Practical 4

Name:Satishkumar Rajan Nadar

DES Algo:

import java.security.InvalidKeyException;

import java.security.NoSuchAlgorithmException;

import javax.crypto.Cipher;

import javax.crypto.KeyGenerator;

import javax.crypto.NoSuchPaddingException;

import javax.crypto.SecretKey;

import com.sun.mail.util.BASE64DecoderStream;

import com.sun.mail.util.BASE64EncoderStream;

public class DES {

private static Cipher ecipher;

private static Cipher dcipher;

private static SecretKey key;

public static void main(String[] args) {

try {

// generate secret key using DES algorithm

key = KeyGenerator.getInstance("DES").generateKey();

ecipher = Cipher.getInstance("DES");

dcipher = Cipher.getInstance("DES");

// initialize the ciphers with the given key

ecipher.init(Cipher.ENCRYPT\_MODE, key);

dcipher.init(Cipher.DECRYPT\_MODE, key);

String encrypted = encrypt("This is a classified message!");

String decrypted = decrypt(encrypted);

System.out.println("Encrypt: " + encrypted);

System.out.println("Decrypted: " + decrypted);

}

catch (NoSuchAlgorithmException e) {

System.out.println("No Such Algorithm:" + e.getMessage());

return;

}

catch (NoSuchPaddingException e) {

System.out.println("No Such Padding:" + e.getMessage());

return;

}

catch (InvalidKeyException e) {

System.out.println("Invalid Key:" + e.getMessage());

return;

}

}

public static String encrypt(String str) {

try {

// encode the string into a sequence of bytes using the named charset

// storing the result into a new byte array.

byte[] utf8 = str.getBytes("UTF8");

byte[] enc = ecipher.doFinal(utf8);

// encode to base64

enc = BASE64EncoderStream.encode(enc);

return new String(enc);

}

catch (Exception e) {

e.printStackTrace();

}

return null;

}

public static String decrypt(String str) {

try {

// decode with base64 to get bytes

byte[] dec = BASE64DecoderStream.decode(str.getBytes());

byte[] utf8 = dcipher.doFinal(dec);

// create new string based on the specified charset

return new String(utf8, "UTF8");

}

catch (Exception e) {

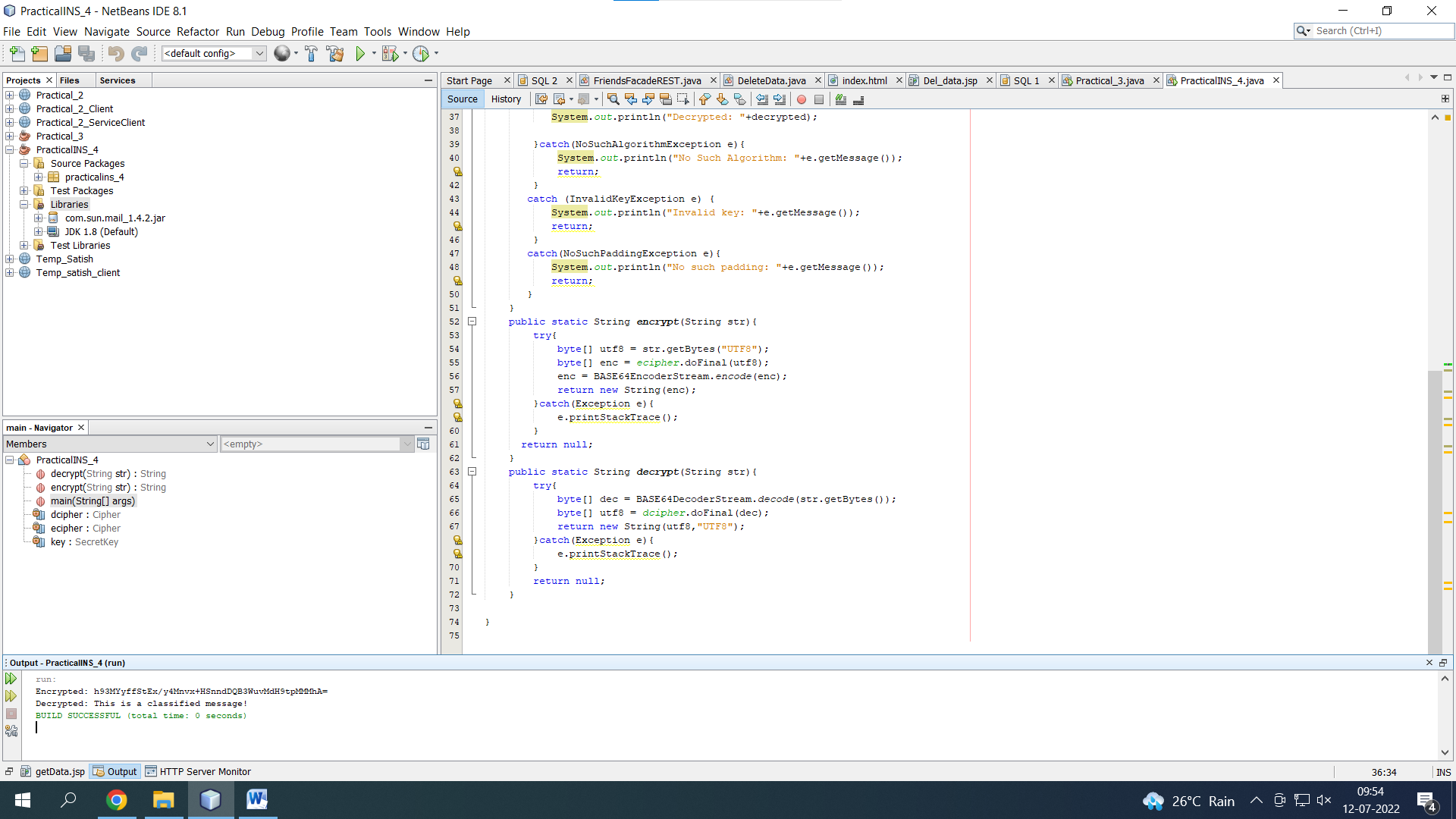
e.printStackTrace();

}

return null;

}

}



AES:-

package aes;

import javax.crypto.Cipher;

import javax.crypto.SecretKey;

import javax.crypto.SecretKeyFactory;

import javax.crypto.spec.IvParameterSpec;

import javax.crypto.spec.PBEKeySpec;

import javax.crypto.spec.SecretKeySpec;

import java.nio.charset.StandardCharsets;

import java.security.InvalidAlgorithmParameterException;

import java.security.InvalidKeyException;

import java.security.NoSuchAlgorithmException;

import java.security.spec.InvalidKeySpecException;

import java.security.spec.KeySpec;

import java.util.Base64;

import javax.crypto.BadPaddingException;

import javax.crypto.IllegalBlockSizeException;

import javax.crypto.NoSuchPaddingException;

public class AES

{

private static final String SECRET\_KEY = "123456789";

private static final String SALTVALUE = "abcdefg";

public static String encrypt(String strToEncrypt)

{

try

{

byte[] iv = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};

IvParameterSpec ivspec = new IvParameterSpec(iv);

SecretKeyFactory factory = SecretKeyFactory.getInstance("PBKDF2WithHmacSHA256");

KeySpec spec = new PBEKeySpec(SECRET\_KEY.toCharArray(), SALTVALUE.getBytes(), 65536, 256);

SecretKey tmp = factory.generateSecret(spec);

SecretKeySpec secretKey = new SecretKeySpec(tmp.getEncoded(), "AES");

Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");

cipher.init(Cipher.ENCRYPT\_MODE, secretKey, ivspec);

return Base64.getEncoder()

.encodeToString(cipher.doFinal(strToEncrypt.getBytes(StandardCharsets.UTF\_8)));

}

catch (InvalidAlgorithmParameterException | InvalidKeyException | NoSuchAlgorithmException | InvalidKeySpecException | BadPaddingException | IllegalBlockSizeException | NoSuchPaddingException e)

{

System.out.println("Error occured during encryption: " + e.toString());

}

return null;

}

public static String decrypt(String strToDecrypt)

{

try

{

byte[] iv = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};

IvParameterSpec ivspec = new IvParameterSpec(iv);

SecretKeyFactory factory = SecretKeyFactory.getInstance("PBKDF2WithHmacSHA256");

KeySpec spec = new PBEKeySpec(SECRET\_KEY.toCharArray(), SALTVALUE.getBytes(), 65536, 256);

SecretKey tmp = factory.generateSecret(spec);

SecretKeySpec secretKey = new SecretKeySpec(tmp.getEncoded(), "AES");

Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5PADDING");

cipher.init(Cipher.DECRYPT\_MODE, secretKey, ivspec);

return new String(cipher.doFinal(Base64.getDecoder().decode(strToDecrypt)));

}

catch (InvalidAlgorithmParameterException | InvalidKeyException | NoSuchAlgorithmException | InvalidKeySpecException | BadPaddingException | IllegalBlockSizeException | NoSuchPaddingException e)

{

System.out.println("Error occured during decryption: " + e.toString());

}

return null;

}

public static void main(String[] args)

{

String originalval = "AES Encryption";

String encryptedval = encrypt(originalval);

String decryptedval = decrypt(encryptedval);

System.out.println("Original value: " + originalval);

System.out.println("Encrypted value: " + encryptedval);

System.out.println("Decrypted value: " + decryptedval);

}

}

