CMPT220 - Program 6

Program Due: Thursday, March 12th, before 9:00 a.m. (saved, submitted and printed)

Name the project **Prog6YourLastName**
Name the main class **MusicDemoYourLastName.java**
Name the Song class **SongYourLastName.java**
Name the Playlist class **PlaylistYourLastName.java**

Be sure to include Javadoc documentation throughout the entire project!

It's time to make a custom Playlist!

First, define a class for a **SongYourLastName**. Information to be tracked includes the following:

- myName – a String, representing the name of the song
- myArtist – a String, representing the name of the artist
- myMinutes – an integer, representing the number of minutes in the runtime
- mySeconds – an integer, representing the number of second in the runtime
- myPrice – a double, representing the download price of this song.

Be **sure** to define the constructors, getters and setters! You'll also need a toString() method.

Next, define a class for a **PlaylistYourLastName**, which is represented by:

- mySongs – an array of references for up to 10 Song objects, and
- mySize – the current number of Song objects in the playlist.

Define (only) the null constructor and getSize(). In addition, we'll need

addToPlaylist, which accepts a Song and adds it to the Playlist (if the playlist isn't already full). This method returns a Boolean indicating whether or not the Song was added. (It will return **true** if it was added; otherwise, it'll return **false**.)
findLongest, accepts no parameters and returns the longest Song in the playlist. You'll use a for-loop in this method (and in the next 3 methods).
findShortest, accepts no parameters and returns the shortest Song.
calcTotal, accepts no parameters and returns the total download cost of the entire Playlist
printFullPlaylist, accepts no parameters and returns nothing. This method prints out details of all Songs currently in the playlist. **This is the *only* method that prints anything!**
deleteLongest, accepts no parameters and deletes the longest song from the playlist. The easiest way to do this is to replace this Song with the last Song on the Playlist.

Finally, define a class for **MusicDemoYourLastName**. Include the following menu.

```
A - Add a Song to the Playlist
L - Find the Longest Song in the Playlist
S - Find the Shortest Song in the Playlist
N - Find the Number of Songs in the Playlist
T - Find the Total Cost of all Songs in the Playlist
P - Print out Details about all Songs in the Playlist
D - Delete the Longest Song from the Playlist
Q - Quit
```

For **Option A**, be sure to prompt the user for the necessary input, store that in a **new** Song, then invoke the correct method. Let the user know whether or not the Song was added, according to the Boolean value that is returned.
Options L&S will print out details about the Song that is returned from the proper method.
Options N&T will simply print out a nicely formatted message, using the numeric value that is returned from the proper method.
Option P will simply call the appropriate method.
Option D will actually remove the longest Song from the list.

Be sure to give descriptive feedback to the user for each menu choice. Allow upper/lower case.