**// SHA-1 hash value**

import java.math.BigInteger; **// mathematical operation which involves very big integer calculations that are outside the limit of all available primitive data types**

import java.security.MessageDigest; **//cryptographic hash function. input to the hash function is of arbitrary length but output is always of fixed length.**

import java.security.NoSuchAlgorithmException**; //This exception is thrown when a particular cryptographic algorithm is requested but is not available in the environment**

import java.util.Scanner**; //obtaining the input of the primitive types like int, double, etc. and strings**

public class Main

{

public static String encryptThisString(String input)

{

try

{

MessageDigest md = MessageDigest.getInstance("SHA-1"); **// getInstance() method is called with algorithm SHA-1**

byte[] messageDigest = md.digest(input.getBytes()); **//convert a string into sequence of bytes and returns an array of bytes digest() method is called to calculate message digest of the input stringreturned as array of byte**

BigInteger no = new BigInteger(1, messageDigest); **// Convert byte array into signum representation**

String hashtext = no.toString(16); **// Convert message digest into hex value**

while (hashtext.length() < 32) **// Add preceding 0s to make it 32 bit**

{

hashtext = "0" + hashtext;

}

return hashtext;

}

**// For specifying wrong message digest algorithms**

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 String: ");

String s1 = in.nextLine();

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

}

}