**Program Code –**

import java.security.SecureRandom;

import javax.crypto.Cipher;

import javax.crypto.KeyGenerator;

import javax.crypto.SecretKey;

import javax.crypto.spec.SecretKeySpec;

import java.util.Random;

import java.util.\*;

public class DES {

byte[] skey = new byte[1000];

String skeyString;

static byte[] raw;

String inputMessage, encryptedData, decryptedMessage;

public DES() {

try {

generateSymmetricKey();

Scanner sc = new Scanner(System.in);

System.out.print("Enter Plaintext: ");

inputMessage = sc.nextLine();

byte[] ibyte = inputMessage.getBytes();

byte[] ebyte = encrypt(raw, ibyte);

String encryptedData = new String(ebyte);

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

byte[] dbyte = decrypt(raw, ebyte);

String decryptedMessage = new String(dbyte);

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

} catch (Exception e) {

System.out.println(e);

}

}

void generateSymmetricKey() {

try {

Random r = new Random();

int num = r.nextInt(1000);

String knum = String.valueOf(num);

byte[] knumb = knum.getBytes();

skey = getRawKey(knumb);

skeyString = new String(skey);

System.out.println("DES Symmetric Key: " + skeyString);

} catch (Exception e) {

System.out.println(e);

}

}

}

void generateSymmetricKey() {

try {

Random r = new Random();

int num = r.nextInt(1000);

String knum = String.valueOf(num);

byte[] knumb = knum.getBytes();

skey = getRawKey(knumb);

skeyString = new String(skey);

System.out.println("DES Symmetric Key: " + skeyString);

} catch (Exception e) {

System.out.println(e);

}

}

public static byte[] getRawKey(byte[] seed) throws Exception{

KeyGenerator kgen = KeyGenerator.getInstance("DES");

SecureRandom sr = SecureRandom.getInstance("SHA1PRNG");

sr.setSeed(seed);

kgen.init(56, sr);

SecretKey skey = kgen.generateKey();

raw = skey.getEncoded();

return raw;

}

private static byte[] encrypt(byte[] raw, byte[] clear) throws Exception{

SecretKeySpec skeySpec = new SecretKeySpec(raw, "DES");

Cipher cipher = Cipher.getInstance("DES");

cipher.init(Cipher.ENCRYPT\_MODE, skeySpec);

byte[] encrypted = cipher.doFinal(clear);

return encrypted;

}

private static byte[] decrypt(byte[] raw,byte[] encrypted) throws Exception{

SecretKeySpec skeySpec = new SecretKeySpec(raw, "DES");

Cipher cipher = Cipher.getInstance("DES");

cipher.init(Cipher.DECRYPT\_MODE, skeySpec);

byte[] decrypted = cipher.doFinal(encrypted);

return decrypted;

}

public static void main(String[] args) {

DES des = new DES();

}

}

private static byte[] decrypt(byte[] raw,byte[] encrypted) throws Exception{

SecretKeySpec skeySpec = new SecretKeySpec(raw, "DES");

Cipher cipher = Cipher.getInstance("DES");

cipher.init(Cipher.DECRYPT\_MODE, skeySpec);

byte[] decrypted = cipher.doFinal(encrypted);

return decrypted;

}

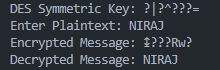
public static void main(String[] args) {

DES des = new DES();

}

}

**Output –**

****