# CS 405 Project Two Script Template

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

| **Slide Number** | **Narrative** |
| --- | --- |
| **1** | Hello, I am Jacob Mousseau and today I am here to talk to you about the Green Pace Security Policy. |
| **2** | Our Security Policy is meant to help focus code and protection efforts in order to make attacks less successful. |
| **3** | Improper management of memory can lead to many potential leaks and issues, Injections are an easy form of attack, improper inputs can be exploited but this is less likely with experienced teams. |
| **4** | This is a table of the ten core principles and a short description of what they mean, these principles include: Validating Input Data and Sanitizing data sent to systems, as well as default deny and quality assurance techniques, and of course the principle of least privilege. |
| **5** | From Green, to Red, we have a list of the coding standards we discussed in our other project, where the color signals a general level of importance and related risk. Memory related issues all are on the higher end of this spectrum. |
| **6** | At Rest we need to prevent attackers from getting in by encrypting data while it is on the disk, and we should also do this whenever the data is being used or being transmitted, in short, we want to always keep our data encrypted! |
| **7** | We want to authenticate all inputs and use those inputs to authorize what a user can do making use of default deny and principle of least privilege, and tracking this information is key because we want to make sure they can’t access anything they should not be able to. |
| **8** | [Insert text.] |
| **9** | The DevSecOps Pipeline is the security integration of the software development life cycle, the pipeline is a set of practices that we are supposed to incorporate in to build, test, and deploy secured software in a more efficient manner and environment. |
| **10** | We can use tools like Klocwork in order to detect vulernabilities in our pipleline and also take note of compliance issues, there are many tools for tasks like this, and Klocwork is just one example, some others include CodeSonar, Helix QAC, and Parasoft. |
| **11** | Risks that are always apparent are potential memory issues or leaks that can lead to easy attacks on the code, SQL injections are another risk to be addressed, making sure everything that uses SQL is properly contained and written would nullify this risk and would benefit the written code. The long and short of it is that the code that’s written needs to be checked for security issues and whether everything gets encrypted, if it does not this is something that needs to happen now with the knowledge given, if these types of things are not addressed and then they get exploited by an insider or outsider, you only have yourselves to blame. |
| **12** | A lot of what we talked about optimizes stuff to avoid potential attacks, which is not an end all be all, these are just guidelines that will help any team secure themselves from base level threats, and using this information and knowledge of defense in depth they can use it in tandem with more layers to create a higher sense of protection. |
| **13** | Encrypt everything that you can in order to further protect your code.  Make sure SQL fields are properly protected as well so they don’t get abused.  Apply defense in depth to ensure extra security. |
| **14** | These are the references I used for putting together the presentation. |