**EXPERIMENT-5**

**Student’s Name: Gurvir Kaur UID: 20BCS2038**

**Section/Group: 707\_WM\_B Subject Code: 20CSP-338**

**Subject Name: WMS Lab Date of performance:06/10/2022**

**Branch: BE CSE Semester:5th**

**Aim:** Write a program to generate message digest for the given message using the SHA/MD5 algorithm and verify the integrity of message.

**Software/Hardware Requirements:**

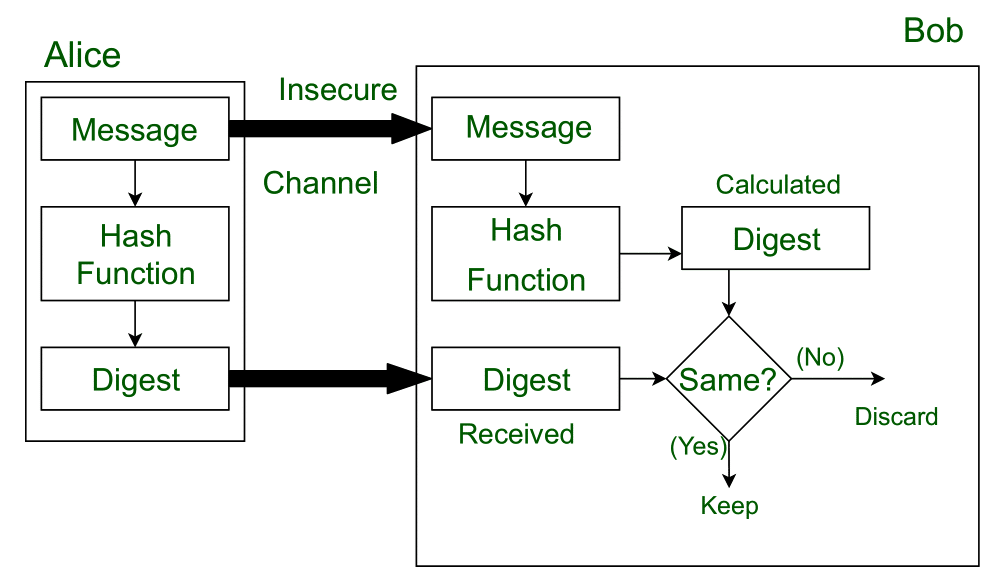
window 7 and above version

**Tools to be used:**

1. Eclipse IDE
2. JDK (Java Development kit)
3. IntelliJ IDEA

**INTRODUCTION**

**Message Digest** is used to ensure the integrity of a message transmitted over an insecure channel (where the content of the message can be changed). The message is passed through a [Cryptographic hash function](https://www.geeksforgeeks.org/passwords-and-cryptographic-hash-function/). This function creates a compressed image of the message called **Digest**.



**Steps/Method/Coding:**

To calculate cryptographic hashing value in Java, **MessageDigest** Class is used, under the package java.security.

MessageDigest Class provides following cryptographic hash function to find hash value of a text as follows:

* MD2
* MD5
* SHA-1
* SHA-224
* SHA-256
* SHA-384
* SHA-512

1.This Algorithms are initialize in static method called **getInstance()**.

2.  After selecting the algorithm it calculate the **digest** value and return the results in byte array.

3. BigInteger class is used, which converts the resultant byte array into its **sign-  magnitude representation**.

4.This representation is then converted into a hexadecimal format to get the expected MessageDigest.

**Coding (MD5 algorithm & SHA algorithm):**

package wms;

import java.math.BigInteger;

import java.security.MessageDigest;

import java.security.NoSuchAlgorithmException;

public class Main {

public static String iMD5(String inp) {

try {

MessageDigest md = MessageDigest.getInstance("MD5");

byte[] msgDgs = md.digest(inp.getBytes());

BigInteger no = new BigInteger(1, msgDgs);

StringBuilder hashTxt = new StringBuilder(no.toString(16));

while (hashTxt.length() < 32) {

hashTxt.insert(0, "0");

}

return hashTxt.toString();

} catch (NoSuchAlgorithmException e) {

throw new RuntimeException(e);

}

}

public static String iSHA(String inp) {

try {

MessageDigest md = MessageDigest.getInstance("SHA-1");

byte[] msgDgs = md.digest(inp.getBytes());

BigInteger no = new BigInteger(1, msgDgs);

StringBuilder hashTxt = new StringBuilder(no.toString(16));

while (hashTxt.length() < 32) {

hashTxt.insert(0, "0");

}

return hashTxt.toString();

} catch (NoSuchAlgorithmException e) {

throw new RuntimeException(e);

}

}

public static void main(String[] args) {

String s = "Coding Ninjas";

System.out.println("Message is: " + s);

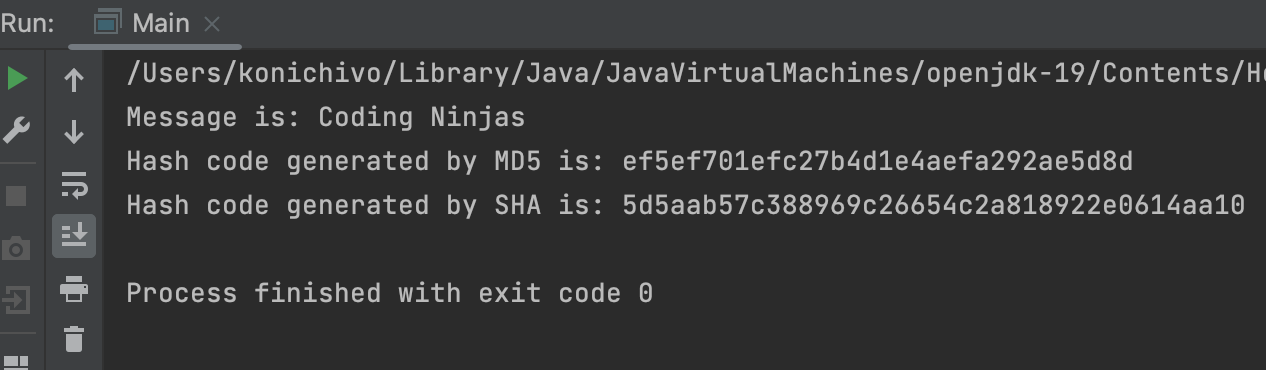
System.out.println("Hash code generated by MD5 is: " + iMD5(s));

System.out.println("Hash code generated by SHA is: " + iSHA(s));

}

}

OUTPUT:



SCREENSHOTS:

