This lab will give you a hands-on introduction to Java syntax through the use of an online practice tool named CodingBat.

1 Overview

CodingBat is a "free site of live coding problems to build coding skill in Java" and was created by Nick Parlante, a computer science lecturer at Stanford. The website provides users with a series of short-form coding exercises, and allows them to write and test code directly in a browser. The provided collection of test cases gives the coder with immediate feedback in regard to their correctness.

CodingBat provides exercises in both Java and Python which target specific programming and problem-solving concepts. It is an excellent tool for getting acclimated to a new language or practicing coding fundamentals, and we encourage you to make use of it in the first couple of weeks as you get acclimated to Java.

2 Setup

First, you will need to create a CodingBat account:

- Navigate your preferred web browser to www.codingbat.com
- Click on the create account link in the upper right hand corner of the page
- Follow the steps to create your account. Specifically, you must do the following:
  - use your @colgate.edu email as your registering email address
  - use a different password than your Colgate email account
  - register your name in the Last, First format

After you have created your account you should now log in. Once logged in, you will link your account to your respective instructor by following the steps below:

- From the CodingBat home page, click on prefs in the upper right corner
- In the Share To field at the bottom of the page enter the email address of your lab's instructor:
  - efourquet@colgate.edu for lab section A (Prof. Fourquet)
  - sjackson1@colgate.edu for labs sections B and C (Prof. Jackson)
  - mlyboult@colgate.edu for lab section D (Matt Lyboult)

Once you have linked your account to your instructor, you may begin working on the problems below.

3 Your Task

For this lab, you will need to complete several Java programming challenges broken up into three "blocks". You are encouraged to reference your notes, textbook and the online Java language documentation, i.e. Application Programming Interface (or API). In particular, review the various String methods available to you out of the API. Learning to use a language’s API is a critical skill to master in order to become a strong programmer. Additionally, Section 3.3
of your textbook provides a collection of commonly used \texttt{String} methods, including a summary table.

Your code, even if incomplete, is saved to your account each time you click the \texttt{Go} button, even if it contains syntax errors or doesn't pass all of the test cases. Additionally, if you log in from a different computer, CodingBat will automatically retrieve your code so long as you log into your account.

Below is a list of exercises you must complete; we will begin with a series of warm-up problems.

### 3.1 Warm-up

Complete the below exercises in the following order.

Note that the problems under the \texttt{Warmup} section have an option to display the solution. You are allowed to show the solution if you are stuck but \textbf{only after you have made an adequate attempt}. If you display the solution, it’s important that you trace and fully understand the solution code. You should then close the solution and solve the problem again using what you learned from referencing the solution.

Alternatively, if you are able to complete without showing the solution, you \textbf{must still} look at the provided solution code, and contrast your implementation’s design and style to the solution.

**Warmup-1**
- \texttt{loneTeen}
- \texttt{startHi}
- \texttt{everyNth}

**Warmup-2**
- \texttt{stringTimes}
- \texttt{doubleX}

**Before moving on** show your code to your instructor, so they can provide you with stylistic feedback on your work thus far. Once you have discussed with your instructor, you may continue to the following section.

### 3.2 Intermediate Exercises

Next, complete the following exercises:

**Logic-1**
- \texttt{in1To10}

**Logic-2**
- \texttt{loneSum}

**String-2**
- \texttt{endOther}
- \texttt{sameStarChar}
- \texttt{oneTwo}
3.3 Advanced Exercises

To make things a bit more exciting, for these final two problems you are not allowed to convert the argument `int` into a `String` at any point. The problems are:

AP-1

- hasOne
- dividesSelf

4 Submission

This assignment is due for all lab sections on **Monday, September 10th at 10:00PM**. You will not need to submit anything via Moodle; your instructor will be able to see timestamps indicating when you completed each of the assigned problems.