CS142 Introduction to Object Oriented Programming

S Bergin & C. O'Floinn

Lab 6: Blue J, Javadocs, Static Methods

---

1. Begin the lab by walking through this week's slides on Moodle.
2. Then create a project in BlueJ called 'Week7". Create a class called HelloWorld that contains two methods. The first method should be the main method and should call a method named `sayHello().` This method should accept a String that represents a person's name and prints Hello followed by the person's name to the screen.
3. Using the slides on Moodle as a guide add javadoc comments that specify the following:
    a. For the class provide a description of the class, the author and version details.
    b. For the method `sayHello()` specify a description of the method and the parameter accepted.
    c. View the documentation that is generated in BlueJ.

Create a class called `StringTester` that contains the following methods and generate Java documentation that specifies (1) a comment, the author and version of the class and (2) for each method below specifies a comment, the parameter(s) and return type if any.

Q1. Write a static method that determines if a String is a palindrome. The method should take a single String as a parameter and return a value of true or false. A String is a palindrome if it is lexicographically the same spelt in reverse. For example, Navan and Hannah are palindromes (ignore the case). Call the method from main and print an appropriate message to state if the String is a palindrome or not.

Q2. Write a static method that accepts two Strings as parameters. The method should determine if the Strings are anagrams. Two Strings are anagrams if they both contain exactly the same letters. For example, the following are anagrams:

Astronomer	*Moon starer*
The eyes	*They see*

You may assume the Strings only contain letters and spaces. A possible iterative solution to this is as follows:

Given two strings string1 and string2
Iterate over the characters in string1
If the character is not a space
Check if the current character of string1 is present in string2
*remove the character from string2 or change to a * etc*
Else
return false as they are not anagrams
If the iteration completes, then the two strings are anagrams.