Object Oriented Programming - Lab 6

Date : 06-10-2021

Topics: Packages, final, Java String API

1. Write a Java program with the following classes:
   - ➢ Class MyEncryption - a ***non-inheritable class*** that provides implementation of simple encryption methods.
     - **given in package *EncryptionSource***

     Fields:
     - int key (should be a constant)

     Methods:
     - String ceaserCipher(String plainText, int key)
       - Given plainText, returns cipherText where each letter of a given text is replaced by a letter some fixed number of positions down the alphabet. For example with a shift of 1, A would be replaced by B, B would become C, and so on.
       - Example: given on page 2
       - key is the number of characters to shift the cipher alphabet.
       - You are not allowed to use array of characters for implementation of logic for this method
     - String ceaserCipher(String plainText)
       - Overloaded ceaserCipher method where the field key is used as the shift value.
       - You are not allowed to use array of characters for implementation of logic for this method
     - String transpositionCipher(String plainText)
       - Given plainText, returns cipherText by writing the plainText row-wise in a matrix and reading it column-wise.
       - Example - given on page 2
       - For this you may use character array if necessary.
     - String findEncryption(String plainText, String cipherText)
       - Given a plainText, cipherText pair, find the encryption algorithm used. The return values can be Caesarcipher, transpostioncipher, or invalid encryption.

   - ➢ Class MyEncryptionMain - driver class for testing MyEncryption
     - **given in package *EncryptionDriver***

     Methods:
     - main

    - Given a plainText, print its encrypted value using caesar and transposition cipher
    - Given plainText, CipherText pair print the name of the encryption used.

Note:

- Make use of the String API to implement your logic. It is referred from: https://docs.oracle.com/en/java/javase/16/docs/api/java.base/java/lang/String.html
- For demonstrating the package usage, run your program in the command prompt and take screenshots of the compilation and execution steps.
- Use appropriate access modifiers for classes and members.
- Define appropriate getter and setter wherever necessary.
- Do not take any input from the user. Hardcode the input values

To be uploaded:

- Two java files **rollNumber_MyEncryptionMain.java** and **rollNumber_MyEncryption.java**
- Screenshot(s) of the output (jpg/png)
    - Try to show all output in a single screenshot if possible, otherwise use two
- Do not zip

Examples:

Caesar Cipher

To pass an encrypted message from one person to another, it is first necessary that both parties have the 'key' for the cipher, so that the sender may encrypt it and the receiver may decrypt it. For the caesar cipher, the key is the number of characters to shift the cipher alphabet.

Here is a quick example of the encryption and decryption steps involved with the caesar cipher. The text we will encrypt is 'defend the east wall of the castle', with a shift (key) of 1.

plaintext: defend the east wall of the castle

ciphertext: efgfoe uif fbtu xbmm pg uif dbtumf

It is easy to see how each character in the plaintext is shifted up the alphabet. Decryption is just as easy, by using an offset of -1.

plain: attack at dawn
key: 4
cipher: exxego ex hear

Obviously, if a different key is used, the cipher alphabet will be shifted a different amount.

Transposition Cipher

It is another type of cipher where the order of the alphabets in the plaintext is rearranged to create the ciphertext. The actual plaintext alphabets are not replaced.

An example is a ‘simple columnar transposition’ cipher where the plaintext is written horizontally with a certain alphabet width. Then the ciphertext is read vertically as shown.

For example, the plaintext is “golden statue is in eleventh cave” and the secret random key chosen is “five”. We arrange this text horizontally in table with number of column equal to key value. The resulting text is shown below.

g	o	l	d	e
n	s	t	a	t
u	e	i	s	i
n	e	l	e	v
e	n	t	h	c
a	v	e		

The ciphertext is obtained by reading column vertically downward from first to last column. The ciphertext is ‘gnuneaoseenvltiltedasehetivc’.