**Secret Code, Part 1**

Strings utilize many different aspects of programming: calling methods (dot notation), passing in parameters, indexing, return types, and more. We haven't looked in-depth at them for quite a while – let's review. **As the material in this lab is intended as a review, it might be useful to utilize the link to review the ‘1st year Java Topics’ located in canvas. (e.g. how to use the String class'** indexOf() **method).**

Before you begin, make sure you understand the following examples:

1. What would "computer science".substring(1) return?
2. What would "computer science".substring(6, 10) return?
3. What would "computer science".substring(6, 10).indexOf("r") return?
4. What would "computer science".substring(10).substring(2, 4).indexOf("n") return?
5. Create a new Project in BlueJ called “SecretCodeProject”. Make a class called StringMethods – this class has no instance variables, and should have a no-parameter (default) constructor**. All the exercises (methods) below will go in this class**. Class names always begin with an upper-case letter!
6. Make a class called Runner with a public static void main(String[] args) method. This class should be used to test all the methods in StringMethods.
7. In the Runner class, inside the main() method, create an object of type StringMethods (e.g. StringMethods test = new StringMethods()). Use this object to call the methods of the StringMethods class (e.g. test.unusualHello("Ron")).

**//the main() method is the 'start button' that Java looks for, to begin execution**

1. Complete the method public void unusualHello(String name), that will print a greeting and "you dummy!" added to the supplied nameparameter.

unusualHello("Bob") >>> Hello Bob, you dummy!

1. Complete the method public String stringRipper(String str), that will return a String containing only the first and last letters of str.

stringRipper("something") >>> "sg"

1. Complete the method public String concatenate(String a, String b, boolean space), that will return Strings a and b concatenated together. The boolean parameter space will be true if the user wants a space between the two words.

concatenate("computer", "science", true) >>> "computer science"

1. Complete the method public boolean hasSub(String str, String sub), that will return true if sub can be found in str (case sensitive) WITHOUT using the contains() method.

hasSub("Computer science", "sci") >>> true

1. Complete the method public boolean evenFooBar(String str), that will return true if "foo" and "bar" appear the same number of times in the String parameter str.

evenFooBar("foofoobarbarfoo") >>> false

evenFooBar("foobarbarfoo") >>> true

1. Complete the method public String microwaveTime(String str), that takes a String of numbers representing time and converts it to "microwave time" – adding a colon and returning a String in the format **minutes:seconds**. A zero should be added before Strings with two or less numbers. Strings with less than 1 or more than 4 numbers should return "invalid input".

microwaveTime("12") >>> "0:12"

microwaveTime("123") >>> "1:23"

microwaveTime("4500") >>> "45:00"

1. Complete the method public String reverseWords(String str), that will return a String where the order of each word in str has been reversed, word by word. The split() method may be useful for this.

reverseWords("hello world") >>> "world hello"

reverseWords("the sky is blue") >>> "blue is sky the"

1. Complete the method public int sumString(String str), that will scan str and sum all the integers that can be found in the String. Use a Scanner object to tokenize (parse) the input (check the "More on Scanners" powerpoint in canvas).

sumString("Hi 5 there are 2 or 3 numbers in this String") >>> 10