Program and Output

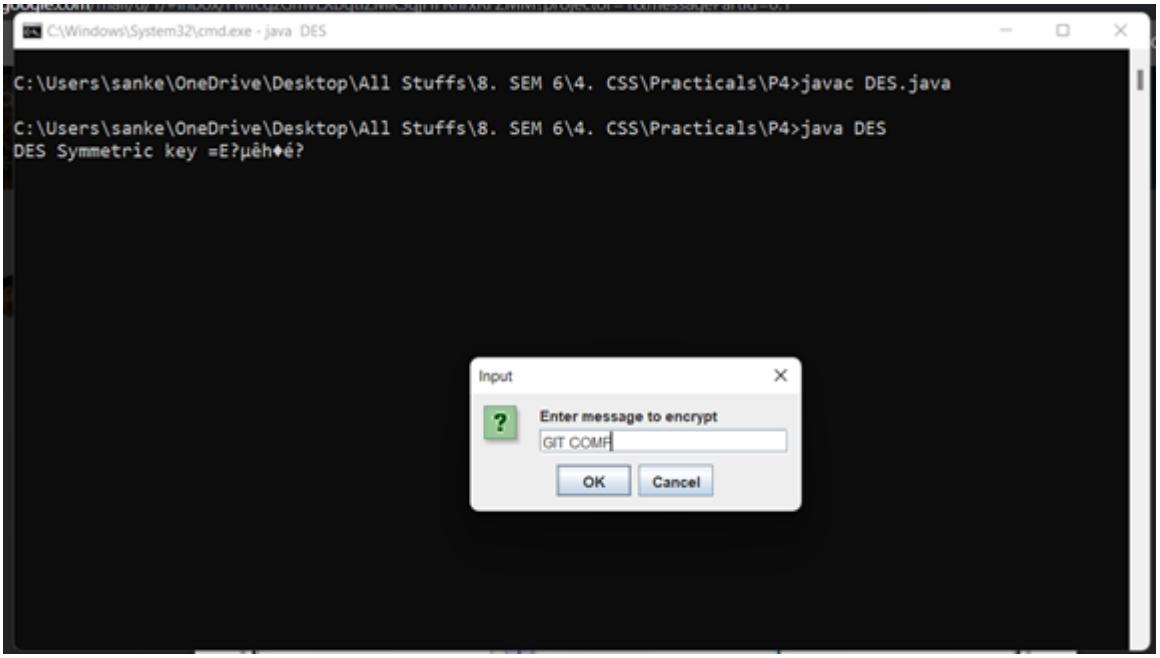
Program :

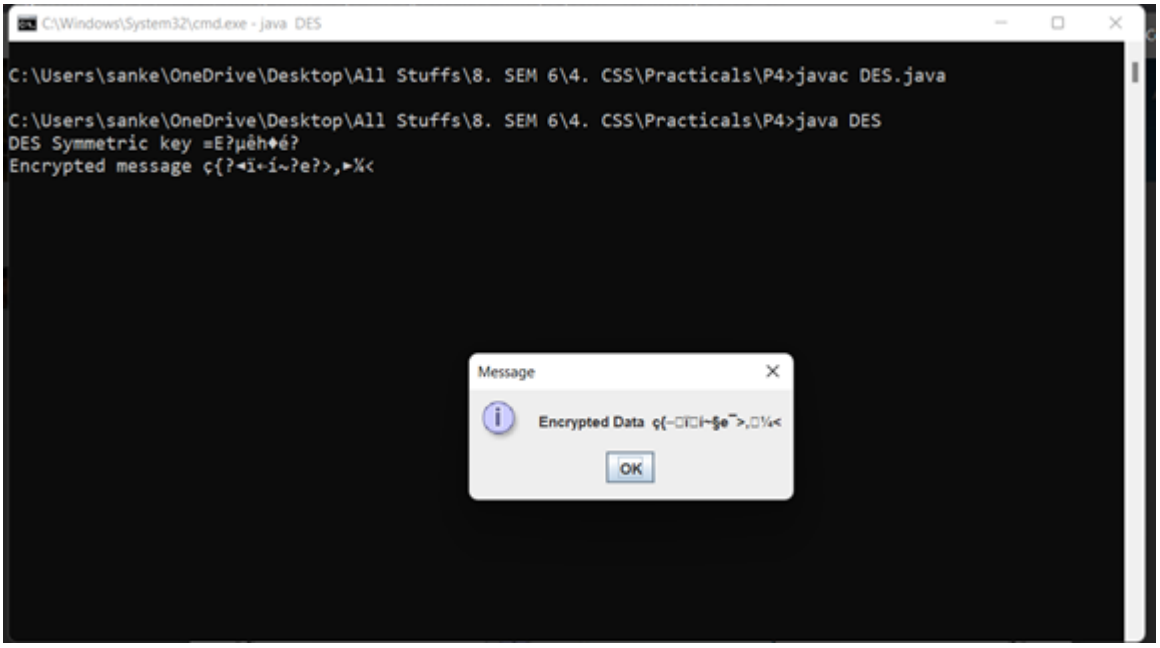
```
import javax.swing.*;
import java.security.SecureRandom;
import javax.crypto.Cipher;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;
import java.util.Random;

class DES {
	byte[] skey = new byte[1000];
	String skeyString;
	static byte[] raw;
	String inputMessage, encryptedData, 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 " + " " + encryptedData);
			byte[] dbyte = decrypt(raw, ebyte);
			String decryptedMessage = new String(dbyte);
			System.out.println("Decrypted message " +
decryptedMessage);
			JOptionPane.showMessageDialog(null, "Decrypted
Data " + " " + decryptedMessage);
		} catch (Exception e) {
```

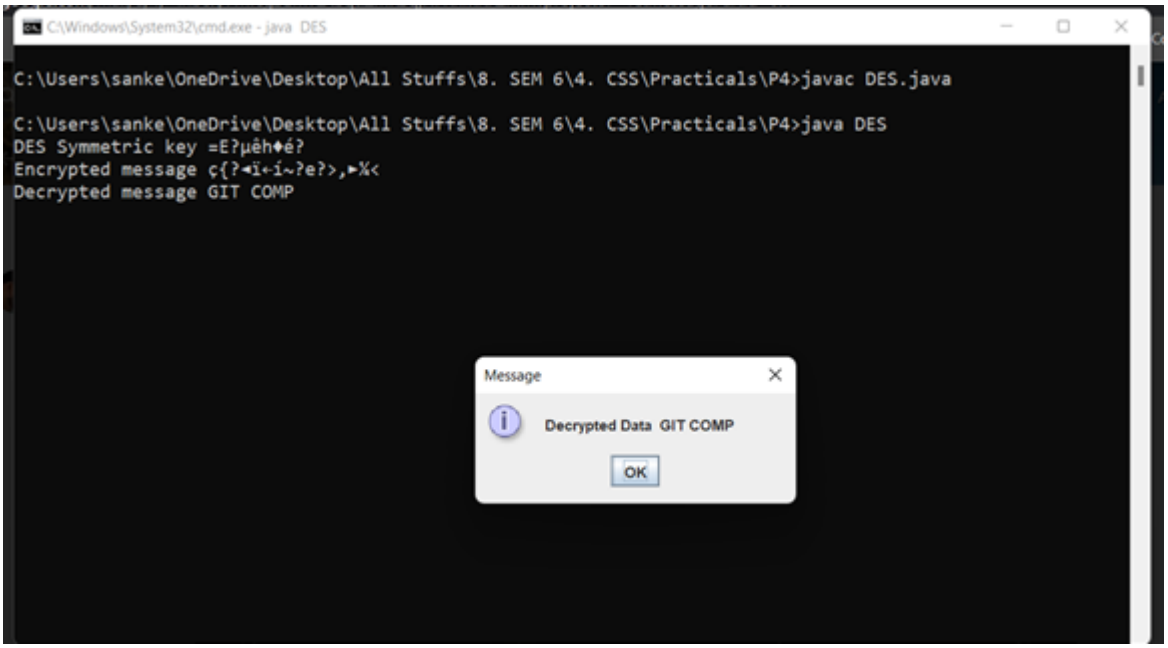
```
                System.out.println(e);
            }
        }

        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 Symmetric key =" +
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 {
            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();
        }
}
```

Output :




C:\Windows\System32\cmd.exe - java DES
C:\Users\sanke\OneDrive\Desktop\All Stuffs\8. SEM 6\4. CSS\Practicals\P4>javac DES.java
C:\Users\sanke\OneDrive\Desktop\All Stuffs\8. SEM 6\4. CSS\Practicals\P4>java DES
DES Symmetric key =E?µêh♦é?
Encrypted message ç{?◄ï←í~?e?>,►¾<
Message
Encrypted Data ç{–□□ı–§e¯>,□¼<
OK


C:\Windows\System32\cmd.exe - java DES
C:\Users\sanke\OneDrive\Desktop\All Stuffs\8. SEM 6\4. CSS\Practicals\P4>javac DES.java
C:\Users\sanke\OneDrive\Desktop\All Stuffs\8. SEM 6\4. CSS\Practicals\P4>java DES
DES Symmetric key =E?µêh♦é?
Encrypted message ç{?◄ï←í~?e?>,►¾<
Decrypted message GIT COMP
Message
Decrypted Data GIT COMP
OK