

# Download Lab

Create a new project named `download_lab`. Copy `SpotifyDownloads.java` and `DownloadInfo.java` into your project. You will implement code in empty routines

Make sure you use correct java conventions.

1.

(a). Write the `SpotifyDownloads` method `getDownloadInfo`, which returns a reference to a `DownloadInfo` object if an object with a title that matches that parameter title exists in the `downloadList`. If no song in `downloadList` has a title that matches the parameter title, the method returns `null`.

For example, suppose variable `webMusicA` refers to an instance of `SpotifyDownloads` and that the table below represents the contents of `downloadList`. The list contains three `DownloadInfo` objects. The object at position 0 has a title of "Human Qualities" and a download count of 5. The object at position 1 has a title of "Postcard from 1952" and has a download count of 3. The object at position 2 has a title of "Your Hand In Mine" and a download count of 10.

| 0                      | 1                         | 2                         |
|------------------------|---------------------------|---------------------------|
| "Human Qualities"<br>5 | "Postcard from 1952"<br>3 | "Your Hand In Mine"<br>10 |

The call `webMusicA.getDownloadInfo("Your Hand In Mine")` returns a reference to the object in position 2 of the list.

The call `webMusicA.getDownloadInfo("Happy Birthday")` returns `null` because there are no `DownloadInfo` objects with that title in the list.

**Implement the method `getDownloadInfo`.**

(b) Write the `SpotifyDownloads` method `updateDownloads`, which takes a list of song titles as a parameter. For each title in the list, the method updates `downloadList`, either by incrementing the download count if a `DownloadInfo` object with the same title exists or by adding a new `DownloadInfo` object with that title and a download count of 1 to the end of the list. When a new `DownloadInfo` object is added to the end of the list, the order of the already existing entries in the `downloadList` remains unchanged.

For example, suppose variable `webMusicB` refers to an instance of `SpotifyDownloads` and that the table below represents the contents of the instance variable `downloadList`.

| 0                      | 1                         | 2                         |
|------------------------|---------------------------|---------------------------|
| "Human Qualities"<br>5 | "Postcard from 1952"<br>3 | "Your Hand In Mine"<br>10 |

Assume that the variable `ArrayList<String> songTitles` has been defined and contains the following entries:

["Welcome Ghosts", "Your Hand In Mine", "Postcard from 1952", "So Long, Lonesome", "Welcome Ghosts", "Postcard from 1952"]

The call `webMusicB.updateDownloads(songTitles)` results in the following `downloadList` with incremented download counts for the objects with titles of "Postcard from 1952" and "Your Hand In Mine". It also has a new `DownloadInfo` object with a title of "Welcome Ghosts" and a download count of 2, and another `DownloadInfo` object with a title of "So Long, Lonesome" and a download count of 1. The order of the already existing entries remains unchanged.

| 0                      | 1                            | 2                            | 3                         | 4                           |
|------------------------|------------------------------|------------------------------|---------------------------|-----------------------------|
| "Human Qualities"<br>5 | "Postcard from<br>1952"<br>5 | "Your Hand In<br>Mine"<br>11 | "Welcome,<br>Ghosts"<br>2 | "So Long,<br>Lonesome"<br>1 |

In writing your solution, you must use the `getDownloadInfo` method. Assume that `getDownloadInfo` works as specified, regardless of what you wrote for part (a).

**Implement the method `updateDownloads()`.**