Comp 401 - Assignment 2: Scanning for Numbers, Words, and Quoted Strings

Date Assigned: Aug 31, 2011
Completion Date: Fri Sep 9, 2010
Early Submission Date: Wed Sep 7, 2010

In this assignment, you will get more practice scanning. This work will build on your previous assignment.

Task
Extend your Java program of the previous assignment to print a greater variety of tokens. So far you were recognizing and printing tokens that were digit sequences. You should also recognize and print the following two tokens:

- **Commands**: a sequences of uppercase and lower case letters.
- **Quoted string**: a sequence of characters enclosed in double quotes

You can assume that the first character of the scanned string is a letter or a digit, and that each token (number/command/quoted string) is followed by exactly one blank. You can assume that an opening double quote will always be matched by a closing double quote. As before, you can halt the program on encountering something unexpected such as an illegal character or the end of the string without a matching double quote. However, as before, you must print all tokens before the first error is encountered, and if you are doing extra credit, continue scanning, if possible, after the first error.

This time, you do not have to do number computations such as finding the sum or product, or store the tokens in an array. Also, this time, before you print a token, give an indication of its type (command, number, or quoted string). Thus, if the String argument to main is:

“move 30 say "you say hello" move 100 say "I say goodbye " $%!@""

You should print:

command: move

number: 30

command: say
quoted string: you say hello
command: move
number: 100
command: say
quoted string: I say goodbye
quoted string: $%!@

As you see here, the double quote character in a main argument is preceded (escaped) by the back slash character `\`, because without the backslash, Java will assume that it is ending the quote used to start the argument. The back slash does not actually get inserted into the string, so the program does not see it. In a program, you can enclose a double quote simply in single quote characters as in ‘”’.

As you also see in the example above, all characters within quotes are to be printed – you should not, for instance, remove spaces.

Copying the example string directly into the Eclipse window can cause problems, so you will have to rewrite it.

**Constraints**

1. You can use all of the Java features allowed in the previous assignment and those taught in class so far. In addition you can use the Character.isLetter() method, which indicates if its character argument is a letter.

2. As before, have at least two methods.

3. It will be best to create a new project for each assignment, which starts off as a copy of the last assignment, and is then modified to add new behavior.

**Extra Credit**

1. Allow an arbitrary number of spaces before and after each token.

2. Give an error message if the closing double quote is missing.

3. Give error messages in case the user does not enter a valid token, that is, the user enters an unexpected character, and keep scanning.

4. Look also for and print (a) the plus token, consisting of the single character, `+' and (b) the minus token, consisting of the single character, `-' Do not make this
*token a part of the number token*. Thus, if the argument is “move – 100” you should print three different tokens and not two:

Command: move

sign: -

number: 100

5. Finish the assignment by the early completion date.

**Submission Instructions**

- These are the same as in the previous assignment except you do not have to show your knowledge of debugging this time. The importance of screenshots will be even greater this time as there are more features to demonstrate. If you don’t show us screenshots to demonstrate a feature, we will assume it does not work.

Good luck!