Lab Exercise for Week 2

Reading API documentation is very important to any programming work. This exercise assumes you understand the code explained in lecture and know how to look for answers in ArcSDE Java Doc (could be found at:
http://courses.washington.edu/geog465/arc_doc/Support_files/sdehelp.htm >>
Development environments >> Java API >> Function reference >> Java API javadoc).

Answer the following questions by coding and testing in JBuilder. Submit your printed code as the answers. ***Please bring your answers to Monday's lecture (Jan. 17th).*** We shall review these answers in section.

1. One of the disadvantages of using the class `java.util.Vector` to store the list of layer objects is that you can only get a layer object by its index number, which is not intuitive. Write an accessor method for class "`SDEAccess`" to return a layer object from a `Vector` of layers, given the name of the layer. For example, suppose the method's name is "`getLayerByName`", we want to use the method this way:

   ```
   int n = this.getLayerByName("BainbridgeIsland");
   ```

Hints: 1) To get an element from a `Vector`, use method `elementAt`;
   2) The `String` class has a method `compareToIgnoreCase` which can be used to compare two strings.

2. Modify the code of class "`SDEAccess`" to access another layer in the geodatabase "classdata" located in collab server and print out all cities names of (and only of) the state of Washington.

Hints: 1) Class "**SeSqlConstruct**" has 4 constructor methods, one of those has a parameter to let you specify a query filter to extract only part of the features that meet a given criteria:

   SeSqlConstruct(java.lang.String table, java.lang.String where)

   In this method, the "`where`" parameter is a Boolean expression which specifies the criteria, e.g., "`males>56000`", "`name='seattle'`". Pay attention to how a string value is given. You might need to use ArcCatalog to explore the attributes of the layer.

   2) The "`fetch`" method of class "`SeQuery`" allows you iterate through all rows in the query result, until it becomes `null`. You can use a `while` loop to get the features one by one with the "`fetch`" method.