

# Lab01 - Secret message

Due: Fri Jan 9, 2015

For Lab 01, you'll find the files you need in this k: drive folder:

```
k:15WI/csc_161_2/common_area/lab01
```

OK. I have sent you an important (cough) secret message.

The message is in the file `secret_message.txt`.

You need to decode this message for Lab 01. Decode... and win!

Details:

- We'll work in pairs for more fun.
- The secret message uses a rotation cypher. It is encoded by rotating all the characters in the file by a fixed amount. For example, rotating the letter 'a' three positions gives 'd'. Rotating the word "Test" 3 positions gives "Whww". Make sense?
- So, you need a general-purpose character rotation decoder. You need to run it a few times (less than 26) and find the right rotation amount so that the message makes sense.
- What do you win? Who knows. Decode the message and find out. The winning team in Lab 01 will win a prize (fortune) and the envy of the rest of class (fame).
- Please implement your classes using the Decoder interface in the Lab 01 folder. That file is named `Decoder.java`. (Hmm. Have you used an `interface` before?)
- Things you might use in this lab (in other words, things I used):
  - `BufferedReader.readLine()` to read lines of text from a file
  - `PrintWriter.println()` to write lines to a file
  - `String.toCharArray()` to turn a `String` into an array of chars
  - The `String` ctor to build a `String` from an array of chars: `String( char [])`
  - `Scanner.nextLine()` and `Scanner.nextInt()` to get user input like our textbook does
  - Java's modulo operator, `%`
  - In Java, you can shift letters: `char ch = (char) ('a' + 3) // ch is 'd'`

We'll talk about this more in class.

Thanks, Bill

PS - We'll add a twist on Wednesday.

## The Wednesday Twist

If you are still working on your decoder, please continue on that.

If you've finished your decoder, please print it out and I'll review it. Please make sure that your code is beautiful.

### Q1 Can you turn your Decoder into an Encoder?

I have added an interface to the k: drive common\_area: Encoder.java.

### Q2 Can you automate your decoding process?

Is it possible to automate the process of choosing with rotation index (1 to 25) is correct to decode the message?

One possible hint: [en.wikipedia.org/wiki/Most\\_common\\_words\\_in\\_English](http://en.wikipedia.org/wiki/Most_common_words_in_English)