# CS 305 Module Five Coding Assignment Checksum Verification Template

## Instructions

Using the instructions from theModule Five Coding Assignment Checksum Verification Guidelines and Rubric, replace the bracketed text with the relevant information in your own words.

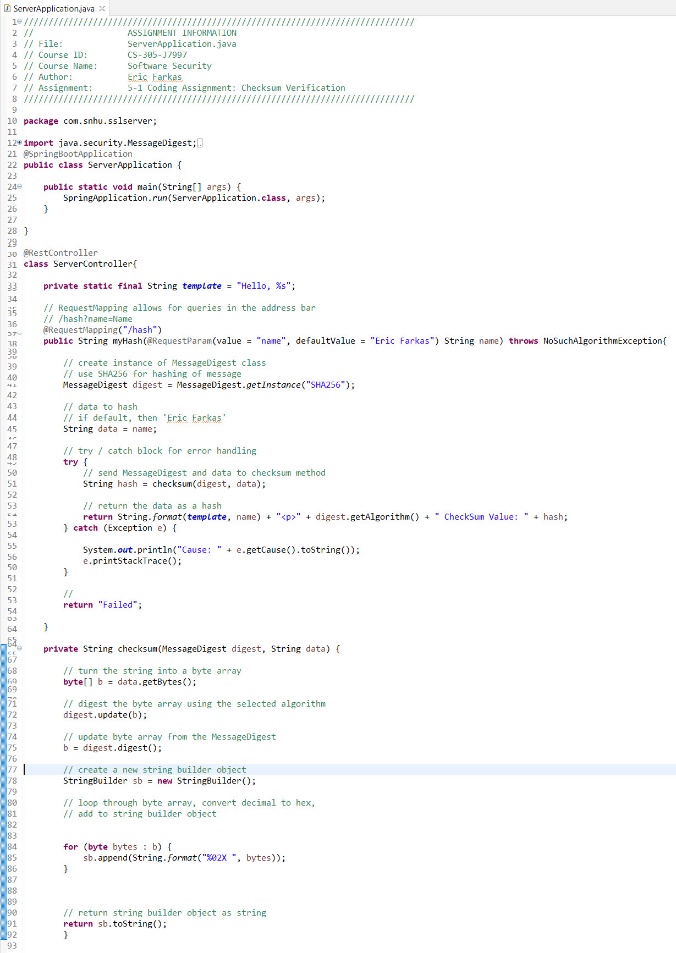
## Algorithm Cipher: Recommend an appropriate encryption algorithm cipher that avoids collisions.

I recommend using SHA-256 as the hashing algorithm for this application. There are 2256 possible values, making it nearly impossible for two entries to have the same hash value. Additionally, SHA-256 changes the hash value any time the entry is changed.

## Justification: Justify your reasoning for the recommended algorithm cipher that avoids collisions.

SHA-256 is an industry standard trusted by organizations required to protect sensitive data. It is lightweight from a processing perspective and has never been compromised. The likelihood of collisions is extremely low due to the number of possible values. SHA-256 does not suffer from the avalanche effect like older hashing algorithms. It is easy to implement and integrate as it works in a single direction.

## Generate Checksum: Refactor the code to encrypt a text string and generate a checksum verification.



## Verification: Demonstrate that a hash value has been created for the unique text string (your first and last name) by executing the Java code.

