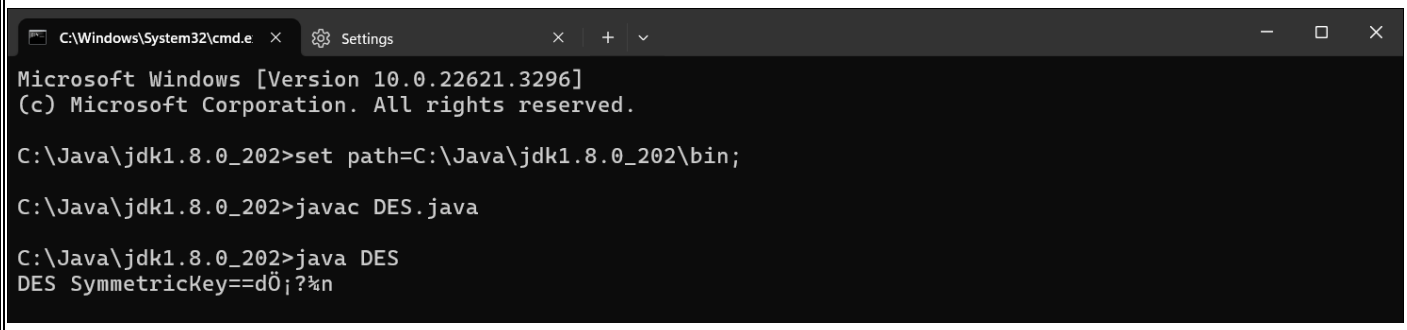
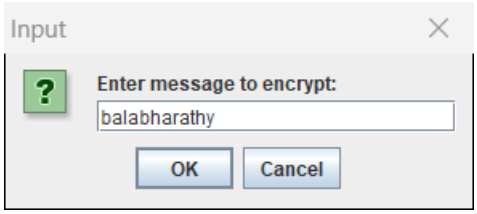
PROGRAM:

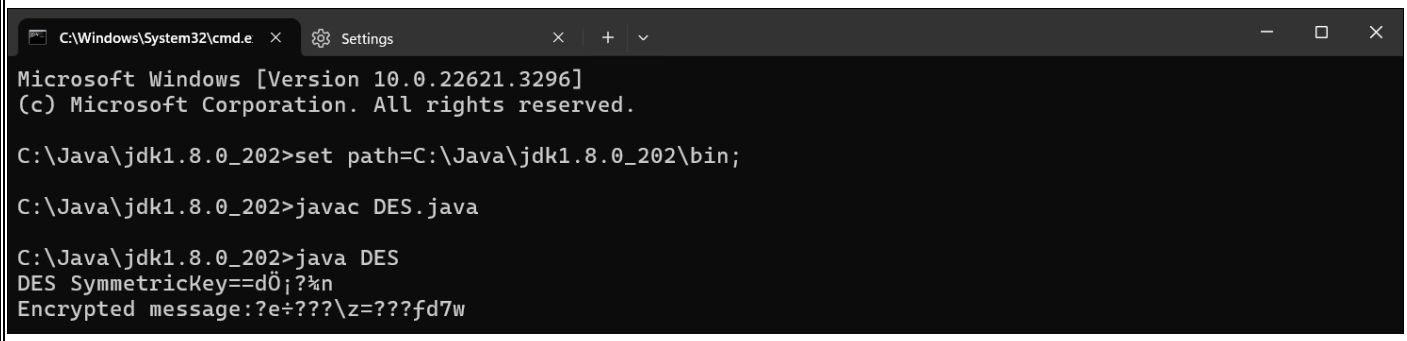
```
import javax.swing.*;
import javax.crypto.Cipher;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;
import java.security.SecureRandom;
import java.util.Random;

class DES {
   byte[] skey = new byte[1000];
   String skeystring;
   static byte[] raw;
   String inputmessage, encryptedata, decryptedmessage;

   public DES() {
      try {
         generatesymmetrickey();
         inputmessage = JOptionPane.showInputDialog(null, "Enter message to encrypt:");
         byte[] ibyte = inputmessage.getBytes();
         byte[] ebyte = encrypt(raw, ibyte);
         String encrypteddata = new String(ebyte);
         System.out.println("Encrypted message:" + encrypteddata);
         JOptionPane.showMessageDialog(null, "Encrypted Data" + "\n" + encrypteddata);
         byte[] dbyte = decrypt(raw, ebyte);
         String decryptedmessage = new String(dbyte);
         System.out.println("Decrypted message:" + decryptedmessage);
         JOptionPane.showMessageDialog(null, "Decrypted Data " + "\n" +
decryptedmessage);
      } catch (Exception e) {
         System.out.println(e);
      }
   }

   void generatesymmetrickey() {
      try {
         Random r = new Random();
         int num = r.nextInt(10000);
         String knum = String.valueOf(num);
         byte[] knumb = knum.getBytes();
         skey = getRawKey(knumb);
         skeystring = new String(skey);
         System.out.println("DES SymmetricKey=" + skeystring);
      } catch (Exception e) {
```

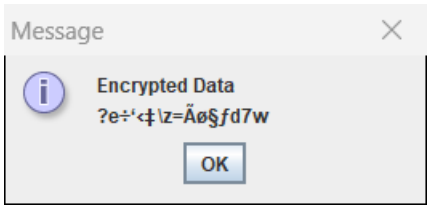
```
            System.out.println(e);
        }
    }

    private 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 {
        SecretKey seckey = new SecretKeySpec(raw, "DES");
        Cipher cipher = Cipher.getInstance("DES");
        cipher.init(Cipher.ENCRYPT_MODE, seckey);
        byte[] encrypted = cipher.doFinal(clear);
        return encrypted;
    }

    private static byte[] decrypt(byte[] raw, byte[] encrypted) throws Exception {
        SecretKey seckey = new SecretKeySpec(raw, "DES");
        Cipher cipher = Cipher.getInstance("DES");
        cipher.init(Cipher.DECRYPT_MODE, seckey);
        byte[] decrypted = cipher.doFinal(encrypted);
        return decrypted;
    }

    public static void main(String args[]) {
        DES des = new DES();
    }
}
```

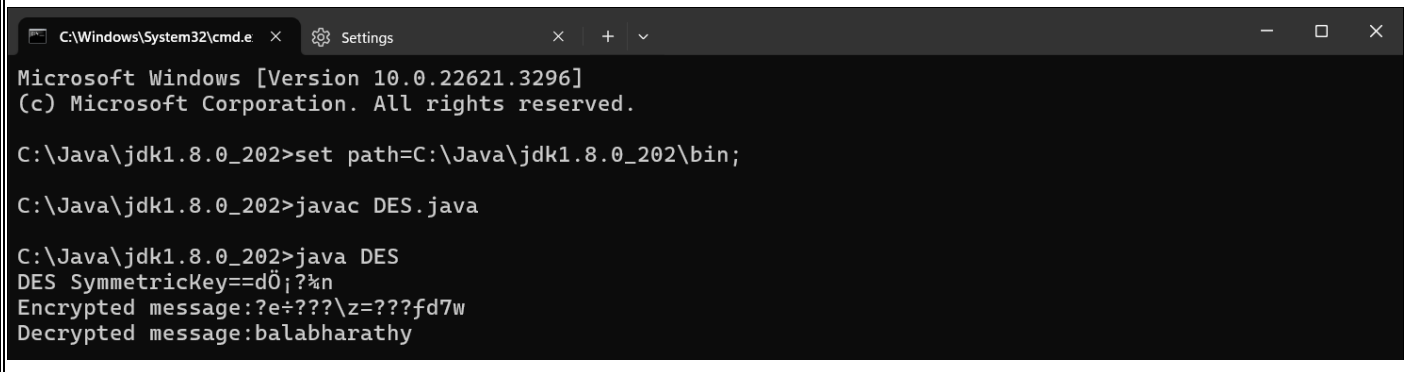
OUTPUT:

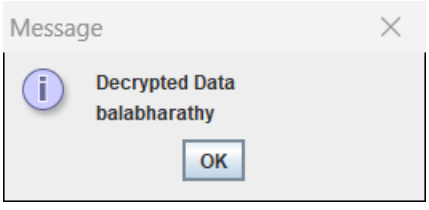

C:\Windows\System32\cmd.e
Settings
Microsoft Windows [Version 10.0.22621.3296]
(c) Microsoft Corporation. All rights reserved.
C:\Java\jdk1.8.0_202>set path=C:\Java\jdk1.8.0_202\bin;
C:\Java\jdk1.8.0_202>javac DES.java
C:\Java\jdk1.8.0_202>java DES
DES SymmetricKey==dÖ¡?%n


Input
Enter message to encrypt:
balabharathy
OK
Cancel


C:\Windows\System32\cmd.e
Settings
Microsoft Windows [Version 10.0.22621.3296]
(c) Microsoft Corporation. All rights reserved.
C:\Java\jdk1.8.0_202>set path=C:\Java\jdk1.8.0_202\bin;
C:\Java\jdk1.8.0_202>javac DES.java
C:\Java\jdk1.8.0_202>java DES
DES SymmetricKey==dÖ¡?%n
Encrypted message:?e÷???\z=???fd7w


Message
Encrypted Data
?e÷'‹‡\z=Ãø§fd7w
OK


C:\Windows\System32\cmd.e
Settings
Microsoft Windows [Version 10.0.22621.3296]
(c) Microsoft Corporation. All rights reserved.
C:\Java\jdk1.8.0_202>set path=C:\Java\jdk1.8.0_202\bin;
C:\Java\jdk1.8.0_202>javac DES.java
C:\Java\jdk1.8.0_202>java DES
DES SymmetricKey==dÖ¡?%n
Encrypted message:?e÷???\z=???fd7w
Decrypted message:balabharathy


Message
Decrypted Data
balabharathy
OK

PROGRAM:

```
import java.io.UnsupportedEncodingException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.Arrays;
import java.util.Base64;
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;
public class AES
{
  private static SecretKeySpec secretKey;
  private static byte[] key;
  public static void setKey(String myKey) {
    MessageDigest sha = null;
    try {
      key = myKey.getBytes("UTF-8");
      sha = MessageDigest.getInstance("SHA-1");
      key = sha.digest(key);
      key= Arrays.copyOf(key, 16);
      secretKey= new SecretKeySpec(key, "AES");
    } catch (NoSuchAlgorithmException e) {
      e.printStackTrace();
    } catch (UnsupportedEncodingException e) {
      e.printStackTrace();
    }
  }
  public static String encrypt(String strToEncrypt, String secret) {
    try {
      setKey(secret);
      Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
      cipher.init(Cipher.ENCRYPT_MODE, secretKey);
      return Base64.getEncoder().encodeToString(cipher.doFinal(strToEncrypt.getBytes
("UTF-8")));
    } catch (Exception e) {
      System.out.println("Error while encrypting: " + e.toString());
    }
    return null;
  }
  public static String decrypt(String strToDecrypt, String secret) {
    try {
      setKey(secret);
      Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5PADDING");
      cipher.init(Cipher.DECRYPT_MODE, secretKey);
      return new String(cipher.doFinal(Base64.getDecoder().decode(strToDecrypt)));
```

```
    } catch (Exception e) {
      System.out.println("Error while decrypting: " + e.toString());
    }
    return null;
  }
  public static void main(String[] args) {
    System.out.println("Enter the secret key: ");
    String secretKey= System.console().readLine();
    System.out.println("Enter the original URL: ");
    String originalString= System.console().readLine();
    String encryptedString = AES.encrypt(originalString, secretKey);
    String decryptedString = AES.decrypt(encryptedString, secretKey);
    System.out.println("URL Encryption Using AES Algorithm\n ---------- ");
    System.out.println("Original URL : " + originalString);
    System.out.println("Encrypted URL : " + encryptedString);
    System.out.println("Decrypted URL : " + decryptedString);
  }
}
```

OUTPUT:

```
C:\Java\jdk1.8.0_202>javac AES.java

C:\Java\jdk1.8.0_202>java AES
Enter the secret key:
annaUniversity
Enter the original URL:
www.annauniv.edu
URL Encryption Using AES Algorithm
 -----------
Original URL : www.annauniv.edu
Encrypted URL : vibpFJW6Cvs5Y+L7t4N6YWWe07+JzS1d3CU2h3mEvEg=
Decrypted URL : www.annauniv.edu
```