Lab 14
Array of Objects and File IO

The following exercises are to be completed during lab class. If you do not have time to finish during lab, they must be completed before the beginning of the following lab session.

Set-Up

- Create a new project in your Eclipse workspace named: **Lab14**
- In the *src* folder, create a package named: **edu.ilstu**
- Import the following files into the package in your *src* folder.
  - *CDInput.java*
  - *CDOutput.java*
  - *CDDriver.java*
  - *Collection.txt*
- Import the input file, *Collection.txt*, into the root of your project.

.

Problem

You will be writing a program to manage a list for a CD collection. Existing CDs are stored in a file. The data in the file (title and artist) is read into an array. The user will be presented with a menu with the following choices: add a new CD, display the list, or quit. You will create one new class and modifying existing classes.

Be sure to write good comments as you will be generating JavaDoc API documents at the end of the lab.

Class diagrams and main algorithm

Song
- title:String - artist:String
+ Song(String title, String artist) + toString( ):String all getters and setters

CDInput
- file:File - keyboardInput:Scanner - fileInput:Scanner
+ readMenuChoice():int + readSong():Song + readMusicCollection(Song[] songArray, String filename):int

CDOutput
- outputFile:PrintWriter
+ printMenu():void + printCollection(Song[] songArray, int count):void + openFile(String fileName):void + writeToFile(Song song):void + closeFile():void

```
main algorithm
      CREATE object for CDOutput
      CREATE object for CDInput

      CREATE array of Song objects
      INITIALIZE the array to hold empty objects

      READ music collection

      PRINT menu

      choice = READ menu choice

      WHILE (choice != 3)
            CASE choice
                  WHEN 1:
                        READ data for new song from keyboard
                        ADD the new song object to the array
                        INCREMENT count
                        WRITE the new song data to the file
                  WHEN 2:
                        PRINT the array
                  OTHERWISE:
                        PRINT "Invalid menu choice.  Please try again"
            END CASE

            PRINT menu
            choice = READ menu choice
      END WHILE

      CLOSE output file
END MAIN
```

Processing Instructions (write the code in this order) – use the comments in the program to help with placement.

1. Create the new class called Song.java using the given class diagram.
   - There is no default constructor, only the one sending in both title and artist.
   - toString - define it so that it prints in the form: *title* by *artist*

2. **CDOutput.java**
   - Write the printCollection method - print the list to the screen.
     - Doing this at the beginning allows you to test that the array has the correct values whenever a change has been made to it.

3. **CDDriver.java**
   - You have been given the declarations for:
     - MAX_ARRAY_SIZE
     - FILENAME
     - count (used to keep track of the number of elements currently in the array)
     - choice (used to capture the menu choice made by the user)
     - objects for CDInput and CDOutput
   - Create the array and fill it with empty objects.
     - Note that the only constructor available must receive data for the title and artist. Pass in empty strings for now.
   - Write the logic to process the menu. (Start where the comment says "Print the menu") Write the call statement for the printMenu method in the CDOutput class. NOTE: The method is called twice – one before the loop and another one at the end of the loop.
   - Add the call to the method in CDOutput to print the array collection of songs (choice 2)
   - Write the call statement for the readMenuChoice method in CDInput. There are also two of these statements. One before the loop and another at the bottom of the loop.

4. **CDInput.java**
   - Note that there are two scanner objects. One object will be used to read data from the keyboard. (This one has already been created for you. Notice the System.in parameter.) The other one will be used to read from the file.
   - Write the readMenuChoice method
     - Run your program to test printing and reading the menu choice to be sure this is working correctly.
   - Write the readMusicCollection method. One advantage to arrays is that you can open and close the file in the same method where you read. Remember to

use a try/catch when attempting to open the file. You will also be keeping track of how many Song objects are put into the array.

5. **CDDriver.java**
   - Write the call statement for the readMusicCollection method in CDInput.
   - Test using menu choice 2 to see if everything loaded correctly.

6. **CDOutput.java**
   - Write the code for the following methods:
     - openFile
       - NOTE: The same file can be used for reading and writing, but at different times. Notice that the file was closed after reading the data into the array and now you are opening it to be used for writing new songs to add (append) to the collection.
       - Use a try/catch for any exceptions.
     - writeToFile
       - Write the title, then the artist on separate lines.
     - closeFile

7. **CDDriver.java**
   - Write the call statement to the openFile method in CDOutput.

8. **CDInput.java**
   - Write the readSong method. (The method was commented out so that the errors wouldn't keep you from being able to test the program as you added code earlier. Highlight the commented out lines and hit Ctrl-/ to uncomment all of them at one time.) Ask the user to read the data from the keyboard and create a new Song object.

9. **CDDriver.java**
   - Add the code for Case 1
     - Call the readSong method written above
     - Add the song to the array
       - Don't forget to add to the count so you know how many items are in the array
     - Add the song to the file
   - Call the method to close the output file
   - Run the code. Add a song, then use menu choice 2 to see if it was added correctly.
     - If both prompts print at the same time, you may have to deal with the newline character between reading choice and the title.

To Be Submitted

The following files should be zipped together into a file called Lab14.zip and submitted to ReggieNet by the beginning of your next lab.

- CDDriver.java
- CDInput.java
- CDOutput.java
- Song.java