Module 1 Spring 2012 CS210
Instructions:

1. There are about 18-20 commands in the Project Grammar on eCampus. The grammar is a reference that you will have to adhere to with great specificity in order to pass testing.
2. The Project 1 requirements are straightforward. When the application runs, the user should get a command line interface. The user should type in a command, and, based on the rules in the Project Grammar, the program should respond with either:
   a) "This is a syntactically correct select statement" (replace 'select' with a one or two word identifier of the statement type) OR
   b) "This is not a correct statement" (The complete implementation of each of these commands will occur in various Projects throughout the semester).
3. Couple of IMPORTANT notes –
   a) Your program should be case insensitive just like standard SQL. The exception is that variables, including **Varchar**, **Dates**, and **Chars** (ALL of which will be entered enclosed in ***SINGLE*** quotes), need to be case sensitive.
   b) All commands are terminated by a semicolon. It is possible that a command will extend over multiple lines, and there may even be one or more blank lines in a command. Your program will need to handle this.
4. Two of the commands in the grammar will need to have their functionality implemented during this project:
   a) "Exit" – 'exit' terminates the program. Obviously we will need this from the start so that we can finish an execution.
   b) "Read" – this command will open and read a file (address of the file is a component of the command). The file should contain any number of commands consistent with the grammar. Your program will need to sequentially process each of these commands just as it would with command line input.
5. Note that the finished product should be polished and "bullet proof". The product should handle any input gracefully, and either process the input or provide a user friendly response if the input is not appropriate for the application. There should never be java error messages output to the user.
6. For Module 1, you will need to create complete javadocs for any classes you submit.
7. You will also need to create a complete set of JUNIT tests for the various regular expressions you create.
8. See Submission Instructions on eCampus or talk to me if you have questions.

HINT: Regular Expressions can be used to parse the input commands. Regular expressions are in reality a *terrible* way to parse a programming language such as the one you are constructing. OTOH, regular expressions are an indispensable tool for programmers and that's why we're using them. Don't try to be too cute with the regular expressions. For example, if the input statement is

"ID_KEYWORD long_string_of_data_to_be_parsed ANOTHER_ID_KEYWORD"

It might be quite challenging to use regex's to parse the string of data, and you might be better off just looking for the two key words so you know what kind of statement you are dealing with, and save the long string of data as a variable so you can parse it later in other ways.