**Counting Characters**

The file contains the skeleton of a program to read in a string (a sentence or phrase) and count the number of blank spaces in the string. The program currently has the declarations and initializations and prints the results. All it needs is a loop to go through the string character by character and count (update the *countBlank* variable) the characters that are the blank space. Since we know how many characters there are (the *length* of the string) we use a count controlled loop -- *for* loops are especially well-suited for this.

1. Add the *for* loop to the program. Inside the for loop you need to access each individual character -- the *charAt* method of the String class lets you do that. The assignment statement
2. ch = phrase.charAt(i);

assigns the variable *ch* (type char) the character that is in index i of the String *phrase*. In your for loop you can use an assignment similar to this (replace i with your loop control variable if you use something other than i). NOTE: You could also directly use phrase.charAt(i) in your if (without assigning it to a variable).

1. Test your program on several phrases to make sure it is correct.
2. Now modify the program so that it will count several different characters, not just blank spaces. To keep things relatively simple we'll count the a's, e's, s's, and t's (both upper and lower case) in the string. You need to declare and initialize four additional counting variables (e.g. *countA* and so on). Your current *if* could be modified to cascade but another solution is to use a *switch* statement. Replace the current *if* with a switch that accounts for the 9 cases we want to count (upper and lower case a, e, s, t, and blank spaces). The cases will be based on the value of the *ch* variable. The switch starts as follows -- complete it.
3. switch (ch)
4. {
5. case 'a':
6. case 'A': countA++;
7. break;
8. case ....
9. }

Note that this switch uses the "fall through" feature of switch statements. If *ch* is an 'a' the first case matches and the switch continues execution until it encounters the *break* hence the countA variable would be incremented.

1. Add statements to print out all of the counts.
2. It would be nice to have the program let the user keep entering phrases rather than having to restart it every time. To do this we need another loop surrounding the current code. That is, the current loop will be nested inside the new loop. Add an outer while loop that will continue to execute as long as the user does NOT enter the phrase *quit*. Modify the prompt to tell the user to enter a phrase or *quit* to quit. Note that all of the initializations for the counts should be inside the while loop (that is we want the counts to start over for each new phrase entered by the user). All you need to do is add the while statement (and think about placement of your reads so the loop works correctly). Be sure to go through the program and properly indent after adding code -- with nested loops the inner loop should be indented.
3. // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
4. // Count.java
5. //
6. // This program reads in strings (phrases) and counts the
7. // number of blank characters and certain other letters
8. // in the phrase.
9. // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
10. public class Count
11. {
12. public static void main (String[] args)
13. {
14. String phrase; // a string of characters
15. int countBlank; // the number of blanks (spaces) in the phrase
16. int length; // the length of the phrase
17. char ch; // an individual character in the string
18. // Print a program header
19. System.out.println ();
20. System.out.println ("Character Counter");
21. System.out.println ();
22. // Read in a string and find its length
23. System.out.print ("Enter a sentence or phrase: ");
24. // read input Scanner
25. // Find the length of the phrase
26. // Initialize counts
27. countBlank = 0;
28. // a for loop to go through the string character by character
29. // and count the blank spaces
30. // Print the results
31. System.out.println ();
32. System.out.println ("Number of blank spaces: " + countBlank);
33. System.out.println ();
34. }
35. }