Programming Assignment 9

forScore

`Shortening`

Overview

perspecTV is a company that designs, creates and markets products that provide a new perspective on television. Their products make television both more interactive and more informative.

They are in the process of developing a suite of products called forScore for judged competitions of various kinds (e.g., sporting events like diving and gymnastics, singing contests, and dance competitions). These products will be used by the organizers of the events, the broadcasters of the events, and the viewers/audience.

They have contracted with you to create a class named `Shortening` that will enable them to determine an event name from a user-provided shorter version (e.g., to determine that `"JMU"` is a shorter version of the event name `"JMU Invitational"`).

This project may lead to additional work in the future.

Specifications

The `Shortening` class must comply with the following specifications:

1. It must have the following four “class constants”:

```
public static final int MISSING = -1;

public static final int AT_THE_START = 0;

public static final int IN_THE_MIDDLE = 5;

public static final int AT_THE_END = 10;
```

2. It must have a method with the following signature:

```
public static int containsIgnoringCase(String shorter,
                                       String longer)
```

This method must determine if the `String` object named `longer` contains the `String` object named `shorter`, ignoring the case of both `String` objects. It must return (**in order of priority**): `AT_THE_START` if `longer` starts with `shorter`, `AT_THE_END` if `longer` ends with `shorter`, `IN_THE_MIDDLE` if `longer` contains `shorter` other than at the start or end, and `MISSING` otherwise. Note that multiple conditions can be satisfied. For example, the `String` object `"JMU vs JMU"` both starts with and ends with `"jmu"` (ignoring case). However, in this case, the method must return AT_THE_START since it has higher priority.

3. It must have a method with the following signature:

```
public static String longerFor(String shorter,
                               String[] longers)
```

   It must return the first element of the `String[]` named `longers` that contains (ignoring case) the `String` object named `shorter`. It must return the empty `String` (i.e., `""`) if no element of `longers` contains (ignoring case) `shorter`.

Note that the “longer” `String` need not actually have the same number as or more characters than the “shorter” `String`. Note also that you may assume that neither `String` will be empty (i.e., `""`) or `null`.

Recommended Process

1. Read and understand the entire assignment.
2. Design and implement the `containsIgnoringCase()` method.
3. Test the `containsIgnoringCase()` method (and debug it if necessary).
4. Design and implement the `longerFor()` method.
5. Test the `longerFor()` method (and debug it if necessary).
6. Submit your implementation of `Shortening.java` in a file named `pa8.zip` using Autolab. Do not include any other files in the `.zip` file.

A Note About Testing

Recall that we have made a distinction between unit testing (in which one module is tested at a time) and integration testing (in which multiple modules are tested together). You have used both in the past.

For this assignment, you should write unit tests for each method. The most convenient way to do this is to use `Test` class, but it is not required that you use the `Test` class and you should not submit the tests.

Hints

This assignment is principally about using objects, in particular, objects in the `String` class. Hence, you should make sure that you read the javadocs for the `String` class so that you know what methods belong to `String` objects. The main page for the most recent version of the javadocs are at:

https://docs.oracle.com/en/java/javase/15/docs/api/index.html

From there, you can search for the documentation for a particular class. For example, the javadocs for the String class are at:

https://docs.oracle.com/en/java/javase/15/docs/api/java.base/java/lang/String.html

Grading

Your code will first be graded by Autolab and then by the Professor. The grade you receive from Autolab is the maximum grade that you can receive on the assignment.

Autolab Grading

Your code must compile (in Autolab, this will be indicated in the section on "Does your code compile?"), and all class names and method signatures comply with the specifications (in Autolab, this will be indicated in the section on "Do your class names, method signatures, etc. comply with the specifications?") for you to receive any points on this assignment.

Autolab will then grade your submission as follows:

Conformance to the Course Style Guide:	**20 points** (All or Nothing)
Correctness:	**80 points** (Partial Credit Possible)
`containsIgnoringCase()`:	60 points
`longerFor()`:	20 points

Manual Grading

After the due date, the Professor may manually review your code. At this time, points may be deducted for inelegant code, inappropriate variable names, bad comments, etc. In particular, **for this assignment, you should take care to take advantage of the capabilities of `String` objects and not to write unnecessary code yourself**.