently and early to find any weaknesses or holes that could allow us to identify defects and errors early on. |
| **11** | When coding, there is always a risk because there is no 100% secure solution. The success of this approach depends heavily on continual education since it is important to always presume that there are dangers and faults in the system and to be persistent in staying up to date with all of the common threats and prevention tactics of today. The advantage of this policy is that we make an effort to stay relevant and up to date in order to provide our clients with the best possible security. |
| **12** | To maintain the degree of security we provide, staying abreast of current security threats and developments is essential. To complete the task, we keep things straightforward but efficient. One of the numerous standards we follow to deliver high-quality service is the use of SQL injection techniques to stop attackers from altering the databases within your system. |
| **13** | Using the standards and guiding principles discussed in this presentation, we can infer that the majority of the significant subjects were addressed in order to illustrate our strategy for developing and maintaining a safe and effective programming. In order to protect security and privacy and ensure that all sensitive information is kept safe and secure for all parties, we have also implemented a zero-trust policy when it comes to accessing items both inside and outside of the company network. |