# 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

Secure Hash Algorithms or SHA have been used for many years and are constantly evolving. One of the earliest we know is SHA-1, which is a secure hash algorithm that uses a 160-bit key. SHA 256 uses a 256-bit key and is going to avoid collision better than SHA-1, that’s why SHA 256 would be the algorithm cypher of choice.

## Justification

SHA-1 is typically faster but is susceptible to collisions due to its shorter key length. SHA 256 uses a 256-bit key and has been proven to be more secure against collision and brute force attacks. A collision occurs when an algorithm produces the same hash for two distinct pieces of data. SHA 256 outputs characters composed of characters ranging from a to z and numerals from 0 to 9, this creates 36^64 different possibilities, which makes it extremely rare for a collision to occur. SHA 256 is slower than SHA 1 since it has a 256-bit key, but in the long run, the shorter run time of SHA-1 does not amount to the provided security that SHA-256 offers.

## Generate Checksum

A screen shot of a computer

Description automatically generated

## Verification

Insert a screenshot below of the web browser with your unique information.

I unfortunately could not figure out this part of the verification process. I tried looking at the resources that were presented in our modules from 1 – 5. I attempted to google how to connect to the RESTful API server to verify that the checksum produced the data and key. In the resources it states that when reading the directions for the assignment it says to make sure the code is executable first and then verify. I attempted to go through the run configurations for the IDE to see if there was a way to export the file to the server and that did not work. I believe that my @requestmapping is correct, so I am just unsure where I have gone wrong with the verification.