String and Character Manipulation

Cryptography Lab

# Main Objectives

* Learn the basics of some of the early cryptography algorithms.
* Manipulate individual characters in a String to build new Strings.
* Count values in a String and store the results into an array.
* Use logical and mathematical operators on characters.

# Prior Knowledge

* Familiarity with String methods.
* Familiarity with looping structures.
* Familiarity with the char primitive type (not on the APCSA exam).
* Familiarity with basic primitive-array creation and modification.

# The Lack of Greenfoot Background

Although this lab is written with Greenfoot to perform the visualizations, students do not need to use any of the Greenfoot API. All of the visualization of data on the screen has been pre-provided. Students will be writing utility methods in a Cryptography class which could be used either by Greenfoot or any other Java file, including regular text-based console windows.

# Getting Started

Install the Greenfoot IDE: <https://www.greenfoot.org/download>

* Greenfoot went through many changes beginning with the 3.0.0 release. Version 2.4.2, although older, tends to be one of the most stable versions but is missing a few of the newer additions.

In your Cryptography folder run the project file with the Greenfoot icon to begin.

# Demonstration

View a sample demonstration of what the Cryptography lab could look like here:

<https://www.greenfoot.org/scenarios/23660>

# Instructions

The instructions are written directly in the Cryptography class as comments. Students will be writing seven methods: parseNonAlphaNumerics, shift, letterCount, numberCount, substitution, isLetter, and isNumber.

The Greenfoot lab itself already contains the needed graphical implementation of the program. The prewritten controls are as follows:

1. Hit the Run button to start the program.
2. The up and down arrows adjust how far the next Caesar shifts will be.
3. The left and right arrows apply the Caesar shift to the given String, shifting either in the positive or negative directions.
4. The enter key allows a new phrase to be typed in.
5. The spacebar allows a new alphabet to be typed in to apply the substitution cipher.

# Extensions and Modifications

* Students who wish a bit more of a challenge can adjust their program to work on any upper or lowercase letters. For instance, when shifting an uppercase letter, it would stay uppercase. When calculating how many ‘A’s are in a phrase, both upper and lowercase ‘A’s should thus count.
* The updateAmount method in the CryptoWorld is an excellent method for students to figure out on their own. The idea is simply to keep a field between 0 and 25 after adjusting it by a certain number.
* Other Cryptography algorithms can be easily applied, but will need a visual update to allow Greenfoot to use them. Students who wish to implement other algorithms can use the format shown in the CryptoWorld as examples of how to visually display their own extra algorithms. Example Crypto algorithms that students may write: A Vigenere or XOR Cipher, morse-code, pigpen cipher, or various others.

