Section Handout #5: Files, Arrays, and ArrayLists

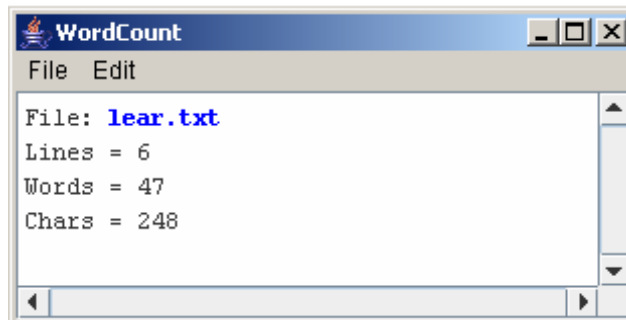## 1. Word count

Write a program **WordCount** that reads a file and reports how many lines, words, and characters appear in it. Suppose, for example, that the file **lear.txt** contains the following passage from Shakespeare's *King Lear*:

```
Poor naked wretches, wheresoe'er you are,
That bide the pelting of this pitiless storm,
How shall your houseless heads and unfed sides,
Your loop'd and window'd raggedness, defend you
From seasons such as these?  O, I have ta'en
Too little care of this!
```

Given this file, your program should be able to generate the following sample run:
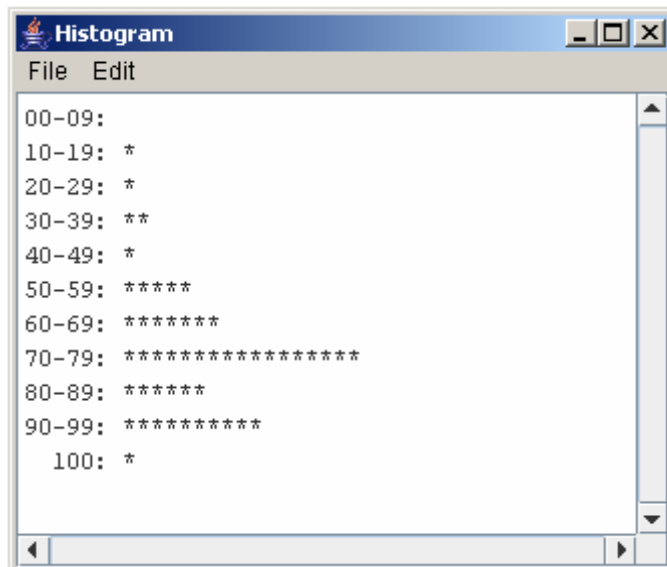

For the purposes of this program, a word consists of a consecutive sequence of letters and/or digits, which you can test using the static method **Character.isLetterOrDigit**. Also, you should not count the characters that mark the end of a line, which will have different values depending on the type of computer.

2. Histograms

Write a program that reads a list of exam scores from the file **MidtermScores.txt** (which contains one score per line) and then displays a histogram of those numbers, divided into the ranges 0–9, 10–19, 20–29, and so forth, up to the range containing only the value 100. If, for example, **MidtermScores.txt** contains the data shown in the right margin, your program should then be able to generate a histogram that looks as much as possible like the following sample run:
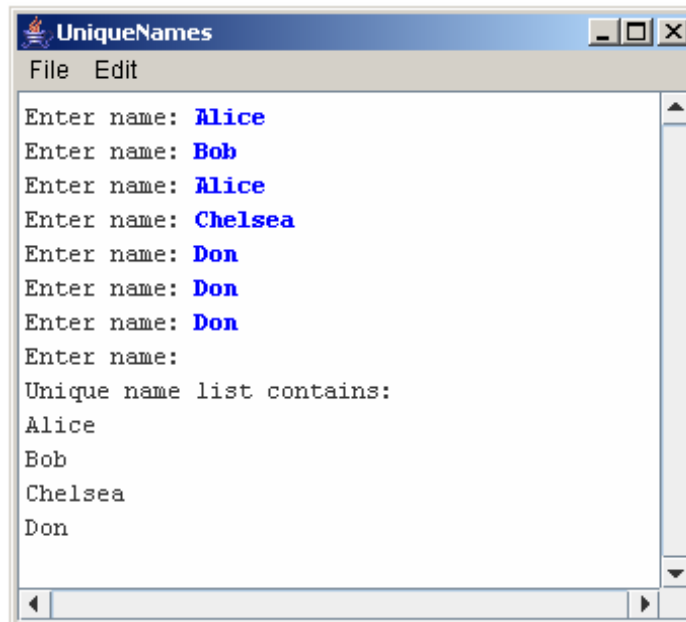
```
Histogram                         _ □ X
File  Edit

00-09:
10-19: *
20-29: *
30-39: **
40-49: *
50-59: *****
60-69: *******
70-79: ******************
80-89: ******
90-99: **********
  100: *
```

MidtermScores.txt

```
73
58
73
93
82
62
80
53
93
52
92
75
65
95
23
100
75
38
80
77
92
60
98
95
62
87
97
73
78
72
55
58
42
31
78
70
78
74
70
60
72
75
84
87
62
17
92
78
74
65
90
```

3. Unique Names

Write a program that asks the user for a list of names (one per line) until the user enters a blank line (i.e., just hits return when asked for a name). At that point the program should print out the list of names entered, where each name is listed only once (i.e., uniquely) no matter how many times the user entered the name in the program. You may find that using an **ArrayList** to keep track of the names entered by user may greatly simplify this problem.

A sample run of this program is shown below.

```
UniqueNames
File   Edit

Enter name: Alice
Enter name: Bob
Enter name: Alice
Enter name: Chelsea
Enter name: Don
Enter name: Don
Enter name: Don
Enter name:
Unique name list contains:
Alice
Bob
Chelsea
Don
```