

**CHANDIGARH UNIVERSITY
UNIVERSITY INSTITUTE OF NGINEERING
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**



Submitted By: Vivek Kumar(21BCS8129)		Submitted To: Er. Himanshi (13362)
Subject Name	Web and Mobile Security Lab	
Subject Code	20CSP-338	
Branch	Computer Science and Engineering	
Semester	5 th	

Experiment - 5

Student Name: Vivek Kumar

UID: 21BCS8129

Branch: BE-CSE(LEET)

Section/Group: WM-20BCS-616/A

Semester: 5th

Date of Performance: 12/09/2022

Subject Name: Web and Mobile Security Lab

Subject Code: 20CSP-338

1. Aim/Overview of the practical:

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

2. Task to be done/ Which logistics used:

Write the Code to perform the MD5 Encryption.

Write the code to perform the SHA-1 Encryption.

3. Apparatus / Simulator Used:

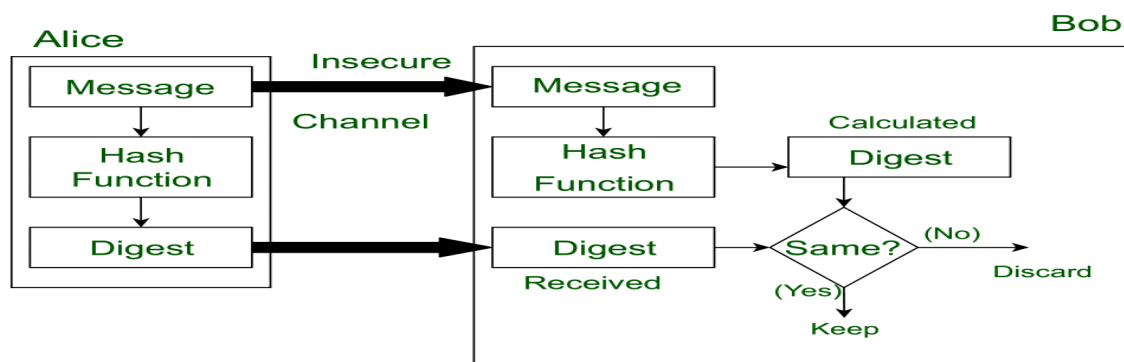
1. Windows 7 & above version.
2. demotest fire site
3. Google Chrome

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**.

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.

4. Program/ Steps/ Method/ Code:

MD5:

```
import java.math.BigInteger;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.Scanner;

public class WMS_EXP5_A {
    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;

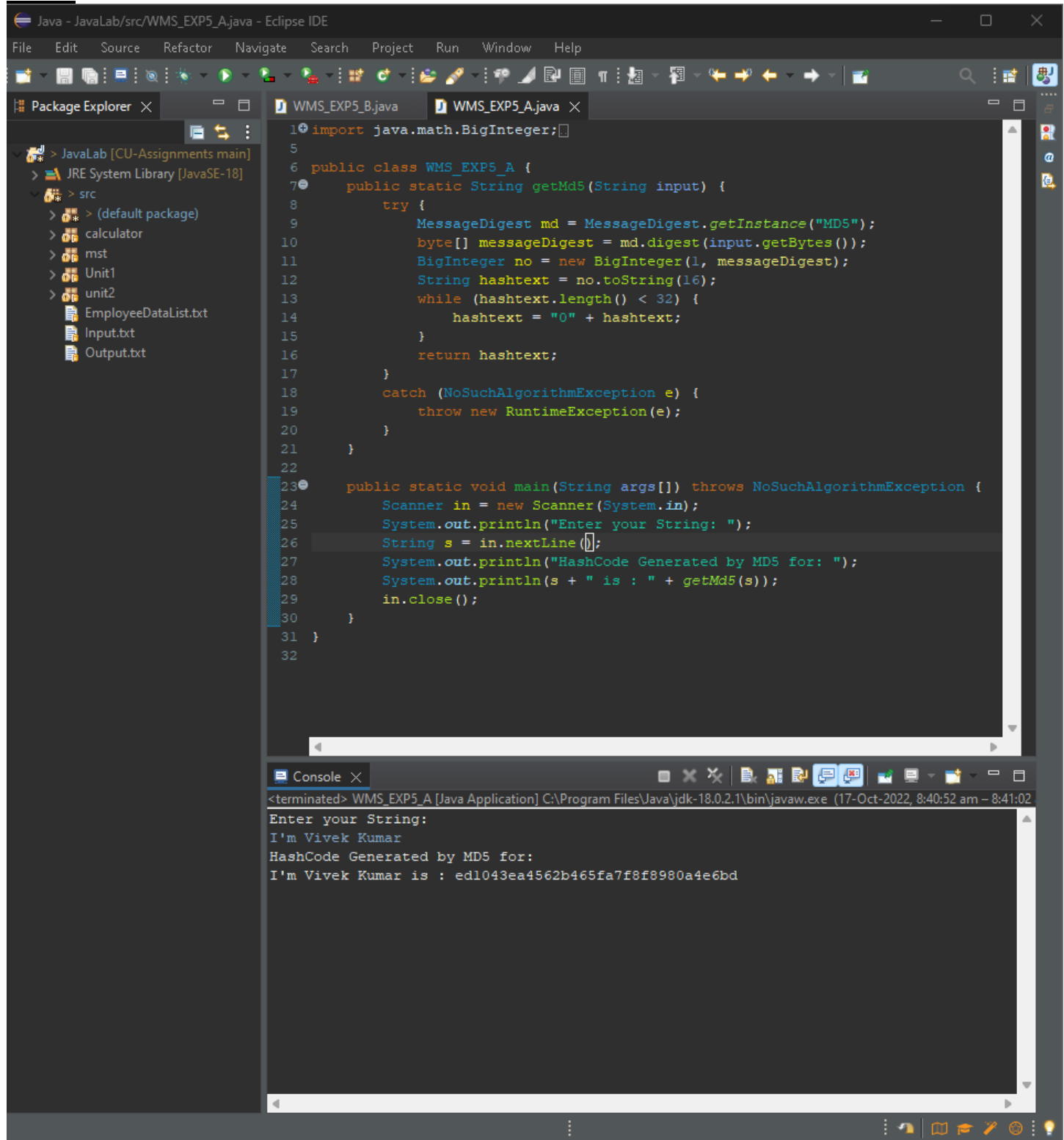
public class WMS_EXP5_B {
    public static String encryptThisString(String input) {
        try {
            MessageDigest md = MessageDigest.getInstance("SHA-1");

            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();  
}  
}
```

5. DBMS Script/Result/Output/Writing Summary:

MD5:



The screenshot displays the Eclipse IDE with a Java project named 'JavaLab'. The Package Explorer on the left shows the project structure, including 'src' and various files like 'EmployeeDataList.txt', 'Input.txt', and 'Output.txt'. The main editor shows the source code for 'WMS_EXP5_A.java'. The code imports 'java.math.BigInteger' and defines a 'getMd5' method that uses 'MessageDigest' to generate an MD5 hash. The 'main' method prompts the user for a string, reads it from 'System.in', and prints the generated MD5 hash.

```

1 import java.math.BigInteger;
2
3
4
5
6 public class WMS_EXP5_A {
7     public static String getMd5(String input) {
8         try {
9             MessageDigest md = MessageDigest.getInstance("MD5");
10            byte[] messageDigest = md.digest(input.getBytes());
11            BigInteger no = new BigInteger(1, messageDigest);
12            String hashtext = no.toString(16);
13            while (hashtext.length() < 32) {
14                hashtext = "0" + hashtext;
15            }
16            return hashtext;
17        }
18        catch (NoSuchAlgorithmException e) {
19            throw new RuntimeException(e);
20        }
21    }
22
23    public static void main(String args[]) throws NoSuchAlgorithmException {
24        Scanner in = new Scanner(System.in);
25        System.out.println("Enter your String: ");
26        String s = in.nextLine();
27        System.out.println("HashCode Generated by MD5 for: ");
28        System.out.println(s + " is : " + getMd5(s));
29        in.close();
30    }
31 }
32

```

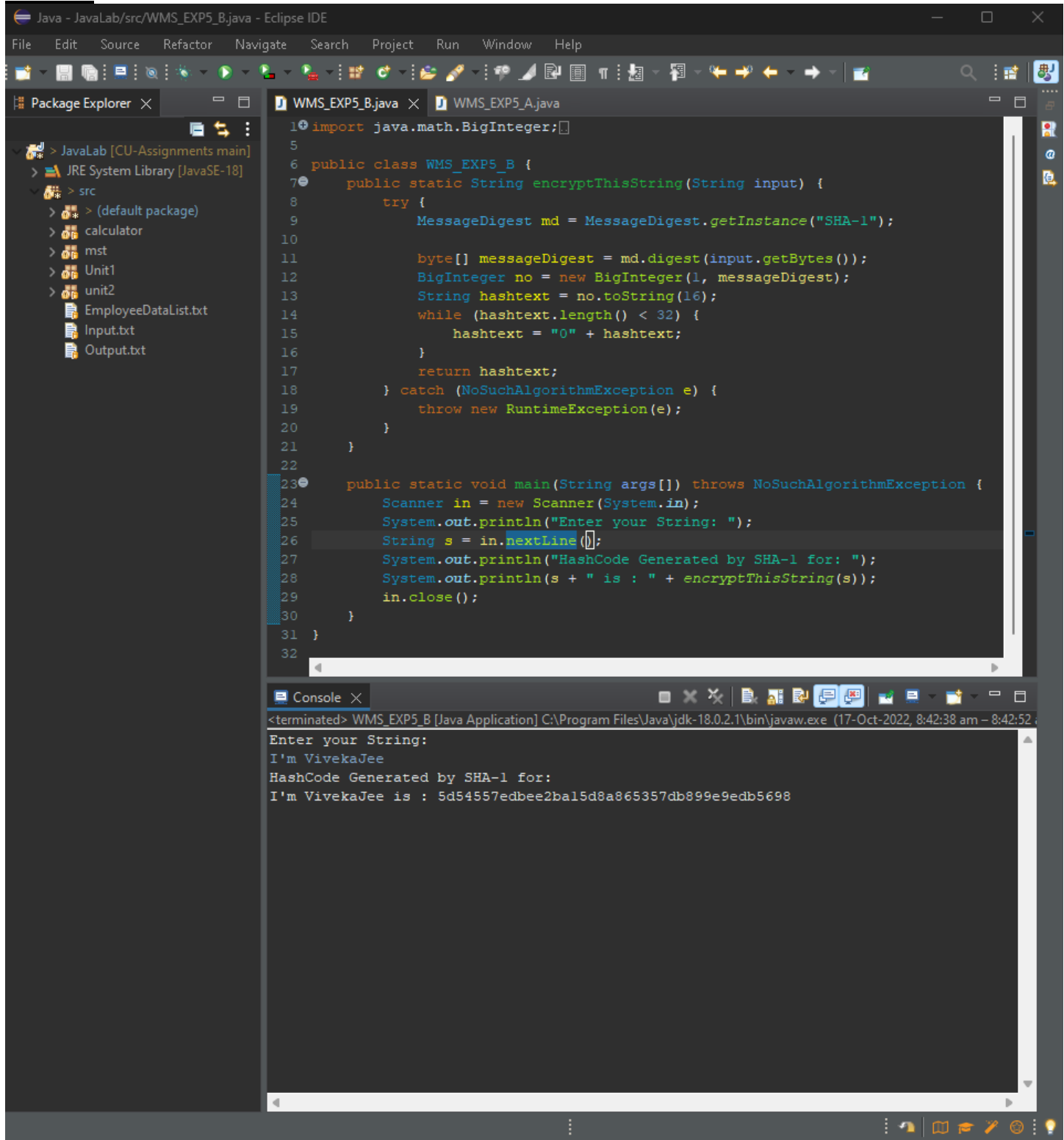
The Console window at the bottom shows the execution output:

```

<terminated> WMS_EXP5_A [Java Application] C:\Program Files\Java\jdk-18.0.2.1\bin\javaw.exe (17-Oct-2022, 8:40:52 am - 8:41:02)
Enter your String:
I'm Vivek Kumar
HashCode Generated by MD5 for:
I'm Vivek Kumar is : ed1043ea4562b465fa7f8f8980a4e6bd

```

SHA-1:



The screenshot displays the Eclipse IDE environment. The Package Explorer on the left shows the project structure for 'JavaLab [CU-Assignments main]', including the 'src' package with files like 'calculator', 'mst', 'Unit1', 'unit2', 'EmployeeDataList.txt', 'Input.txt', and 'Output.txt'. The main editor window shows the code for 'WMS_EXP5_B.java'. The code imports 'java.math.BigInteger' and defines a class 'WMS_EXP5_B' with a static method 'encryptThisString' and a 'main' method. The 'encryptThisString' method uses 'MessageDigest' to generate a SHA-1 hash. The 'main' method uses a 'Scanner' to take user input and prints the generated hash. The Console window at the bottom shows the execution output, where the user input 'I'm VivekaJee' is hashed to '5d54557edbee2ba15d8a865357db899e9edb5698'.

```
1 import java.math.BigInteger;
2
3
4
5
6 public class WMS_EXP5_B {
7     public static String encryptThisString(String input) {
8         try {
9             MessageDigest md = MessageDigest.getInstance("SHA-1");
10
11             byte[] messageDigest = md.digest(input.getBytes());
12             BigInteger no = new BigInteger(1, messageDigest);
13             String hashtext = no.toString(16);
14             while (hashtext.length() < 32) {
15                 hashtext = "0" + hashtext;
16             }
17             return hashtext;
18         } catch (NoSuchAlgorithmException e) {
19             throw new RuntimeException(e);
20         }
21     }
22
23     public static void main(String args[]) throws NoSuchAlgorithmException {
24         Scanner in = new Scanner(System.in);
25         System.out.println("Enter your String: ");
26         String s = in.nextLine();
27         System.out.println("HashCode Generated by SHA-1 for: ");
28         System.out.println(s + " is : " + encryptThisString(s));
29         in.close();
30     }
31 }
32
```

<terminated> WMS_EXP5_B [Java Application] C:\Program Files\Java\jdk-18.0.2.1\bin\javaw.exe (17-Oct-2022, 8:42:38 am - 8:42:52)

Enter your String:
I'm VivekaJee
HashCode Generated by SHA-1 for:
I'm VivekaJee is : 5d54557edbee2ba15d8a865357db899e9edb5698

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.

Evaluation Grid (To be created per the faculty's SOP and Assessment guidelines):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.	Worksheet completion including writing learning objectives/Outcomes. (To be submitted at the end of the day).		
2.	Post-Lab Quiz Result.		
3.	Student Engagement in Simulation/Demonstration/Performance and Controls/Pre-Lab Questions.		
	Signature of Faculty (with Date):	Total Marks Obtained:	