import javax.crypto.Cipher;

import javax.crypto.KeyGenerator;

import javax.crypto.SecretKey;

import java.util.Base64;

import java.util.Scanner;

class DesEncrypter {

    private Cipher ecipher;

    private Cipher dcipher;

    private SecretKey key;

    // Constructor to generate and initialize key

    public DesEncrypter() throws Exception {

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

        this.ecipher = Cipher.getInstance("DES");

        this.dcipher = Cipher.getInstance("DES");

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

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

    }

    // Encrypt method

    public String encrypt(String str) throws Exception {

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

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

        return Base64.getEncoder().encodeToString(enc); // Convert to Base64

    }

    // Decrypt method

    public String decrypt(String str) throws Exception {

        byte[] dec = Base64.getDecoder().decode(str);

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

        return new String(utf8, "UTF8");

    }

}

    class DES {

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

        Scanner scanner = new Scanner(System.in);

        DesEncrypter encrypter = new DesEncrypter(); // Automatically generates a key

        // Get user input

        System.out.print("Enter text to encrypt: ");

        String plaintext = scanner.nextLine();

        // Encrypt and decrypt

        String encrypted = encrypter.encrypt(plaintext);

        System.out.println("Encrypted Text: " + encrypted);

        String decrypted = encrypter.decrypt(encrypted);

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

        scanner.close();

    }

}