/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package aescrypto;

import java.security.SecureRandom;

import java.util.Scanner;

import javax.crypto.Cipher;

import javax.crypto.KeyGenerator;

import javax.crypto.SecretKey;

import javax.crypto.spec.IvParameterSpec;

import javax.crypto.spec.SecretKeySpec;

/\*\*

\*

\* @author crawford, lamont

\*/

public class AesCrypto {

/\*\*

\* @param args the command line arguments

\* @throws java.lang.Exception

\*/

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

// Input

Scanner sc = new Scanner(System.in);

System.out.println("Enter Message:");

String message = sc.nextLine();

// Create and Initialize Keygenerator Instance

KeyGenerator keyGenerator = KeyGenerator.getInstance ("AES");

keyGenerator.init(256);

//Generating Symmetric Secret Key

SecretKey key = keyGenerator.generateKey();

// Initialization Vector

byte[] IV = new byte[16];

SecureRandom random = new SecureRandom();

random.nextBytes(IV);

// Output

System.out.println("message:" + message);

byte[] cipherText = encrypt(message.getBytes(), key, IV);

System.out.println("Encrypted Message :" + cipherText);

String decryptMessage = decrypt(cipherText, key, IV);

System.out.println("Decrypted Message :" + decryptMessage);

}

public static byte[] encrypt(byte[] message, SecretKey key, byte[] IV) throws Exception{

// Creating Cipher Instance

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

// Creating SecretKeySpec

SecretKeySpec keySpec = new SecretKeySpec(key.getEncoded(), "AES");

// Creating IvParameterSpec

IvParameterSpec ivSpec = new IvParameterSpec (IV);

// Initialize Cipher

cipher.init(Cipher.ENCRYPT\_MODE, keySpec, ivSpec);

// Encryption

byte[] cipherText = cipher.doFinal(message);

return cipherText;

}

public static String decrypt (byte[] cipherText, SecretKey key, byte[] IV) throws Exception{

// Creating Cipher Instance

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

// Creating SecretKeySpec

SecretKeySpec keySpec = new SecretKeySpec (key.getEncoded(), "AES");

// Creating IvParameterSpec

IvParameterSpec ivSpec = new IvParameterSpec(IV);

// Initialize Cipher

cipher.init(Cipher.DECRYPT\_MODE, keySpec, ivSpec);

// Decryption

byte[] decryptMessage = cipher.doFinal(cipherText);

return new String(decryptMessage);

}

}