

OPTION D Revision

A music Web site keeps track of downloaded music. For each download, the site uses a DownloadInfo object to store a song's title and the number of times it has been downloaded.

```
public class DownloadInfo {

    private String title;
    private int numTimes;

    /* constructor method */
    public DownloadInfo(String title) {
        this.title = title;
    }

    public String getTitle() {
        return title;
    }

    public void incrementTimesDownloaded() {

        numTimes++;
    }
}
```

The list of downloaded information is stored in a MusicDownloads object. A partial declaration for the MusicDownloads class is shown below.

```
public class MusicDownloads {
    private DownloadInfo[] downloadList;
    private int index;

    /* constructor method */
    public MusicDownloads() {
        this.downloadList = new DownloadInfo[100];
        index = 0;
    }

    public DownloadInfo getDownloadInfo(String title) {
        /*to be implemented in Part A*/
    }

    public void updateDownloads(String[] titles) {
        /*to be implemented in Part B*/
    }
}
```

- (a) Write the `MusicDownloads` method `getDownloadInfo`, which returns a reference to a `DownloadInfo` object if an object with a title that matches the 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 `MusicDownloads` 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 "Hey Jude" and a download count of 5. The object at position 1 has a title of "Soul Sister" and a download count of 3. The object at position 2 has a title of "Aqualung" and a download count of 10.

0	1	2
"Hey Jude" 5	"Soul Sister" 3	"Aqualung" 10

The call `webMusicA.getDownloadInfo("Aqualung")` 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.

```
public DownloadInfo getDownloadInfo(String title){
```

```
}
```

(b) Write the `MusicDownloads` 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 at `index`. For example, a `MusicDownloads` object contains the following array of `DownloadInfo` objects:

"Hey Jude" 5	"Love is" 3	"Marry Me" 9	null	null
0	1	2	3	4
index				

Assume that the parameter `titles` contains the following entries.

`{"Hey Jude", "Marry Me", "Hey Jude", "Sister"}`

This will result in the following `downloadList` after a call to the `updateDownloads` method:

"Hey Jude" 7	"Love is" 3	"Marry Me" 10	"Sister" 1	null
0	1	2	3	4
index				

```
public void updateDownloads(String[] titles){
```

}

(c) Draw a UML diagram describing the relationship between the `MusicDownloads` and `DownloadInfo` classes.