# 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

For this assignment regarding the algorithm cipher that avoids collisions I decided to implement the SHA-256 (Secure Hash Algorithm 256-bit algorithm. This algorithm is especially useful in avoiding collisions where different inputs may produce the same hash value and thus making plaintext/raw data insecure. It’s also one of the recommended algorithms from the Java Security Standard Algorithm Names by Oracle.

## Justification

SHA-256 is widely implemented and trusted for its security by many organizations. It is utilized in many security applications and protocols, including digital signatures, certifi9cate generations, and checksum verification. SHA-256 also features collision avoidance, which is a critical property of cryptographic hash functions. It’s widely accepted across many industries and its hash functions are heavily analyzed. Making it one of the most secure cryptographic algorithms for hash functions.

It’s important to consider cryptographic hash functions that implement collision avoidance because it is incredibly useful in situations regarding checksum verification. If two different sets of data happen to produce the checksum, this can lead false positives which can be an attack vector for threat actors who may create malicious applications that register the same checksum, thus taking advantage of security protocols or worse. When a client must download a public key, the collision-resistant hash function like SHA-256 ensure that the downloaded public key is identical to the original, making sure that the key is properly authenticated without risk of tampering by unauthorized individuals.

## Generate Checksum

You’ll submit your refactored code to your instructor. Your instructor will review it and this document.

## Verification

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

A screenshot of a computer

Description automatically generated

**Bibliography**

GeeksforGeeks. (2023a, February 24). *Cryptographic hash function in Java*. https://www.geeksforgeeks.org/cryptographic-hash-function-in-java/

GeeksforGeeks. (2022, May 31). *Implementing checksum using Java*. <https://www.geeksforgeeks.org/implementing-checksum-using-java/>

*Java Cryptography - Message Digest*. (n.d.). https://www.tutorialspoint.com/java\_cryptography/java\_cryptography\_message\_digest.htm

*MessageDigest (Java Platform SE 8 )*. (2023, October 4). <https://docs.oracle.com/javase/8/docs/api/java/security/MessageDigest.html>

Wagner, L. (2022, October 12). What is SHA-256? *What is SHA-256?* https://blog.boot.dev/cryptography/how-sha-2-works-step-by-step-sha-256/