# CS 305 Module Five Checksum Verification Assignment

**Brandon Hobbs**

**June 6, 2022**

**Instructions:** Replace the bracketed text with your answers in your own words.

## Algorithm Cipher

Recommend an appropriate encryption algorithm cipher that avoids collisions.

SHA2-256 was selected for the hashing algorithm. SHA2-256 has a probability of two hashes accidentally colliding of approximately 4.3\*10-60. MD5, SHA-0 and -1 all have collisions found.

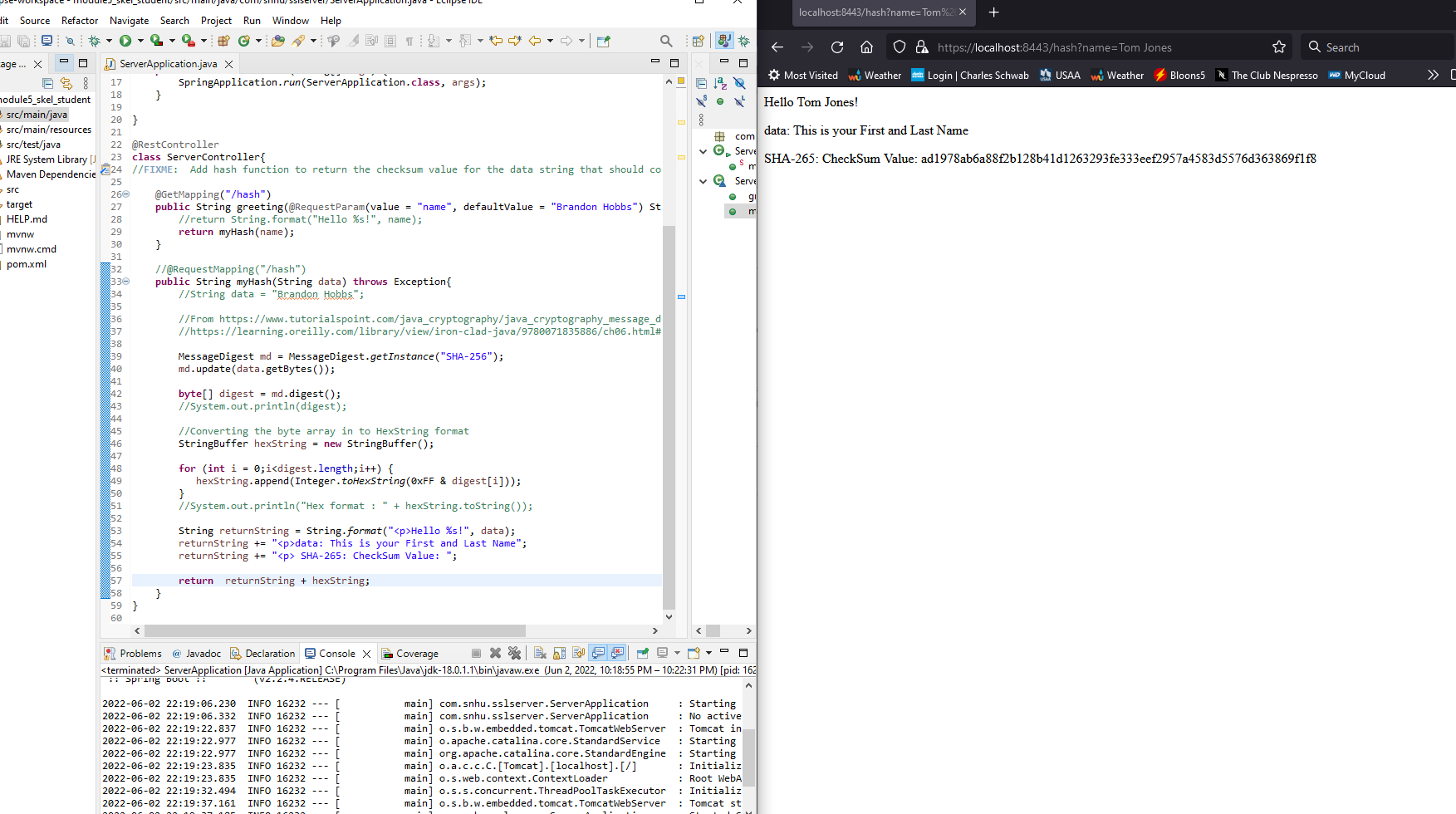
## Justification

SHA-256 is slower than, say, MD5 by about 60% but much stronger than SHA-1. For a typical, non-critical system 256 bits is probably enough. There is no need to suffer the slower performance of SHA-384 or -512.

## Generate Checksum

Refactor the code to encrypt a text string and generate a checksum verification. You will submit your refactored code for your instructor to review in addition to this document.

## Verification



*package com.snhu.sslserver;*

*import org.springframework.boot.SpringApplication;*

*import org.springframework.boot.autoconfigure.SpringBootApplication;*

*import org.springframework.web.bind.annotation.RestController;*

*import org.springframework.web.bind.annotation.RequestParam;*

*import org.springframework.web.bind.annotation.GetMapping;*

*import java.security.MessageDigest;*

*@SpringBootApplication*

*public class ServerApplication {*

*public static void main(String[] args) {*

*SpringApplication.run(ServerApplication.class, args);*

*}*

*}*

*@RestController*

*class ServerController{*

*//FIXME: Add hash function to return the checksum value for the data string that should contain your name.*

*@GetMapping("/hash")*

*public String greeting(@RequestParam(value = "name", defaultValue = "Brandon Hobbs") String name) throws Exception {*

*//return String.format("Hello %s!", name);*

*return myHash(name);*

*}*

*//@RequestMapping("/hash")*

*public String myHash(String data) throws Exception{*

*//String data = "Brandon Hobbs";*

*//From https://www.tutorialspoint.com/java\_cryptography/java\_cryptography\_message\_digest.htm*

*//https://learning.oreilly.com/library/view/iron-clad-java/9780071835886/ch06.html#ch06lev2sec7*

*MessageDigest md = MessageDigest.getInstance("SHA-256");*

*md.update(data.getBytes());*

*byte[] digest = md.digest();*

*//System.out.println(digest);*

*//Converting the byte array in to HexString format*

*StringBuffer hexString = new StringBuffer();*

*for (int i = 0;i<digest.length;i++) {*

*hexString.append(Integer.toHexString(0xFF & digest[i]));*

*}*

*//System.out.println("Hex format : " + hexString.toString());*

*String returnString = String.format("<p>Hello %s!", data);*

*returnString += "<p>data: This is your First and Last Name";*

*returnString += "<p> SHA-265: CheckSum Value: ";*

*return returnString + hexString;*

*}*

*}*