LetterCount.java

Objective: Become familiar with some basic programming conventions and file I/O.

Background:
Counting the frequency of a letter in text is useful. Early mathematicians used it to break substitution ciphers. In games like hangman or "Wheel of Fortune" guessing the most frequently used letters first can help find the solution quickly. Linguists use letter frequency to describe language and designers use it to determine the layout of keyboards.

You are going to build a program that counts the number of times a letter is used in a text file. Both uppercase and lowercase letters will be counted and combined into one score for each letter of the alphabet. Non-letters like punctuation, digits, and white space (blanks and tabs) will be ignored. The results will be reported in a letter frequency chart or histogram of each letter in the alphabet.

Assignment:

1. Download the file **LetterCount.zip** and unzip or expand it. It will create the directory "**LetterCount**" and you will do all of your work in that directory. Inside the directory you will find four files: **birmingham.txt**, **magi.txt**, **panama.txt**, and **LetterCount.java**. The text files can be used as test input. The **LetterCount.java** file is a skeleton Java file in which you will create your program. Be sure to copy in your **FileUtils** class for the project.

2. Create an **int** array to store the count for each letter. This is declared as a field and should be made **private**. All fields in this course will be private. The array should be sized and initialized in the constructor (not when it is declared).

3. Open the input file using a **FileUtils** static method.

4. Create a method to read in the file and count each letter case-insensitive. For example, "Annabelle" has two "a"s in it. Close the file when you are done counting.

5. Print out a header declaring "Histogram of letter frequency in" and mention the input file name.

6. Print a histogram of frequencies in alphabetic order (see below). It should include the absolute count of the letter and a relative frequency bar with a maximum length of 60 characters. Do not use long if-else chains or switch statements to implement this portion. You will lose points if you do.

7. Provide comment blocks at the beginning of each method you create. Comment your fields and let us know what they contain and how they are used.

8. Include your name as "@author" and the date you created the program as "@since". You will lose points if you do not include these.

A sample run:

```
% java LetterCount birmingham.txt

Histogram of letter frequency in birmingham.txt

a:   2442 ++++++++++++++++++++++++++++++++++
b:    484 +++++++
c:   1027 ++++++++++++++
d:   1181 ++++++++++++++++
e:   4072 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
f:    716 ++++++++++
g:    705 ++++++++++
h:   1717 ++++++++++++++++++++++++
i:   2530 ++++++++++++++++++++++++++++++++++++
j:    107 +
k:    128 +
```

```
l:   1227 ++++++++++++++++++
m:    861 ++++++++++++
n:   2313 +++++++++++++++++++++++++++++++++
o:   2541 ++++++++++++++++++++++++++++++++++++
p:    577 ++++++++
q:     25
r:   1929 ++++++++++++++++++++++++++++
s:   2092 ++++++++++++++++++++++++++++++
t:   3113 +++++++++++++++++++++++++++++++++++++++++++++
u:   1025 ++++++++++++++
v:    382 +++++
w:    633 +++++++++
x:     58
y:    587 ++++++++
z:     25
```