

## Lab Assignment-7

### (I/O Streams, Serialization & JUnit Testing)

IIT-Delhi. 28<sup>th</sup> September. Due by 23:59pm on 29<sup>th</sup> September 2017

Instructor: Vivek Kumar

No extensions will be provided. Any submission after the deadline will not be evaluated. If you see ambiguity or inconsistency in a question, please seek a clarification from the teaching staff.

**Plagiarism:** All submitted homeworks are expected to be the result of your individual effort. You should never misrepresent someone else's work as your own. In case any plagiarism case is detected, it will be dealt as per IITD policy for plagiarism.

**NOTE:** You will have to create a "PRIVATE" git repository to manage your code for this lab assignment. This is not going to be a part of evaluation criteria in this lab but you should continue doing this as you have to use git extensively in your upcoming project.

---

#### Problem Description:

(I)

You are developing a Music App and you are in a stage of designing interactive music playlists. NOTE that no GUI programming is required in this lab assignment.

Your music app must upon invocation perform:

1. Enter the name of the music playlist which must exist in the working directory. Read in the music playlist from its binary file.
2. Display the number of songs stored within the playlist.
3. Provide the user a menu with options to
  - a. **Add** a new song in playlist. Display the number of songs in playlist after adding song. Assume that name of songs in playlist should be unique.
  - b. **Delete** a song from playlist by song's name. Display an error message if name of song does not exist else display the number of songs in playlist after deleting song.
  - c. **Search** for a song in playlist by song's name. Display an error message if name of song does not exist else display complete information of the song.
  - d. **Show** all songs in the playlist with their complete information. If no song exist, display "No Song Exist".
  - e. **Back to menu** option where all playlists in the current directory is shown.(Step 1)
  - f. **Exit** the app option.
4. Serialize the entire updated playlist to its file in the current directory.

Information of a song is represented by song name, singer, song's duration (in seconds).

Do implement Serializable in whatever class you write to implement an individual song entry in the playlist.

**(II)**

Create a sample database for your program that has a playlist with at least three songs. Write a JUnit test suite that contains following three tests:

1. First adds a song then validates the playlist by selecting the show songs option.
2. Then, deletes a song and then test the playlist again with the output of the show songs option.
3. Search for a song and validates its output.

All the three test cases in this test suite should pass successfully. Include your sample database in your submission.

**(III)** You will have to create two packages for this lab. One package will contain your actual program and the other will contain the JUnit test cases.