# CS 305 Module Two Written Assignment Template

## Instructions

Replace the bracketed text with the relevant information in your own words. If you choose to include images or supporting materials, make certain to insert them in all the relevant locations in the document.

**Areas of Security:** The application, a Spring-based API, includes two methods (number and greeting) returning Greeting-type objects. It's designed to receive user input, outputting array elements or customized greeting messages. Key focus areas for security would be Input Validation, API Security, Error Handling, and Data Encapsulation. Input Validation ensures user inputs are sanitized to prevent errors or malicious input. Securing API interactions safeguards against untrustworthy connections. Spring provides some error handling, but it needs to be generic to avoid revealing system details to potential hackers. Encapsulation, especially in methods, guards sensitive data, necessitating private variables and getter/setter methods.  
  
**Areas of Security Justification:** The prioritization of Input Validation is justified by the need to eliminate risks from malformed or malicious inputs, which can disrupt application functionality or lead to security breaches. API Security is crucial to ensure reliable and secure communication channels, particularly important in web applications handling sensitive data or user interactions. The focus on Error Handling is to prevent exposing system information through generic error messages, a common vector for security exploits. Encapsulation in software design is essential for protecting sensitive data within the application, limiting access and modifications to authorized methods only. This approach aligns with best practices for secure coding in web-based applications, particularly those using frameworks like Spring.

**Code Review Summary:** The Greeting class incorporates getter/setter methods, with parameters marked private. However, GreetingController doesn't fully utilize these. Inputs to the API aren't adequately sanitized. For instance, in the number method, user-supplied input directly influences output, posing an injection risk. Additionally, there's a lack of input validation against the static array's length. Error messages need obfuscation to prevent information leakage. The greeting method also requires input sanitization to avoid buffer overruns. Outdated dependencies in pom.xml suggest necessary updates to Spring Framework and Java JDK versions.

**Mitigation Plan:** Address potential security weaknesses by focusing on input sanitization and validation. Utilize the Greeting class's getter/setter methods for enhanced data encapsulation. Implement error message sanitization to avoid revealing program details like array length. Regular updates to Spring controllers and pom.xml are advised, alongside vulnerability checks for dependencies.