

Section Handout 5

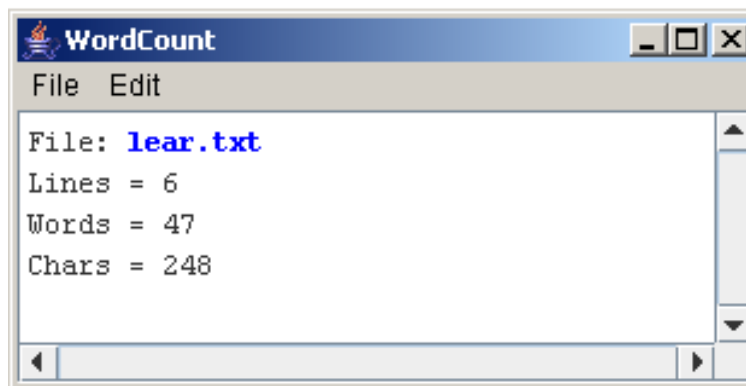
Based on a handout by Eric Roberts, Jeremy Keeshin, and Mehran Sahami

Problem One: 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:

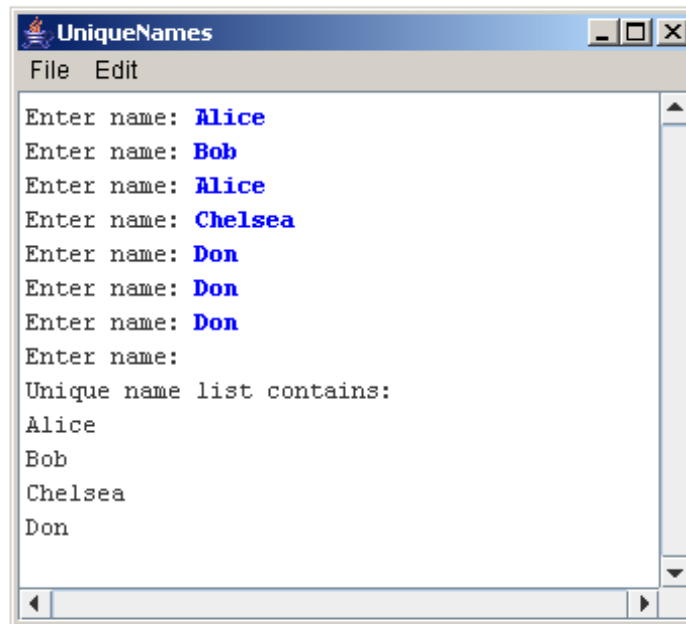


For the purposes of this program, a word consists of a consecutive sequence of letters and/or digits, which you can test using the static method `Character.isLetterOrDigit`. Also, you should not count the characters that mark the end of a line, which will have different values depending on the type of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

Problem Two: How Unique!

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.



Problem Three: A Christmas Carol

Using the `student` class from Chapter 6 as an example, write a class definition for a class called `Employee`, which keeps track of the following information:

1. The name of the employee
2. The employee's nine-digit tax id number
3. The employee's job title
4. A flag indicating whether the employee is still active
5. The employee's annual salary

The first two fields should be set as part of the constructor, and it should not be possible for the client to change these values after that. For the other fields, your class definition should provide getters and setters that manipulate those fields. Once you have made this definition, write the code necessary to initialize the following `Employee` objects:

ceo
Ebenezer Scrooge
161803399
CEO
<i>active</i>
£1000

partner
Jacob Marley
271828182
Former Partner
<i>inactive</i>
£0

clerk
Bob Cratchit
314159265
Clerk
<i>active</i>
£25

What would you need to do to double Bob Cratchit's salary?

Problem Four: Be My Valentine

For the program below, trace through its execution by hand to show what output is produced when it runs.

```
/* Valentine.java */
import acm.program.*;

public class Valentine extends ConsoleProgram {
    public void run() {
        Flower rose = new Flower("Rose", 137);
        Flower roseByAnotherName = rose;
        Flower tulip = new Flower("Tulip", 42);

        println(rose.getPetals());
        println(roseByAnotherName.getPetals());
        println(tulip.getPetals());

        int heart = 2;
        if (heart <3) { // D'awwww...
            roseByAnotherName.setPetals(3);
        }

        roseByAnotherName.setPetals(4);
        tulip.setPetals(5);

        println(rose.getPetals());
        println(roseByAnotherName.getPetals());
        println(tulip.getPetals());
    }
}

/* Flower.java */
public class Flower {
    private String name;
    private int numPetals;

    public Flower(String flowerName, int numPetals) {
        name = flowerName;
        numPetals = numPetals;
    }

    public int getPetals() {
        return numPetals;
    }

    public void setPetals(int petals) {
        numPetals = petals;
    }

    public String getName() {
        return name;
    }
}
```