1. import java.io.UnsupportedEncodingException;
2. import java.security.MessageDigest;
3. import java.security.NoSuchAlgorithmException;
4. import java.util.Arrays;
5. import javax.crypto.Cipher;
6. import javax.crypto.spec.SecretKeySpec;
7. import org.apache.commons.codec.binary.Base64;
8. /\*\*
9. Aes encryption
10. \*/
11. **public** **class** AES
12. {
14. **private** **static** SecretKeySpec secretKey ;
15. **private** **static** **byte**[] key ;
17. **private** **static** String decryptedString;
18. **private** **static** String encryptedString;
20. **public** **static** **void** setKey(String myKey){

23. MessageDigest sha = **null**;
24. **try** {
25. key = myKey.getBytes("UTF-8");
26. System.out.println(key.length);
27. sha = MessageDigest.getInstance("SHA-1");
28. key = sha.digest(key);
29. key = Arrays.copyOf(key, 16); // use only first 128 bit
30. System.out.println(key.length);
31. System.out.println(**new** String(key,"UTF-8"));
32. secretKey = **new** SecretKeySpec(key, "AES");

35. } **catch** (NoSuchAlgorithmException e) {
36. // TODO Auto-generated catch block
37. e.printStackTrace();
38. } **catch** (UnsupportedEncodingException e) {
39. // TODO Auto-generated catch block
40. e.printStackTrace();
41. }


45. }
47. **public** **static** String getDecryptedString() {
48. **return** decryptedString;
49. }
50. **public** **static** **void** setDecryptedString(String decryptedString) {
51. AES.decryptedString = decryptedString;
52. }
53. **public** **static** String getEncryptedString() {
54. **return** encryptedString;
55. }
56. **public** **static** **void** setEncryptedString(String encryptedString) {
57. AES.encryptedString = encryptedString;
58. }
59. **public** **static** String encrypt(String strToEncrypt)
60. {
61. **try**
62. {
63. Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
65. cipher.init(Cipher.ENCRYPT\_MODE, secretKey);

68. setEncryptedString(Base64.encodeBase64String(cipher.doFinal(strToEncrypt.getBytes("UTF-8"))));
70. }
71. **catch** (Exception e)
72. {
74. System.out.println("Error while encrypting: "+e.toString());
75. }
76. **return** **null**;
77. }
78. **public** **static** String decrypt(String strToDecrypt)
79. {
80. **try**
81. {
82. Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5PADDING");
84. cipher.init(Cipher.DECRYPT\_MODE, secretKey);
85. setDecryptedString(**new** String(cipher.doFinal(Base64.decodeBase64(strToDecrypt))));
87. }
88. **catch** (Exception e)
89. {
91. System.out.println("Error while decrypting: "+e.toString());
92. }
93. **return** **null**;
94. }
95. **public** **static** **void** main(String args[])
96. {
97. **final** String strToEncrypt = "My text to encrypt";
98. **final** String strPssword = "encryptor key";
99. AES.setKey(strPssword);
101. AES.encrypt(strToEncrypt.trim());
103. System.out.println("String to Encrypt: " + strToEncrypt);
104. System.out.println("Encrypted: " + AES.getEncryptedString());
106. **final** String strToDecrypt = AES.getEncryptedString();
107. AES.decrypt(strToDecrypt.trim());
109. System.out.println("String To Decrypt : " + strToDecrypt);
110. System.out.println("Decrypted : " + AES.getDecryptedString());
112. }
114. }