Experiment - 5

Name: Aniket Kumar **UID: 20BCS5306**

Semester: 5 **Section: 703/B**

Subject: Web and Mobile Security LAB **Subject Code: 20CSP-338**

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:

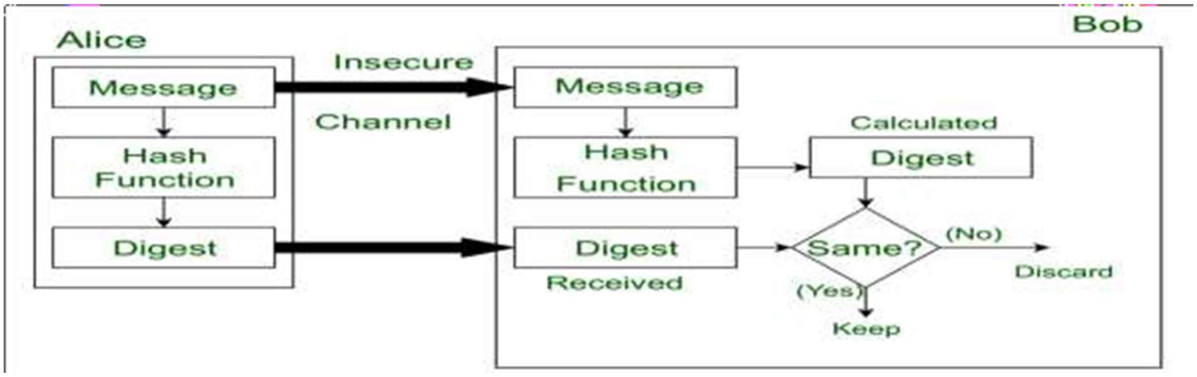
Window 7 and above version **Tools**

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. 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 **get Instance()**.

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

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

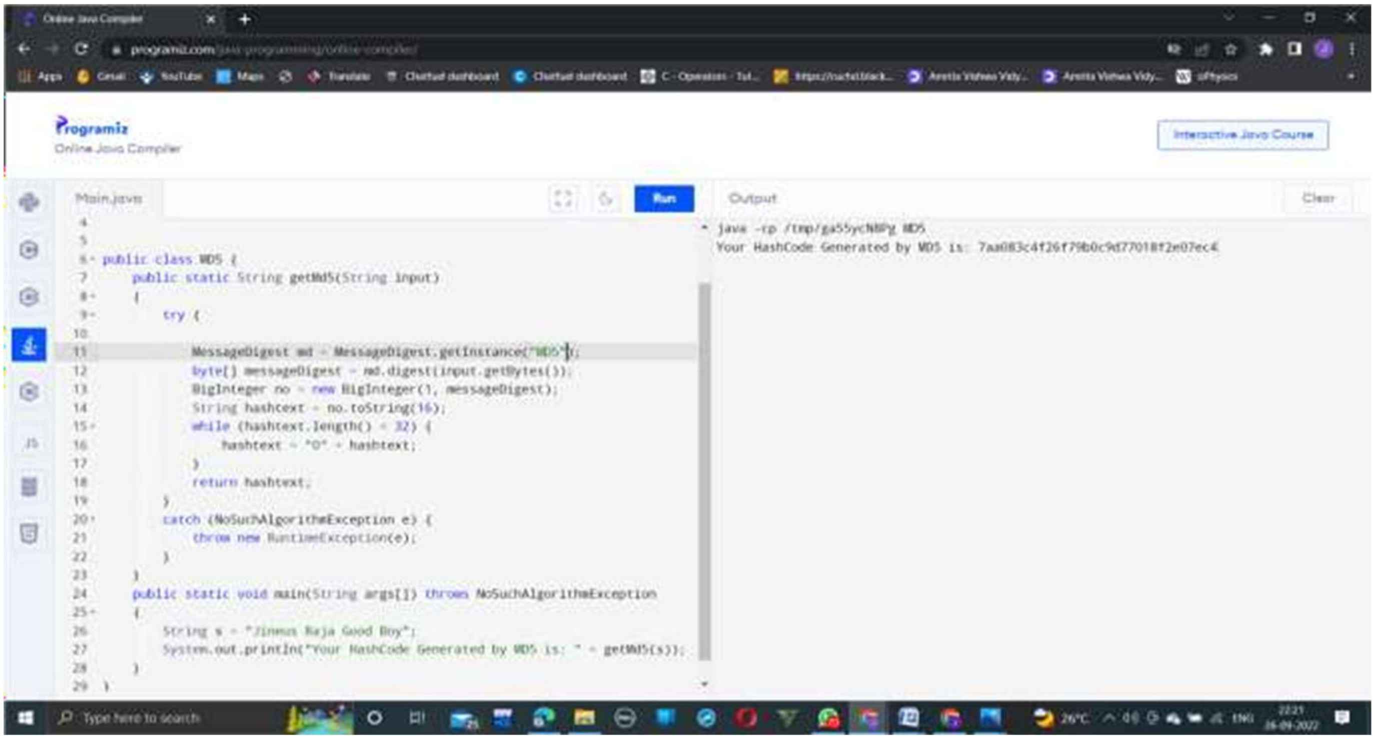
4. This representation is then converted into a hexadecimal format to get the expected Message Digest.

Coding (MD5 algorithm):………………………………………………………….

```
import java.math.BigInteger; import

java.security.MessageDigest; import

java.security.NoSuchAlgorithmException;



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

    {

        String s = "Jineus Raja Good Boy";

 System.out.println("Your HashCode Generated by MD5 is: " + getMd5(s));

    }

}
```

OUTPUT:



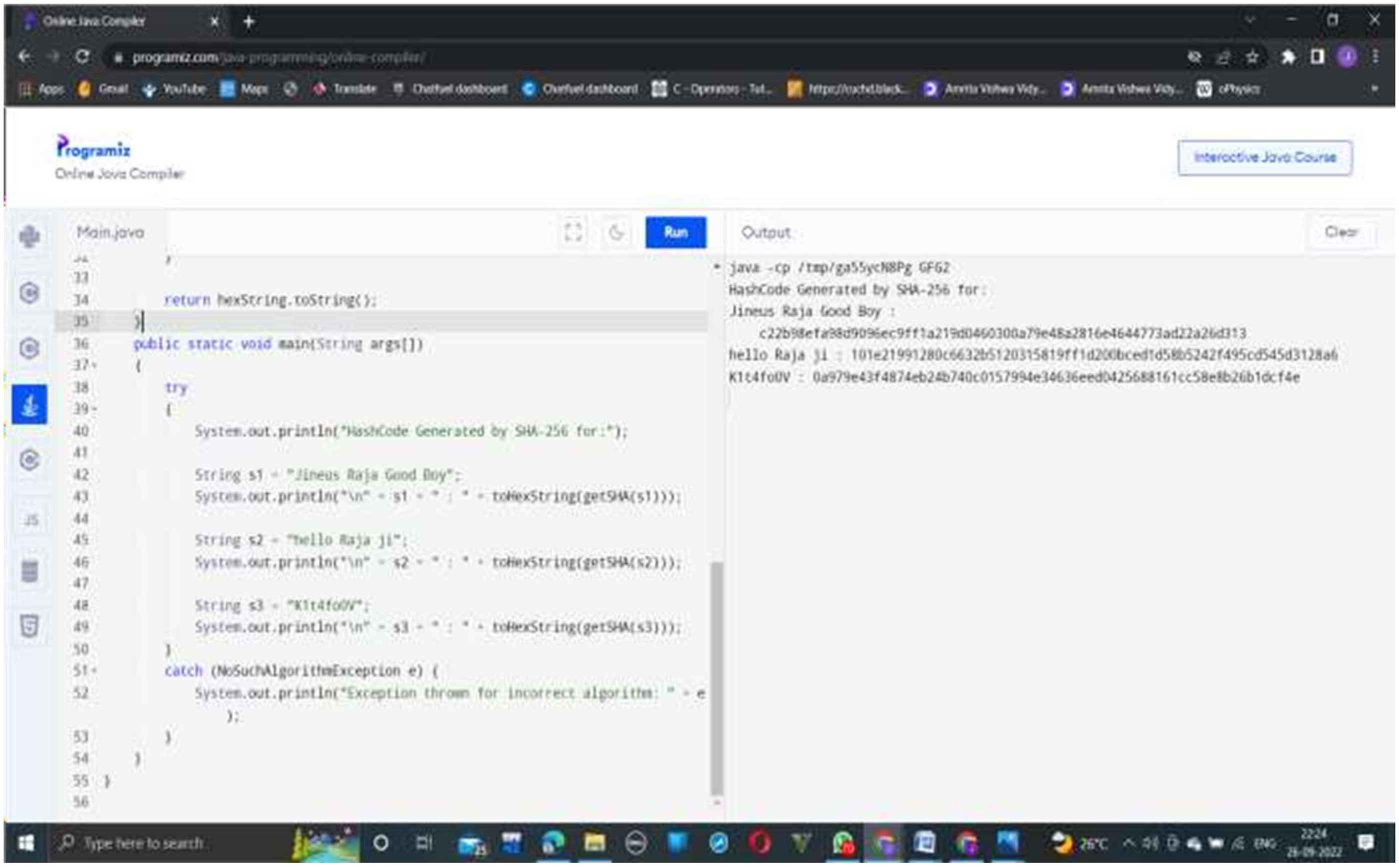
```
public class MD5 {
    public static String getMd5(String input)
    {
        try {

            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
    {
        String s = "Jinnus Raja Good Boy";
        System.out.println("Your HashCode Generated by MD5 is: " + getMd5(s));
    }
}
```

```
java -cp /tmp/ga55ycNBPg MD5
Your HashCode Generated by MD5 is: 7aa083c4f26f79b0c9d77018f2e07ec4
```

Coding (SHA-256 algorithm):

```
import java.math.BigInteger; import java.nio.charset.StandardCharsets; import
java.security.MessageDigest; import java.security.NoSuchAlgorithmException; class
GFG2 {        public static byte[] getSHA(String input) throws
NoSuchAlgorithmException

    {

            MessageDigest md = MessageDigest.getInstance("SHA-256");
    return md.digest(input.getBytes(StandardCharsets.UTF_8));

    }

    public static String toHexString(byte[] hash)
    {

        BigInteger number = new BigInteger(1, hash);

        StringBuilder hexString = new StringBuilder(number.toString(16));

    while (hexString.length() < 64)

        {

            hexString.insert(0, '0');

        }


        return hexString.toString();

    }

    public static void main(String args[])

    {

    try

        {

            System.out.println("HashCode Generated by SHA-256 for:");


            String s1 = "Jineus Raja Good Boy";

  System.out.println("\n" + s1 + " : " + toHexString(getSHA(s1)));


            String s2 = "hello Raja ji";

  System.out.println("\n" + s2 + " : " + toHexString(getSHA(s2)));                    String s3 = 
"K1t4fo0V";

            System.out.println("\n" + s3 + " : " + toHexString(getSHA(s3)));
            }
            catch (NoSuchAlgorithmException e) {
                System.out.println("Exception thrown for incorrect algorithm: " + e);
            }
        }
}
```

OUTPUT:-



```
        }

        return hexString.toString();
    }
    public static void main(String args[])
    {
        try
        {
            System.out.println("HashCode Generated by SHA-256 for:");

            String s1 = "Jineus Raja Good Boy";
            System.out.println("\n" + s1 + " : " + toHexString(getSHA(s1)));

            String s2 = "hello Raja ji";
            System.out.println("\n" + s2 + " : " + toHexString(getSHA(s2)));

            String s3 = "K1t4fo0V";
            System.out.println("\n" + s3 + " : " + toHexString(getSHA(s3)));
        }
        catch (NoSuchAlgorithmException e) {
            System.out.println("Exception thrown for incorrect algorithm: " + e
                );
        }
    }
}
```

```
java -cp /tmp/ga55ycN8Pg GFG2
HashCode Generated by SHA-256 for:
Jineus Raja Good Boy :
    c22b98efa98d9096ec9ff1a219d0460300a79e48a2816e4644773ad22a26d313
hello Raja ji : 101e21991280c6632b5120315819ff1d200bced1d58b5242f495cd545d3128a6
K1t4fo0V : 0a979e43f4874eb24b740c0157994e34636eed0425688161cc58e8b26b1dcf4e
```