

CS 145: Preassignment 1

Rotex

1 Description

Your primary goal in this homework is to acquaint yourself with expressing your deepest algorithmic desires in Java. In this case, your deepest desires are fairly narrow—you wish to encode text in ROT13. This encoding rotates each alphabetic character in a text to the corresponding letter 13 positions later in the alphabet, wrapping around back to ‘a’ as necessary. The full mapping is:

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m

ROT13 is used in online forums to facilitate the discussion of plots or the exchange of riddles without revealing spoilers to casual readers. A similar encoding was used by Julius Caesar to encrypt secret messages.

As you implement this ROT13 encoder, you will learn about the indispensable **Scanner** and **String** classes. Consult their documentation to see how they can help you accomplish your task.

2 Requirements

To receive full credit for this assignment, you must satisfy the following requirements:

- Write all code in a class named **Rotex** in a package named **pre1**. Case and spelling matter. And they will all semester.
- Prompt the user to enter a message to encode with the prompt “Message: ”. Get a line of text from the user, encode it, and print it to the console entirely in lowercase, preceded by the text “Rotated: ”, and followed by a linebreak. Following is the interaction seen at the console for two runs of the program, with user input underlined. Your code must produce identical interactions for the same user input.

```
Message: abcdefghijklmnopqrstuvwxyz0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ
Rotated: noprstuvwxyabcdefghijklmnopqrstuvwxyz0123456789noprstuvwxyabcdefghijklmnopqrstuvwxyz
```

```
Message: Jo and Laurie would have been perfect together, but alas!
Rotated: wb naq ynhevr jbhyq unir orra cresrpg gbtrgure, ohg nynf!
```

- Correctly submit your code.

3 Suggested Process

Teaching a machine to do something can be challenging. Please follow this suggested process to getting working code:

1. Do some examples by hand first. On paper. Write down your steps in plain English as a first draft to your Java solution.
2. Read the `String` and `Scanner` documentation and book examples.
3. Translate your draft algorithm into Java, little by little. Write a line and test. Repeat.
4. Test your code yourself. Insert temporary print statements or learn to use the Eclipse debugger. The SpecChecker is intentionally not built to help pinpoint errors.

4 Files

- SpecChecker: twodee.org/teaching/cs145/2012A/homework/speccheck_pre1.jar

5 Submission

This is a preassignment, and preassignments are graded by SpecCheckers, not your instructor or TAs. The good news is you get to tweak your code until you have a perfect score. The bad news is that there's a little setup involved—and you must leave yourself enough time to work out kinks. **Deadlines are not extended for botched configurations.** Please follow these directions closely to test and submit your work:

1. Right-click on your project and choose Build Path → Add Libraries. Select JUnit, click Next, choose JUnit 4 from the dropdown menu, and click Finish. This only needs to be done once per project. So, if you use the same project for all your code in this class, you'll never need to execute this step again.
2. Download the SpecChecker JAR file.
3. Drag the JAR file onto your `pre1` package in Eclipse. Right-click on it, and select Build Path → Add to Build Path. It will migrate to the Referenced Libraries node in your project.
4. Run the JAR file by selecting it and hitting the Play button in the Eclipse toolbar. Run it as a Java application if prompted.
5. Address any errors. **Barring submission errors, the score you see will be the score you are assigned in the gradebook.**

6. Once you are satisfied with your score, choose a directory in which to save `pre1.zip`. This file contains a compressed version of `Rotex.java`. Open it in your favorite archiver utility to ensure it contains your code.
7. Drop your submission into `W:\c s\CJohnson\cs145\<YOUR-USERNAME>`. You may overwrite this file as often as you like before the deadline. Just make sure the directory contains only one `*.zip` file with lowercase `pre1` in its name.