**Worksheet-2.1**

**Student Name:-** Pushpraj Roy **UID:-** 20BCS9866

**Branch:-** BE- CSE **Section/Group:-** WM\_617 “A”

**Subject Code:-** 20CSP-338 **Semester:-** 5th

**Subject Name:-** Web and Mobile Lab

1. **Aim/Overview of the practical: -**

Perform Penetration testing on a web application to gather information about the system (Foot Printing)

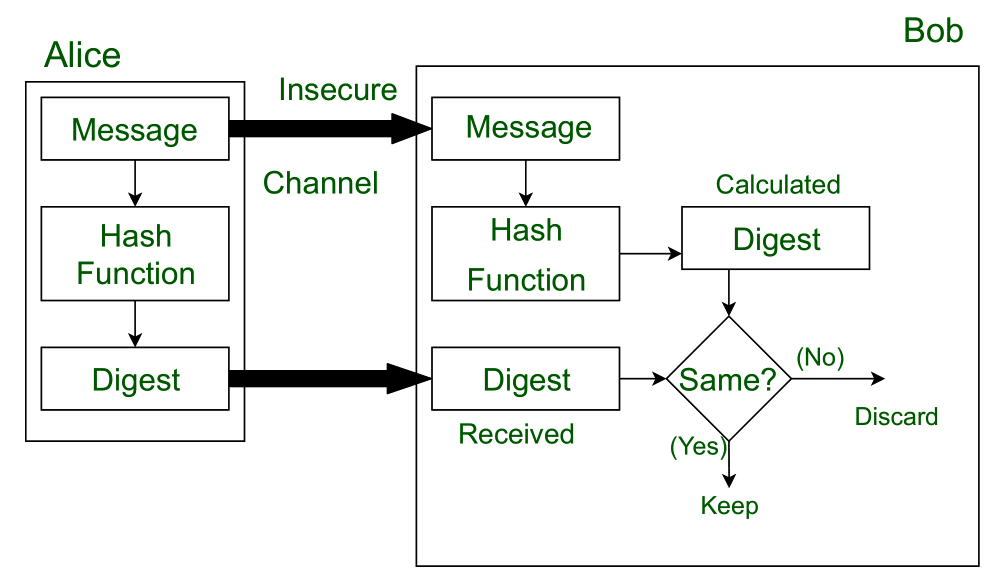
1. **Task to be done/ Which logistics used:**
2. Write the Code to perform the MD5 Encryption.
3. Write the code to perform the SHA-1 Encryption.
4. **Apparatus / Simulator Used:**

* Windows 7 & above version.
* Demotest fire site
* Google Chrome

1. **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. This function creates a compressed image of the message called Digest.

Let’s assume, Alice sent a message and digest pair to Bob. To check the integrity of the message Bob runs the cryptographic hash function on the received message and gets a new digest. Now, Bob will compare the new digest and the digest sent by Alice. If, both are same then Bob is sure that the original message is not changed.



This message and digest pair is equivalent to a physical document and fingerprint of a person on that document. Unlike the physical document and the fingerprint, the message and the digest can be sent separately.

* Most importantly, the digest should be unchanged during the transmission.
* The cryptographic hash function is a one-way function, that is, a function which is practically infeasible to invert. This cryptographic hash function takes a message of variable length as input and creates a **digest / hash / fingerprint** of fixed length, which is used to verify the integrity of the message.
* Message digest ensures the integrity of the document. To provide authenticity of the message, digest is encrypted with sender’s private key. Now this digest is called digital signature, which can be only decrypted by the receiver who has sender’s public key. Now the receiver can authenticate the sender and also verify the integrity of the sent message.

1. **Program/ Steps/ Method/ Code:**

**MD5:**

import java.math.BigInteger;

import java.security.MessageDigest;

import java.security.NoSuchAlgorithmException;

import java.util.Scanner;

public class MD5 {

public static String getMd5(String input) {

try {

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

byte[] messageDigest = md.digest(input.getBytes());

BigInteger no = new BigInteger(1, messageDigest);

String hashtext = no.toString(16);

while (hashtext.length() < 32) {

hashtext = "0" + hashtext;

}

return hashtext;

}

catch (NoSuchAlgorithmException e) {

throw new RuntimeException(e);

}

}

public static void main(String args[]) throws NoSuchAlgorithmException {

Scanner in = new Scanner(System.in);

System.out.println("Enter your String: ");

String s = in.nextLine();

System.out.println("HashCode Generated by MD5 for: ");

System.out.println(s + " is : " + getMd5(s));

in.close();

}

}

**SHA-1:**

import java.math.BigInteger;

import java.security.MessageDigest;

import java.security.NoSuchAlgorithmException;

import java.util.Scanner;

public class SHA{

public static String encryptThisString(String input) {

try {

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

byte[] messageDigest = md.digest(input.getBytes());

BigInteger no = new BigInteger(1, messageDigest);

String hashtext = no.toString(16);

while (hashtext.length() < 32) {

hashtext = "0" + hashtext;

}

return hashtext;

} catch (NoSuchAlgorithmException e) {

throw new RuntimeException(e);

}

}

public static void main(String args[]) throws NoSuchAlgorithmException {

Scanner in = new Scanner(System.in);

System.out.println("Enter your String: ");

String s = in.nextLine();

System.out.println("HashCode Generated by SHA-1 for: ");

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

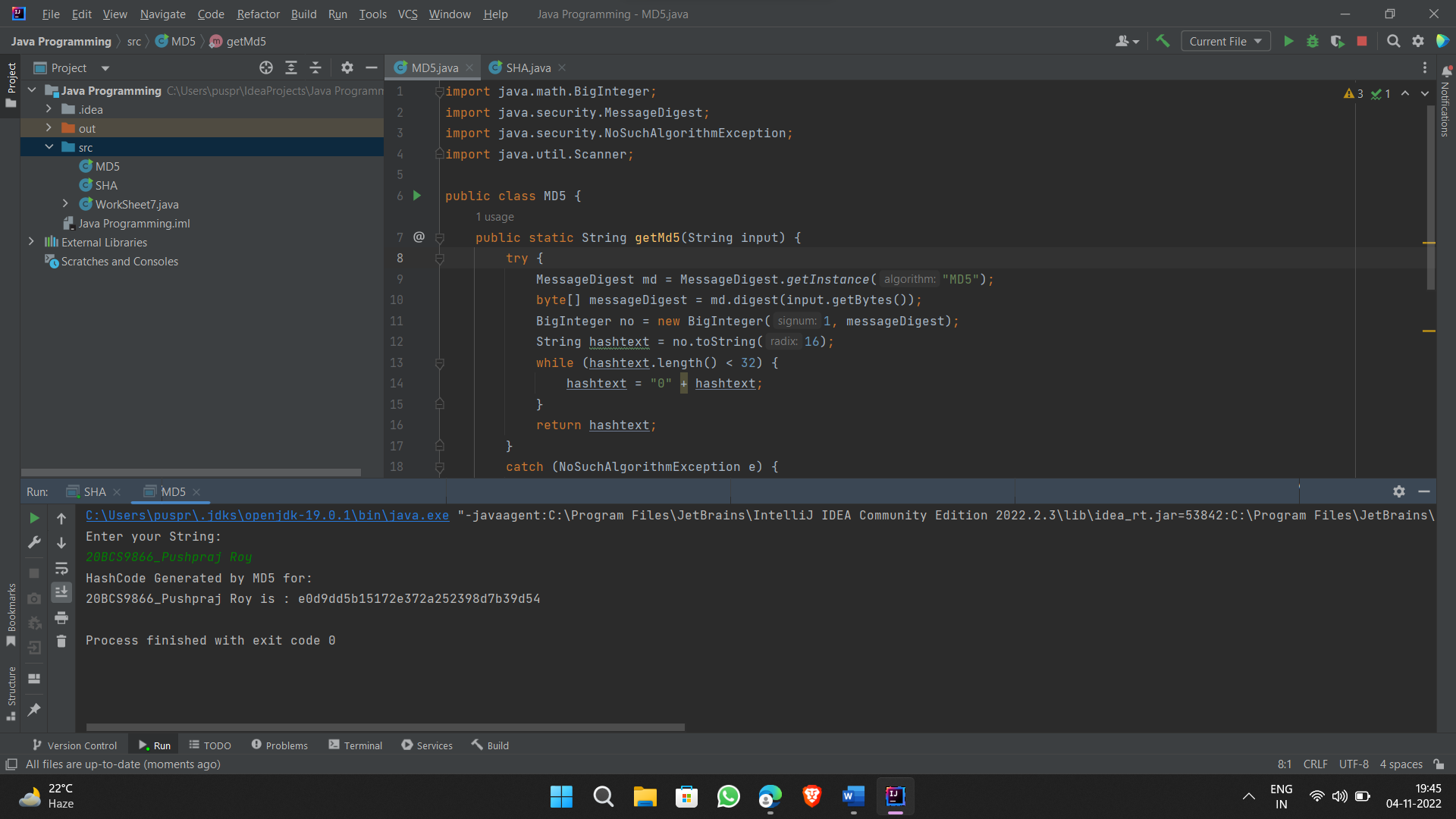
in.close();

}

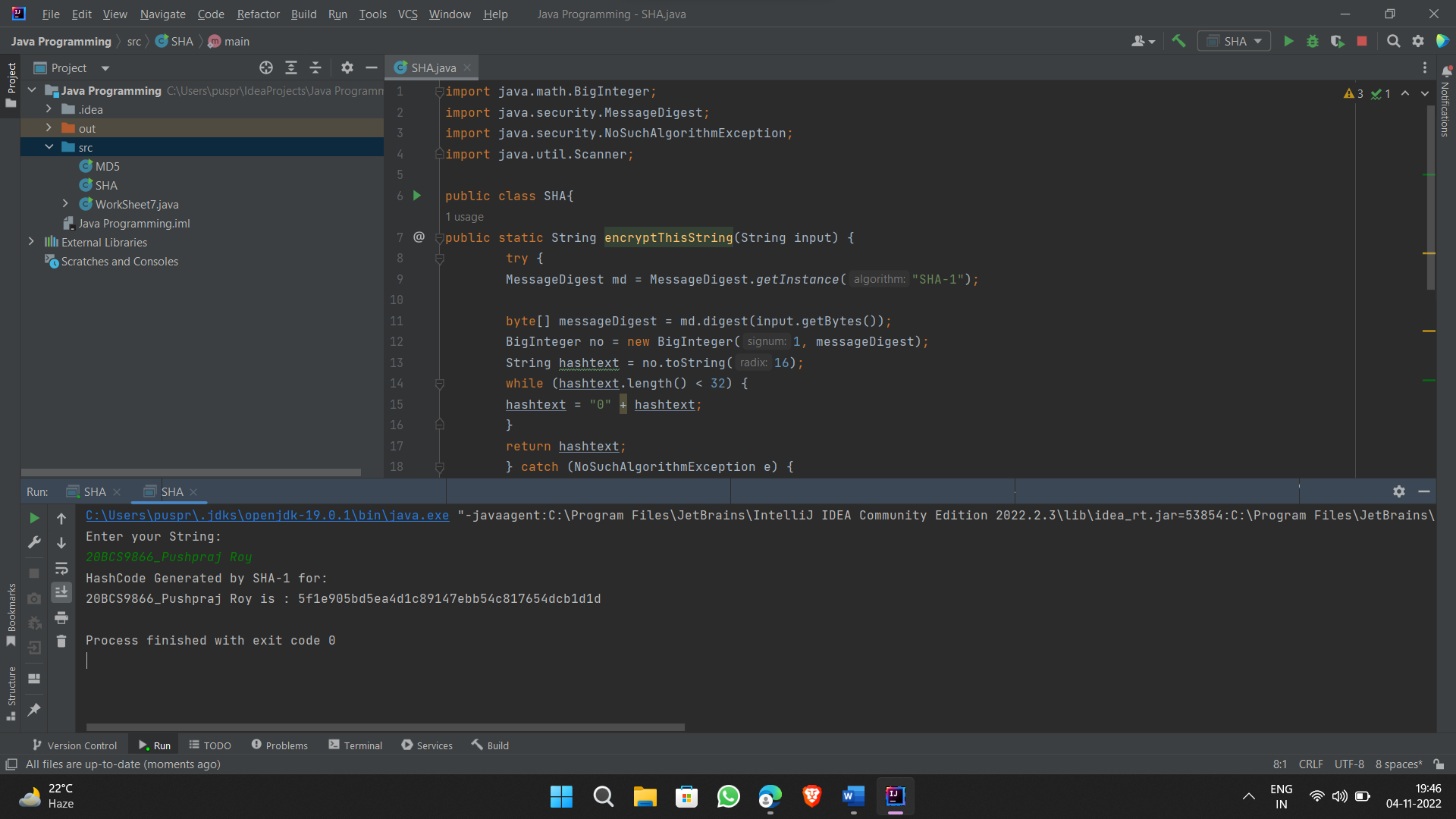
}

1. **Output:-**

MD5:

****

SHA-1

****

1. **Learning outcomes (What I have learnt):**

* Output is often known as hash values, hash codes, message digest.
* The length of output hashes is generally less than its corresponding input message length.