# Experiment: 2.1

**Student Name**: Yash Kumar **UID**: 20BCS9256

**Branch**: BE-CSE **Section/Group:** 616 ‘B’

**Semester:** 5th **Subject Name**: WMS Lab

**Subject Code:** 21 CSP-338

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

## 2. Steps:

To calculate cryptographic hashing value in Java, **MessageDigest** Class is used, under thepackage java.security.
MessageDigest Class provides following cryptographic hash function to find hash value of atext 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 bytearray.

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

**4.** This representation is then converted into a hexadecimal format to get the expectedMessageDigest.

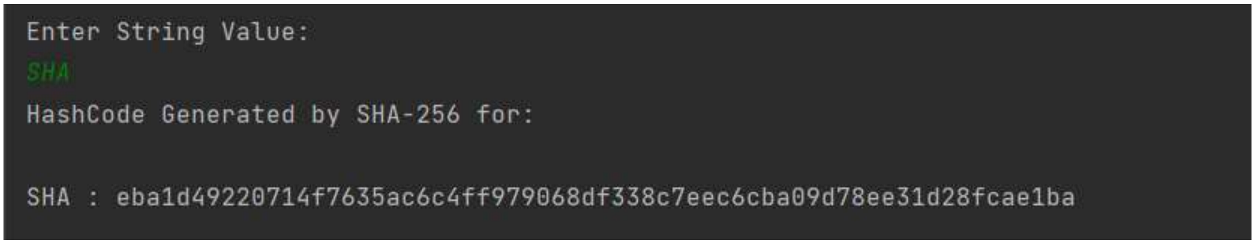
## 3. Code and output:

## <u>Coding (SHA-256 algorithm):</u>

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

class WMS4 {
  public static byte[] getSHA(String input) throws
NoSuchAlgorithmException
  {
    // Static getInstance method is called with hashing SHA
    MessageDigest md = MessageDigest.getInstance("SHA-256");

    // digest() method called
    // to calculate message digest of an input
    // and return array of byte
    return md.digest(input.getBytes(StandardCharsets.UTF_8));
  }

  public static String toHexString(byte[] hash)
  {
    // Convert byte array into signum representation
    BigInteger number = new BigInteger(1, hash);

    // Convert message digest into hex value
    StringBuilder hexString = new StringBuilder(number.toString(16));
```

```
        // Pad with leading zeros
        while (hexString.length() < 64)
        {
            hexString.insert(0, '0');
        }
        return hexString.toString();
    }
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);

        try
        {
            System.out.println("Enter String Value:");
            String s1 = sc.nextLine();
            System.out.println("HashCode Generated by SHA-256 for:");
            System.out.println("\n" + s1 + " : " + toHexString(getSHA(s1)));
        }
        // For specifying wrong message digest algorithms
        catch (NoSuchAlgorithmException e) {
            System.out.println("Exception thrown for incorrect algorithm: " +
e);
        }

    }
}
```

**OUTPUT:**



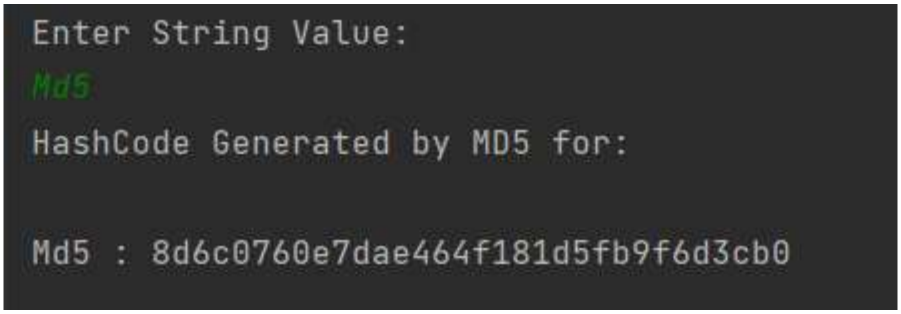
```
Enter String Value:
SHA
HashCode Generated by SHA-256 for:

SHA : eba1d49220714f7635ac6c4ff979068df338c7eec6cba09d78ee31d28fcae1ba
```

## Coding(Md5 Algorithm):

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

class WMS4 {
    public static String getMd5(String input)
    {
        try {
            // Static getInstance method is called with hashing MD5
            MessageDigest md = MessageDigest.getInstance("MD5");
            // digest() method is called to calculate message digest
            // of an input digest() return array of byte
            byte[] messageDigest = md.digest(input.getBytes());
            // Convert byte array into signum representation
            BigInteger no = new BigInteger(1, messageDigest);
            // Convert message digest into hex value
            String hashtext = no.toString(16);
            while (hashtext.length() < 32) {
                hashtext = "0" + hashtext;
            }
            return hashtext;
        }
        // For specifying wrong message digest algorithms
        catch (NoSuchAlgorithmException e) {
            throw new RuntimeException(e);
        }
    }
```

```
        public static void main(String args[])
        {
            Scanner sc=new Scanner(System.in);

            System.out.println("Enter String Value:");
            String s1 = sc.nextLine();
            System.out.println("HashCode Generated by MD5 for:");
            System.out.println("\n" + s1 + " : " + getMd5(s1));

        }
    }
```

## OUTPUT:



```
Enter String Value:
Md5
HashCode Generated by MD5 for:

Md5 : 8d6c0760e7dae464f181d5fb9f6d3cb0
```

## Learning Outcomes:

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.