# CS 405 Project Two Script Gibson

https://www.youtube.com/watch?v=nPd2wGYuOiA

| **Slide Number** | **Narrative** |
| --- | --- |
| **1** | Hello. I am Garrett Gibson and I am a developer for Green pace. Today, we’ll be discussing our security policy and how we plan to adopt it moving forward. |
| **2** | When developing, it is vital to consider security throughout the entire process rather than at just the end. That is the goal of our new security policy, coding principles, and coding standards. |
| **3** | In this threat matrix, we identify Likely as there being a good chance something will occur, Unlikely as there being a good chance something will NOT occur, Priority, something is urgent and needs to be addressed, and Low Priority, something that is not urgent and can be revisited later. |
| **4** | These are our 10 coding principles in our security policy. I’d like to highlight validating user input, keeping it simple, and effective QA techniques as especially vital to our secure coding approach. |
| **5** | \*Read the blurb at the bottom of slide\*  SQL Injection, assertions, and exceptions are at the top because they can really get us started in the right direction. SQL Injection is a highly utilized but highly preventable attack vector. Assertions and exceptions are standard parts of secure coding and should be implemented immediately. |
| **6** | \*Read these definitions on the slide\* |
| **7** | \*Read these descriptions briefly.\* |
| **8** | In this unit test, we aim to verify a collection increases when the size of the collection is increased. We assert that the collection is empty, then we resize it to 1. Lastly we assert that the size of the collection is now 1, verifying that it does increase. This test will pass. |
| **9** | In this unit test, we verify resizing decreases the collection size. We start off with adding 10 entries to the collection and asserting this as true. Then, we resize the collection to 1 and assert that the size is 1. We successfully test this when the size of the collection is 1. |
| **10** | Similarly to last test, we are asserting that the collection becomes 0. Again, we start with 10 entries, assert that it has 10 entries, then resize to 0 and assert that it becomes 0. |
| **11** | For this final unit test, we verify that a collection exists. We then clear the collection. Once cleared we assert that the collection is empty. This test proves that the collection was cleared and is now empty. |
| **12** | This is a graphic of how our DevSecOps pipeline functions. |
| **13** | \*Read/summarize what is on this slide\* |
| **14** | \*Read slide\* |
| **15** | \*Read slide\* |
| **16** | \*Conclude/Read Slide\* |