Name:Mohan Ramchandra Patil
Reg. No : 19141267

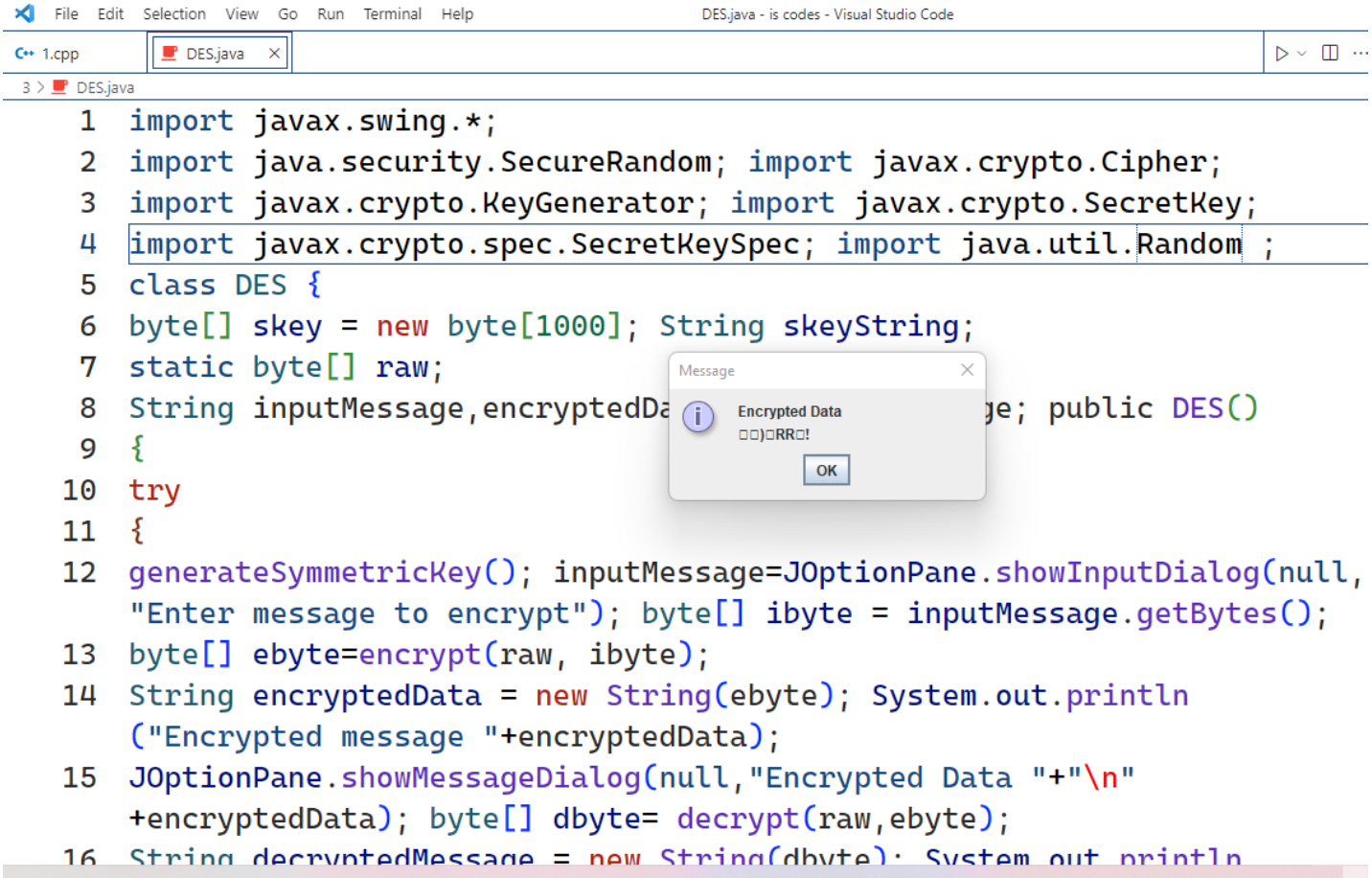
IS Exp3- Implementation of DES ALGORITHM

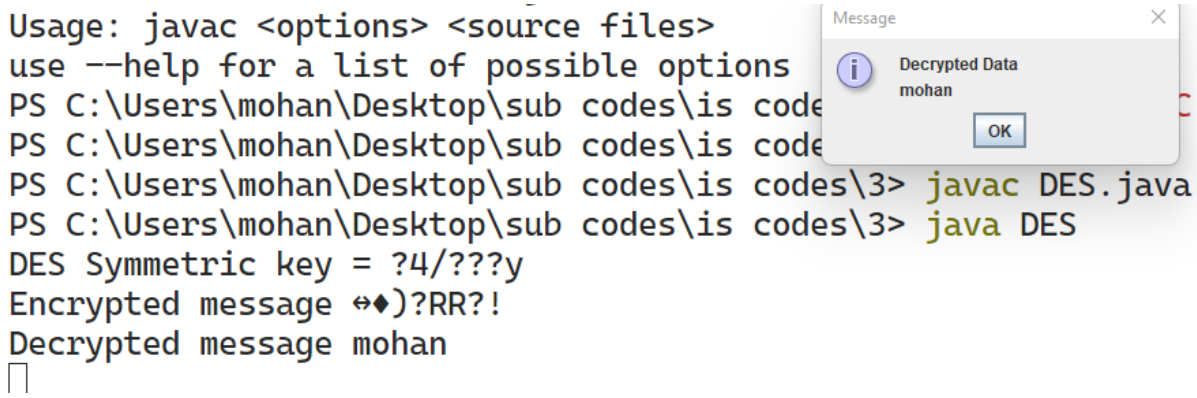
Code:

```
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 "+"\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 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:


File Edit Selection View Go Run Terminal Help
DES.java - is codes - Visual Studio Code
1.cpp
DES.java
3 > DES.java
1 import javax.swing.*;
2 import java.security.SecureRandom; import javax.crypto.Cipher;
3 import javax.crypto.KeyGenerator; import javax.crypto.SecretKey;
4 import javax.crypto.spec.SecretKeySpec; import java.util.Random ;
5 class DES {
6 byte[] skey = new byte[1000]; String skeyString;
7 static byte[] raw;
8 String inputMessage,encryptedDa ge; public DES()
9 {
10 try
11 {
12 generateSymmetricKey(); inputMessage=JOptionPane.showInputDialog(null,
"Enter message to encrypt"); byte[] ibyte = inputMessage.getBytes();
13 byte[] ebyte=encrypt(raw, ibyte);
14 String encryptedData = new String(ebyte); System.out.println
("Encrypted message "+encryptedData);
15 JOptionPane.showMessageDialog(null,"Encrypted Data "+"\n"
+encryptedData); byte[] dbyte= decrypt(raw,ebyte);
16 String decryptedMessage = new String(dbyte); System.out.println
Message
Encrypted Data
□□)□RR□!
OK


Usage: javac <options> <source files>
use --help for a list of possible options
PS C:\Users\mohan\Desktop\sub codes\is code
PS C:\Users\mohan\Desktop\sub codes\is code
PS C:\Users\mohan\Desktop\sub codes\is codes\3> javac DES.java
PS C:\Users\mohan\Desktop\sub codes\is codes\3> java DES
DES Symmetric key = ?4/???y
Encrypted message ↔♦)?RR?!
Decrypted message mohan
Message
Decrypted Data
mohan
OK