package java\_cryptography;

// Imports

import java.security.KeyPair;

import java.security.KeyPairGenerator;

import java.security.PrivateKey;

import java.security.PublicKey;

import java.security.SecureRandom;

import java.security.Signature;

import java.util.Scanner;

import javax.xml.bind.DatatypeConverter;

public class Digital\_Signature\_GeeksforGeeks

{

// Signing Algorithm private static final String

SIGNING\_ALGORITHM = "SHA256withRSA";

private static final String RSA = "RSA";

private static Scanner sc;

// Function to implement Digital signature // using SHA256 and RSA algorithm // by passing private

key. public static byte[] Create\_Digital\_Signature

(

byte[] input, PrivateKey Key) throws Exception

{

Signature signature = Signature.getInstance( SIGNING\_ALGORITHM);

signature.initSign(Key);

signature.update(input);

return signature.sign();

}

// Generating the asymmetric key pair // using SecureRandom class // functions and RSA algorithm.

public static KeyPair Generate\_RSA\_KeyPair() throws Exception {

SecureRandom secureRandom = new SecureRandom();

KeyPairGenerator keyPairGenerator = KeyPairGenerator .getInstance(RSA);

keyPairGenerator.initialize( 2048, secureRandom);

return keyPairGenerator .generateKeyPair();

} // Function for Verification of the // digital signature by using the public key

public static boolean Verify\_Digital\_Signature( byte[] input, byte[] signatureToVerify, PublicKey key) throws Exception {

Signature signature = Signature.getInstance( SIGNING\_ALGORITHM);

signature.initVerify(key);

signature.update(input);

return signature .verify(signatureToVerify); }

// Driver Code

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

String input = "GEEKSFORGEEKS IS A" + " COMPUTER SCIENCE PORTAL";

KeyPair keyPair = Generate\_RSA\_KeyPair();

// Function

Call byte[] signature = Create\_Digital\_Signature( input.getBytes(), keyPair.getPrivate());

System.out.println( "Signature Value:\n " + DatatypeConverter .printHexBinary(signature));

System.out.println( "Verification: " + Verify\_Digital\_Signature( input.getBytes(),

signature, keyPair.getPublic()));

}

}