

To be submitted via Piazza in the folder "bonus lab" or in Blackboard depending on instructor.

## Welcome to the Bonus Lab!

This is a bonus lab, which means you are not required to submit it, but if you do, it may improve your current lab grades.

During the last week of classes you will review most of the concepts learned so far, this lab will help you to prepare for the final exam.



You should expect to work about 6 to 8 extra hours outside the lab session to complete this assignment.

It means that you need to make sure that Java works on your own computer, or that you go to open labs to work some more.

Extra time on labs includes completing the activities and taking the time to make sure that your submission is picture perfect!

### Keeping information secure...

In this lab, you will implement a system that allows a user to encrypt and decrypt text using a numeric key defined by the user. For simplicity, the system will only convert letters of the alphabet in a given text and replace them with numbers. The building block of our encryption system is a character. A character contains the following information:

- The character itself, e.g., "A", "a";
- The number corresponding number, e.g., 10.

A	a	E	e	I	i	O	o	U	u
1	2	3	4	5	6	7	8	9	10

Each letter in the message to be encrypted will be replaced by the corresponding number and surrounded by dots ".". For example, using the keys of the characters above, the sentence "Very secure message" will be replaced with "V.4.ry s.4.c.10.r.4. m.4.ss.2.g.4."

## Activity 1.

A character can be represented as a user-defined type (a.k.a. object): **Letter**. You will implement this new type, with the above attributes, and corresponding methods: i) constructor(s), ii) accessor(s), a.k.a. getter(s), one per attribute, iii) mutator(s), a.k.a. setter(s), one per attribute, and a print method that prints the information in any current **Letter**, in the file **Letter.java** that is provided to you.

The print method for an object whose attributes are as follows:

character: "A"

number: 1

should print the following:

The character "E" should be replaced with .1.

## Activity 2.

Then in another java file, **Encryption.java** that contains an Array of Letters. This is a user-defined type (a.k.a. object) that contains another user-defined type (a.k.a. object). You will implement this new type, with the Array attributes, and corresponding methods: i) constructor(s), ii) accessor, a.k.a. getter, iii) mutator, a.k.a. setter(s), one per attribute, and a print method that prints each one of the Letters in the Array. In addition, you will create the following methods:

1. A method, **readFromFile**, that takes a file name as a string – this file contains: i) the letters that will be replaced separated by comma in the first line, and ii) the numbers that will replace each letter separated by comma in the second line. This method should: i) handle file-reading errors via exceptions, ii) verify that it contains the same number of letters and numbers, and iii) that the numbers are not repeated.
2. A specific mutator (a.k.a. setter), **setLetter** that inserts the information of a new Letter in the array.
3. A specific accessor (a.k.a. getter), **getNumber** that receives a character and returns the number corresponding to the Letter containing said character.
4. A specific mutator (a.k.a. setter), **readKey** that uses some of the methods above to: i) read a file with letters and their keys, and ii) initializes the Array of Letters with each one of the Letter read from the file.
5. A **main** method that: i) reads a file with letters and numbers, ii) creates the Array of Letters, iii) and gives a user the following options in a menu:
  1. Read key
  2. Encrypt sentence
  3. Decrypt sentence.
  4. Exit

*Option 1 – Read key* will allow the user to provide the name of the file with the key.

*Option 2 – Encrypt*, will ask the user for a sentence and print the sentence encrypted.

*Option 3 – Decrypt*, will ask the user for a sentence and print the sentence decrypted.

*Option 4 – Exit*, will terminate the program.

What you have to turn in:

- In a docx file, write the pseudocode of each of the above methods as well as a clear description of the new classes Letter and Encryption: remember that a good pseudocode can be handed out to a programmer without further explanation and the programmer should be able to translate it to code.
- In this same docx file, you will describe and justify each of the exceptions you handle in your code.
- In the two java files that were provided to you, complete the description of the methods as described above and according to your pseudocode + include in Encryption.java's main method the lines of code that are necessary to test the above methods.

Important notes:

- If your code is not properly indented, it risks to lose up to 10 points out of your grade. For code indentation, follow the guidelines available at: <http://www.oracle.com/technetwork/java/javase/documentation/codeconventions-136091.html>.
- Spend time working on your pseudocode as the amount of points you get for the pseudocode is bigger than the amount of points you get for your code (usually, close to a 60/40 ratio).

That's all! Looking forward to seeing you in lab!