

## The Music Player (part 1)

**Purpose:** To understand how to use arrays

**Learning Objectives:**

- Creating, initializing, searching arrays
- Utilizing collections as a component of an object
- Understanding the power of the toString() method
- Understanding static variables
- Utilizing constants effectively
- Searching a collection

**Artist:** Create a class Artist with instance variables for first name and last name that has the following:

- A two-argument constructor.
- A no-argument constructor.
- All getters and setters. The setter for last name should set the last name to “(last name unknown)” if the String parameter is of zero length, do nothing if this parameter is of length 1, and update the instance variable if the length is greater than 2.
- Now think about the two-argument constructor. Is there a way to perform the same validation on last name without repeating code?
- Create the toString() method to enable display of the artist’s full name, e.g., “Katy Perry”

**Song:** Create a class Song with instance variables for song title, genre, release year, and a list of up to three artists that has the following:

- A final, static variable for the max number of artists per song (i.e., 3)
- A final, static integer (ARTIST\_NOT\_ADDED) that acts as a special value if an Artist cannot be added to the array. This should be initialized to Integer.MIN\_VALUE;
- A three-argument constructor for title, genre and release year.
- All getters and setters.
- An addArtist() method which adds an Artist to the array of artists. This method should return the index position for the Artist that is being added. If the Artist cannot be added to the array, the method should return the integer value ARTIST\_NOT\_ADDED.
- Create the toString() method to enable display of the song and the year, e.g., “Roar (2013)”

**Playlist:** Create a class Playlist with instance variables for a playlist name and a list of songs that has the following:

- A final, static integer (SONG\_NOT\_ADDED) that acts as a special value if an Artist cannot be added to the array. This should be initialized to Integer.MIN\_VALUE;
- Another final, static integer (SONG\_NOT\_FOUND) that should also be initialized to Integer.MIN\_VALUE.
- A two-argument constructor.
- All getters and setters.
- A final, static variable for the max number of songs per playlist (i.e., 10)
- An addSong() method which adds an Song to the array of songs. This method should return the index position for the Song that is being added. If the Song cannot be added to the array, the method should return the integer value SONG\_NOT\_ADDED.

**Driver:** Create a driver class that will

- Make all the objects needed to complete the table below:

Song Title	Artist First Name	Artist Last Name	Genre	Release Year
Roar	Katy	Perry	Pop	2013
Lady Marmalade	Christina	Aguilera	Disco	2001
		P!nk		
	Mya			
Just Give Me A Reason	Lil'	Kim	Pop	2012
	Nate	Ruess		
Evolve	Mya		R&B	2012
Centuries		Fall Out Boy	Rock	2014
Crush On You	Mya		R&B	1997
		The Notorious B.I.G		
	Lil	Cease		
Crush On You		The Jets	Pop	1985

## OOPDA: Compare and Contrast Assignment

- Make a playlist named “complete” which has all of the above songs on it

In the Driver, create a method `displayPlaylist()` where you can pass in a Playlist, and the method will display the Songs and Artists as shown below. (Note Lady Marmalade will only have three artists, even if you try to add the fourth.) *Note: This is to give us more practice in creating methods in our Driver class. However, in the next version of this lab, we will move this method somewhere else...*

### `displayPlaylist(complete);`

1. Roar (2013)  
Katy Perry
2. Lady Marmalade (2001)  
Christine Aguilera  
P!nk  
Mya (last name unknown)
3. Just Give Me A Reason (2012)  
P!nk  
Nate Ruess
4. Evolve (2012)  
Mya (last name unknown)
5. Centuries (2014)  
Fall Out Boy
6. Crush On You (1997)  
Lil' Kim  
Lil Cease  
The Notorious B.I.G.
7. Crush On You (1985)  
The Jets

In the Playlist class, create a `search()` method where you can pass in the name of the Song, and the method will return the position on the playlist for the first song which matches the title parameter.

In the Driver class, prompt the user for a song title and use the new `search()` method to display where the song was found on the playlist and its primary artist (the first one added to the array).

```
Enter song title:  
lady marmalade  
lady marmalade is song 2  
The primary artist is Christine Aguilera
```

```
Enter song title:  
stairway to heaven  
stairway to heaven not found
```