CPSC 1060: Introduction to Programming in Java

Project 4: Letter Frequencies

Learning Objectives

- Write a multi-module program in Java that utilizes static methods, variables (local and static), assignments, expressions, inputs, outputs, decision statements, loops, nested loops, various nested statements, and arrays of objects.
- Design and implement an algorithm to solve a problem.
- Apply knowledge of creating, compiling, running, testing, and debugging a Java program in a Unix environment.
- Use basic Unix commands.
- Use good programming style and standards.

Introduction

For this project, you are going to complete a multi-module Java program that uses many programming concepts we covered this semester (methods, loops, strings, arrays of strings, static variables/methods, etc.). Your assignment is to complete the two methods in the provided file **LetterFrequencies.java** such that when that module is compiled with the provided code in **LetterFrequenciesDriver.java**, the program outputs the same letter counts as shown in the Examples section of this document when run on our Unix machines. Before you start this project, read through the provided files and the examples at end of the document. Once you get an idea of what the program should do, then implement the methods in **LetterFrequencies.java**. After you finish implementing all the methods in **LetterFrequencies.java**, your program should compile and run (without any warnings or errors) using the following command on our Unix machines.

```
                     javac LetterFrequenciesDriver.java
 java LetterFrequenciesDriver ‘Command’ ‘line arguments’ ‘are fun!!!’
```

Note: students should type the commands above instead of copying/pasting them on a Unix command line. Copying a command from a document like this to a Unix command line may result in errors.

Program Requirements

1. Download all the provided files and place them into the same folder on one of our Unix machines.
2. Read through the provided files and our examples to understand how the program should work.
3. Complete the two methods in **LetterFrequencies.java** in such a manner that your complete multi-module program produces the same output as shown in the Examples section.
4. Once you get your program working correctly with our examples in **LetterFrequenciesDriver.java**, then add your own examples to

LetterFrequenciesDriver.java to further test that your program works for any valid set of input.

Additional Requirements

1. Your program must include a comment as shown below between the import statements and the class declaration. In your source code, type the comments below, agree to the comments below, and fill in the file name, your name, the submission date, and the program's purpose.

   ```
   /*
   * [Class name here].java
   * Author:  [Your name here]
   * Submission Date:  [Submission date here]
   *
   * Purpose: A brief paragraph description of the
   * program. What does it do?
   *
   * Statement of Academic Honesty:
   *
   * The following code represents my own work. I have neither
   * received nor given inappropriate assistance. I have not copied
   * or modified code from any source other than the course webpage
   * or the course textbook. I recognize that any unauthorized
   * assistance or plagiarism will be handled in accordance
   * with the policies at Clemson University and the
   * policies of this course. I recognize that my work is based
   * on an assignment created by the School of Computing
   * at Clemson University. Any publishing or posting
   * of source code for this project is strictly prohibited
   * unless you have written consent from the instructor.
   */
   ```

2. All instructions stated in this document must be followed for full credit to be awarded.

Project Grading

All projects are graded out of a possible 100 points. Programs not submitted on time will receive a grade of zero. Programs that do not compile on our Unix machines will receive a grade of zero. You must make absolutely certain your program compiles before submitting, and you must thoroughly test your program with many different valid inputs to verify that it is working correctly.

This project will be graded for both correctness and style:

Style [20pts]

- 5 points for including the class comment required for all projects in all of your submitted **.java** files

- 5 points for using well-named variables, well-named methods, and briefly commenting all methods and classes
- 5 points for proper and consistent code indentation and readability
- 5 points for your source code compiling without any warnings on our Unix machines. Correct any compiler warnings before submitting your assignment. Note: programs that we cannot compile on our Unix machines due to one or more compiler errors will result in a grade of zero on this assignment.

Correctness [80pts]

- 80 points for correct output on various test cases. You can assume that we will test your program by using a **LetterFrequenciesDriver.java** file like the provided **LetterFrequenciesDriver.java**. However, the **LetterFrequenciesDriver.java** file that we use for grading may contain different test cases than the one provided for this assignment. We will grade your submission for this assignment by compiling it with our grading version of **LetterFrequenciesDriver.java**.

Project Submission

After you have tested, tested, and retested that your code compiles and run correctly on many test cases (those provided in this document and test cases that you create on your own), submit the file **LetterFrequencies.java** (and only that file) via our Canvas labs website under this project's assignment (if you have any questions about how to submit your project, then you'll need ask your lab instructor days before the assignment is due). After you submit file(s) on Canvas for this program and other programs that you write this semester, always double check that you submitted the correct file(s) (to do this, download your submission from Canvas, view it, and make sure the submitted file(s) compiles and runs correctly on our Unix machines).

Examples

When you complete the required methods correctly, your program should output what is shown below when you run the command immediately below on our Unix machines.

```
java LetterFrequenciesDriver ‘Command’ ‘line arguments’ ‘are fun!!!’
```

This output assumes that you've compiled your completed program with an unmodified version of our provide file **LetterFrequenciesDriver.java** on our Unix machines.

```
===Test 1===
a count = 2
b count = 0
c count = 0
d count = 0
e count = 1
f count = 0
g count = 0
```

```
h count = 0
i count = 1
j count = 0
k count = 1
l count = 3
m count = 0
n count = 2
o count = 0
p count = 0
q count = 0
r count = 2
s count = 2
t count = 3
u count = 1
v count = 0
w count = 0
x count = 0
y count = 1
z count = 0

===Test 2===
a count = 21
b count = 5
c count = 9
d count = 16
e count = 21
f count = 8
g count = 7
h count = 7
i count = 18
j count = 2
k count = 1
l count = 11
m count = 3
n count = 12
o count = 32
p count = 8
q count = 0
r count = 18
s count = 13
```

```
t count = 14
u count = 5
v count = 1
w count = 1
x count = 1
y count = 9
z count = 1

===Test 3===
a count = 8
b count = 4
c count = 0
d count = 3
e count = 7
f count = 0
g count = 2
h count = 3
i count = 7
j count = 1
k count = 3
l count = 4
m count = 5
n count = 7
o count = 10
p count = 3
q count = 0
r count = 8
s count = 4
t count = 8
u count = 3
v count = 0
w count = 1
x count = 0
y count = 1
z count = 0

===Test 4===
a count = 3
b count = 0
c count = 1
```

```
d count = 1
e count = 3
f count = 1
g count = 1
h count = 0
i count = 1
j count = 0
k count = 0
l count = 1
m count = 3
n count = 4
o count = 1
p count = 0
q count = 0
r count = 2
s count = 1
t count = 1
u count = 2
v count = 0
w count = 0
x count = 0
y count = 0
z count = 0

===Test 5===
a count = 0
b count = 0
c count = 0
d count = 0
e count = 0
f count = 0
g count = 0
h count = 0
i count = 0
j count = 0
k count = 0
l count = 0
m count = 0
n count = 0
o count = 0
```

```
p count = 0
q count = 0
r count = 0
s count = 0
t count = 0
u count = 0
v count = 0
w count = 0
x count = 0
y count = 0
z count = 0

===Test 6===
a count = 0
b count = 0
c count = 0
d count = 0
e count = 0
f count = 0
g count = 0
h count = 0
i count = 0
j count = 0
k count = 0
l count = 0
m count = 0
n count = 0
o count = 0
p count = 0
q count = 0
r count = 0
s count = 0
t count = 0
u count = 0
v count = 0
w count = 0
x count = 0
y count = 0
z count = 0
```