Project #1

Your boss has come to ask for your help with another task. She would like you to write a simple program that encodes and decodes “secret messages” which she plans to demonstrate to elementary school students to kindle their interest in Computer Science.

The encoding/decoding scheme is very simple. Each letter is substituted by some other letter according to a given mapping as shown below.

```
// String letters = "abcdefghijklmnopqrstuvwxyz";
      String enc = "kngcadsxbvfhjtiumylzqropwe";
```

For example, every 'a' becomes a 'k' when encoding a text, and every 'k' becomes an 'a' when decoding.

You will write a program, SecretMessage.java, which asks the user whether they would like to encode or decode a message, and then encodes or decodes the message using the mapping above.

Capital letters are mapped the same way as the lower case letters above, but remain capitalized. For example, every 'A' becomes 'K' when encoding a text, and every 'K' becomes an 'A' when decoding. Numbers and other characters are not encoded and remain the same. The program repeatedly asks the user for input until the user quits.

Below is one run of the program:

```
Enter 1 to encode, 2 to decode, 3 to quit: 1
Enter the text to encode:
Santa Monica College Spring 2014
Lktzk Jitbgk Gihhasa Luybts 2014
Enter 1 to encode, 2 to decode, 3 to quit: 1
Enter the text to
encode: programming is
fun uyisykjjbts bl dqt
Enter 1 to encode, 2 to decode, 3 to quit: 2
Enter the text to decode:
Lktzk Jitbgk Gihhasa Luybts 2014
Santa Monica College Spring 2014
Enter 1 to encode, 2 to decode, 3 to quit:
3 Good bye
```

Hints:

Think about decomposing the problem into simpler problems. You will likely find it helpful to write a few very simple methods first:

- *isLowerCaseLetter* takes a char and returns true if that letter is a lower case letter (between 'a' and 'z'). *isUpperCaseLetter* and *isLetter* should work similarly.
- *toLowerCase* takes an upper case letter and returns the same letter in lower case. Do not use conditionals to implement this (if/else or switch). Instead, remember that a char is “a number” code. Print that number code for 'A', 'B', 'a', 'b' to see how it looks

and come up with a very simple way to change an upper case letter to lower case. *toUpperCase* should work similarly.

- *codeChar* takes a char and a boolean flag. If the flag is true, it returns the encoded char according to the given mapping. If the flag is false, it returns the decoded char according to the given mapping. Do not use conditionals to do the mapping. Instead, notice that the letter 'a' is encoded to the first character of the given encoding map. The letter 'b' to the second,
  and so on. For the decoding, you can write a decoding string by hand that you can use to map the encoded letters 'a' to 'z' to their decoded counterparts.
  Given a String in Java you can use the charAt method to get the letter at a given position. For example:

```
String s = "example";
char first = s.charAt(0);      // first is assigned 'e'
char second = s.charAt(1);     // second is assigned 'x'
```

  Notice that to get the character at the first position we pass the argument 0 to the charAt method and not 1.
- *codeMessage* takes a String and a boolean flag. If the flag is true, it prints the given message encoded. If the flag is false, it prints the given message decoded. This method just calls the *codeChar* method for every character in the String and prints the character that *codeChar* returns. It is really short and simple.
  Given a String in Java you can use the length method to get the number of characters in that String. For example:

```
String s = "hello";
int len = s.length();              // len is assigned 5
```

Deliverables:

You should submit a zip file named project1_first_last.zip (where first and last are your first and last name) containing **ONLY** the **1 files below.**

SecretMessage.java

How you lose points:

- If your code does not follow Java coding standards regarding naming of classes, variables, methods, constants, etc.
- If your code is not formatted properly and I find it hard to read.
- If your code is not commented properly:
- Every class file should have a header comment that includes your name and assignment number and briefly documents what the program does.
- If there are known deficiencies with your program such as known problems or incomplete features, these should be clearly listed in the header comment.
- Every method should have a method header comment that documents what the method does, what its parameters are, and what it returns.

- Every complex block of code should have at least a line comment documenting that code's intent.
- If your source code has print out statements you used for debugging or commented out code you used to try out ideas. Clean up your code before you submit. Do not leave clutter behind.