# Assignment 7: File Input, 2D arrays, String Processing

## Learning Objectives

* Write a java program using objects
* Use 2D arrays, and String data structures
* Write a program that reads input from files.

## Evaluation

* Solutions which do not compile will receive an automatic 0.
* Refer to the CS 176 Grading Criteria.pdf for descriptions of the criteria for each below.
	+ 1a, 1b. Separation of Concern (classes, methods) – 10 points
	+ 2a, 2b. Design for reuse(classes, methods) – 10 points
	+ 3.Design only what is needed – 20 points
	+ 5.Method comments- 5 points
	+ 6. Readability – 5 points
	+ 7a, 7b – Testing – 20 points

Overview

In this assignment you will develop a program that encrypts data using a Vigenère (pronounced vee-zhen-air) cipher which works as follows:

* We require a matrix which maps letters to other letters for example something like this: 
	+ Given a message such as “ATTACK AT DAWN” and a key “LEMON” the cipher is computed by
		- First the message and the key are aligned (repeating the key as necessary) as shown

|  |  |
| --- | --- |
| Plaintext: | ATTACKATDAWN |
| Key: | LEMONLEMONLE |
| Ciphertext: | LXFOPVEFRNHR |

* + - Then each position of the cipher text is computed column by column. The ith letter of the cipher is the letter appearing in matrix[row][col] where row is the ith letter of the message and col is the ith-letter of the repeated key. In the example above since the first letter of the message is A row = A, and since the first letter of the repeated key is L col =L, and since matrix[A][L] = L the first letter of the cipher text is ‘A’. For the 2nd letter of the cipher, row = T and col = E, matrix[T][E] = x, so the 2nd letter of the cipher is ‘X’.

## Your Tasks

Design a java class called Vigenere.java that does the following listed below. Take some time to design your class to incorporate encapsulation, separation of concern, design for reuse and design only what is needed.

1. The program reads a file lettermapping.txt which is formatted as follows:
	* 1. The file contains 26 lines. Each line contains 26 characters as separate tokens delineated by spaces. For example, the file might look like the inside part of the matrix shown above without the first row and column.
		2. Note: Your encryption/decryption should output different results based on the letter mappings given in this file. Assume I will be testing your code with different lettermappings.
2. The program should store the tokens found on each text line in a 2D array. The tokens can be stored as Strings or chars.
3. The program should have a public method String encrypt(String m, String k) where m is the plain text message and k is the key.
4. The method should use the letter mapping read in from the file, along with the key to encrypt the plain text m.
5. The method should return the cipher text produced as a String in all lowercase.
6. m,k may be given in upper or lower case.
7. The program has a public method void printFrequencies(String c) that takes a String and outputs the frequency of each letter appearing in the String. Thus for example if the string is “HELLO” the method should output:

H 1

E 1

L 2

O 1

1. Extra credit +10: Contains a method String decrypt(String c, String k) where c is a cipher text message and k is a key. The method should use the letter mappings read in from the file and the key to decrypt the cipher text c and return the plain text produced as a String.
	* There is no partial credit on this – points are earned if it passes my tests
2. In a main method write code to unit test the methods in your class.
3. Submit your Java source code file to canvas.