

APCS Problem Set 10: **Strings** and File I/O**3.3 Summary of String Methods**

Refer to the instructions in the online form of this section. (21pts)

String method	Description — what is returned and what the parameters mean
<code>length()</code>	For a given string, this method returns...
<code>charAt(pos)</code>	For a given string, this method returns... ...at position “pos”
<code>substring(fromPos)</code>	
<code>substring(fromPos, toPos)</code>	
<code>concat(str)</code>	
<code>compareTo(s2)</code>	
<code>compareToIgnoreCase(s2)</code>	
<code>equals(s2)</code>	

String method	Description — what is returned and what the parameters mean
<code>equalsIgnoreCase(s2)</code>	
<code>indexOf(ch)</code>	
<code>indexOf(ch, fromPos)</code>	
<code>indexOf(str)</code>	
<code>indexOf(str, fromPos)</code>	
<code>lastIndexOf(ch)</code>	
<code>lastIndexOf(ch, fromPos)</code>	
<code>lastIndexOf(str)</code>	
<code>lastIndexOf(str, fromPos)</code>	
<code>trim()</code>	

String method	Description — what is returned and what the parameters mean
<code>replace(oldChar, newChar)</code>	
<code>toUpperCase()</code>	
<code>toLowerCase()</code>	

8.3 Implementing LipogramAnalyzer

Create the `LipogramAnalyzer` class per the description on Litvin pp225-226.

Teacher's Initials: _____ (30pts)

11.2 File I/O Exercise #1: FileRewinder

In a new project, called `PS10-FileRewinder`, create class `FileRewinder`. Have it perform the following actions.

1. Open up a text file in your program's working directory. (*Note: See online version of the problem set for the footnote text.*)
2. Read the file, one line at a time, saving each line as a `String` object in an `ArrayList<String>`.
3. Write, to a different file in the working directory, the `Strings` in your `ArrayList<String>` in reverse order. (*Again, see online version for footnote.*)

You may create your own input text file or download a prepared one from <http://feromax.com/apcs/problemsets/PS10/downloads/ascii.txt>.

Teacher's Initials: _____ (20pts)

11.3 File I/O Exercise #2: Animals

4. Once you're confident that you can read configuration values from the config file, make your `Animals` class output the following to the screen:

```
<name> is a <height>-foot-tall <animal> that stands on <numfeet> feet.
```

For example, the provided configuration file text should cause `Animals` to print

```
Michael is a 2.5-foot-tall penguin that stands on 2 feet.
```

Teacher's Initials: _____ (20pts)