Delta College

CST 183
Programming Assignment 5

Fall 2019
Instructor: T. Klingler

Objective

To build a complete working Java program that offer practice with Java string manipulation.

Overview & Instruction

Write a Java program that performs encryption of a given prioritized message.

The chosen encryption routine requires a one-word key known by both the sender and receiver. The letters of the key are used to shift the characters of the message. For example if the key is: **MATH**, and the message is: **Delta College is open.**, the encryption would be:

DELTACOLLEGEISOPEN	Capitalize the message and remove spaces and punctuation.
MATHMATHMATHMATHMA	Repeat the key as needed and align with characters of message.
PEEAMCHSXEZLUSHWQN	The encrypted message. Note that '**A**' shifts zero positions, '**B**' shifts one position, and so on through '**Z**' that shifts 25 positions. If the shifting rolls past the end of the alphabet, then it must "wrap around" (i.e. '**Z**' shifts next to '**A**', etc.)

You are only responsible for the encryption routine. Assume another team is handling the decryption.

The behavior of your program should include a simple input via a dialog box. This could be an example:

Message: **P,Delta College is closed.**
Key: **MATH**

All messages should be prefaced with one of four possible characters priority codes:

Z - FLASH **O - IMMEDIATE** **P - PRIORITY** **R - ROUTINE**

These decrease in criticality as your read left-to-right. FLASH implies life or death urgency, while ROUTINE of course deals with messages in accords to its name.

Output from the example above there would be:

PRIORITY
PEEAMCHSXEZLUSHWQN

A legitimate message can only include one of the four valid characters, exactly one comma, and then one or more characters following the comma. Also, the key must be one capitalized word of at least four characters in length. Be sure to include an error message if these basic formatting requirements are not followed for the input strings.

Finally, design your solution using an object oriented approach. This implies a **`Message`** class that will include the ability to validate and manage the priority code, and to encrypt the message. A suggested design would be to pass or "set" the user entry into the **`Message`** object as it is being constructed.

Your main driver application class can then be narrowed to managing the user interaction as well as method calls to the one **`Message`** object in the solution. Utilize dialog boxes for both input and output.

Deliverables

Deliver the following to the online course management system **dropbox** as your final product:

- **Upload** your **source code** (.java) files

Notice

This is an individual assignment. You must complete this assignment on your own. You may not discuss your work in detail with anyone except the instructor. You may not acquire from any source (e.g., another student or an internet site), a partial or complete solution to a problem or project that has been assigned. You may not show another student your solution to an assignment. You may not have another person (current student, former student, tutor, friend, anyone) "walk you through" how to solve the assignment.
