# Final Project - Musify App

## Introduction

Welcome to the Final Project for COMP90041 - Programming and Software Development!

For this project, we will create a Musify App similar to YouTube or Spotify. In your real life, you may use some kind of database. But we will handle and save data using files.

There are three different media types that you will handle -

- Song
- Podcast
- ShortClip

You can create one or more playlists of these types. A playlist only contains one type of Media i.e. there can be different playlists like playlist1, playlist2, playlists.txt or any other name, etc. But playlist1 can only have songs, playlist2 can only have podcasts and so on.

A song has a Genre that can take one of these values - POP, ROCK, JAZZ. A podcast has a category that can take one of these values HEALTH, EDUCATION, TECHNOLOGY.

These are the following operations that you can do -

- Create one or more playlists
- Add or remove playlists.
- View all the playlists that you have created.
- A Playlist may contain media of the same type. They can be of type either SONG, PODCAST or SHORTCLIP (similar to YouTube Shorts or Instagram Reels)
- You can add content to a playlist
- You can remove media in a playlist.
- You can view the contents of a playlist
- You can play a playlist.

Additionally, you have to manage two tasks related to file handling -

- Read the data from the files at the beginning of the program.
- Write the data in the files while exiting the program.

# Preamble: "The Specifications"

The slides from this lesson module for the assignment act as "the specifications" for the system we are about to build.

**"The specifications"** in real-life software development is the document that details the features that should be implemented in any particular project. They are the features and requirements that you, the *Software Developer*, and the client have agreed should be implemented in the system.

As such, you should read these specifications carefully and ensure that your program implements the requirements of the specification correctly.

Tests will be run on your program to check that you have implemented these specifications correctly. **Note** that for the Final Project, we will provide 20 visible tests that you can run to check that the basic functionality of your system is correct.

> **There will be no hidden test cases. However, we will manually upload some different data files to rerun your program and assess the correctness of your code. Your final marks for the test cases will be assessed based on how many tests passed with the different data files that we upload. The use cases that are tested using these test cases will remain same. The data format will remain same but the data itself will change. DO NOT HARD CODE ANY DATA IN YOUR PROGRAM.**

So once your program passes the basic tests, *we strongly recommend that you perform further tests yourself*, to ensure that the required features have been implemented correctly.

### How to read the specifications?

- We will show some of the scenarios as valid/expected/best-case scenarios. This section will be marked in green. This means that for these sections we will not provide any incorrect inputs or ill-formatted inputs.

> **Best Case Scenario: This section describes a real-life intended scenario.**

- Some inputs to the program can be invalid and should be handled accordingly by the program. These unexpected but valid scenarios will be tested and are shown below -

> **Worst Case Scenario: Your program may behave incorrectly if not handled explicitly.**

- Just like the previous assignment, please read carefully through the **out-of-scope specifications** as well. This may be embedded as a red strip in the specifications sometimes.

> **Out of Scope Scenario: This scenario ........ is out of scope for this assignment**

- Other than this, we will use the general informational, warning, error, and assumption callouts using blue, yellow, red, and green coloured callouts.
- Also, note that the code snippets have some characters in bold that represent the inputs to the program.

> Students can assume that test cases only contain outputs that are explicitly part of specifications. The test cases do not produce any output that is not mentioned in the specifications explicitly. Please note that the specifications will give you warnings and important notes where we expect special input handling. You should observe those and implement them accordingly as they will be tested while marking your program.

# Preamble: Intended Learning Outcomes

The Intended Learning Outcomes for the final Project are mentioned below -

Similar to previous assignments -

- **Control Flows** - use branching and looping to navigate the special use cases in specifications.
- **Classes** - identify the entities and encapsulate their data and actions together in a class.
- **Arrays or ArrayLists** - to use 1D and 2D arrays/ArrayList and perform associated operations
- **Packages** - identify and group logical entities(classes) together
- **Javadoc** - generate javadocs from comments used in classes

Exclusive to the final Project -

- **UML** - generate a UML Diagram for the classes
- **Inheritance** - implement polymorphism correctly
- **Interfaces** - implement some operations via interfaces
- **File Handling** - reading and writing data from/to different files
- **Exception Handling** - handle exceptions gracefully and appropriately
- **Java Collections Framework** - using ArrayList or HashMaps to store and process data efficiently
- **Generics** - Generics is optional for this final Project but you can gain some bonus marks if Generics is implemented correctly. **Implementing Generics is not same as using Generics like ArrayList<>.**

⚠ A warning to those who have previous programming experience in other programming languages (like C or Python) that follow a procedural programming paradigm: Be careful to develop your program using object-oriented principles, as programming and structuring your code in a way that is not object-oriented is an easy way to lose marks for structure in the assignments.

## Preamble: Structure and Style

We will also be assessing your code for good structure and style.

Use methods when appropriate to simplify your code, and **avoid duplicate code.**

**Use correct Java naming conventions** for class names, variable names, and method names and put them in a well-organised way in your code i.e. it should be readable as well. Look at the conventions provided by Oracle **here.**

**Code structure** in a file is very important and **improves readability**. Look at the code organisation conventions provided by Oracle **here.**

Make sure the **names** you choose are **meaningful**, to improve the readability of your code.

Ensure to add meaningful comments in between your code and **Javadoc comments** for classes and methods.

> We will provide a marking scheme to help guide you in the development of your program.

## Academic Honesty

- All assessment items (assignments, tests, and exams) must be your own, individual, original work.
- Any code that is submitted for assessment will be automatically compared against other students' code and other code sources using sophisticated similarity-checking software.
- Cases of potential copying or submitting code that is not your own may lead to a formal academic misconduct hearing.
- Potential penalties can include getting zero for the project, failing the subject, or even expulsion from the university in extreme cases.
- Generative AI Tools like ChatGPT, Github CoPilot etc are prohibited.
- For further information, please see the university's Academic Honesty and Plagiarism website, or ask your lecturer.

You can find the Academic Integrity Module in your Canvas here.

**DO NOT SHARE YOUR CODE WITH ANYONE WITHIN OR OUTSIDE OF THE UNIVERSITY. DO NOT COPY FROM GPT or COPILOT. Severe actions will be taken.**

## File Handling & Command Line Args

# Command Line Arguments

Your program will either have two command line inputs or none. These are -

- User name (this will only be a first name)
- Input file name for the list of playlists.

⚠ **Worst Case Scenario: Your program may behave incorrectly if the optional params are not handled explicitly. These inputs will be tested with varied data. The data files name may change but will always be String.**

```
$java MusifyApp John playlistall.txt
```

If there are no command line arguments present then you will -

- set the user name as "Stranger"
- No file processing is needed to read the initial data. There will be no playlist set at the beginning of the program.

# File Handling

If both command line arguments are present then it is important to read some data from the files and load them in the appropriate objects of classes. There are three types of files available to you. These files are present in the data folder.

**Folder Structure**

- data - There will be a data folder next to your code files and packages. This is the root folder that contains all the files and subfolders.
- playlist - within the data folder, there will be a playlist folder. This folder contains files that define individual playlists for different kinds of media.
- mediatext - within the data folder, there will be another folder named mediatext. This folder contains files for captions/subtitles for different media.

✓ **Assumption: While handling the files in code, you can assume the folder names remain unchanged and append the folder names to the files. If the folder names don't change then they should be created as __________ ?**

**Files Description**

1. PlaylistList - This file contains data for all the playlists that the user has created. There will only be one file that has a list of all playlists. The name of the file can vary. This file will be present in the "data" folder.
2. Playlist- This file contains all the songs, podcasts or short clips that the user has added to one single playlist. There will be many playlist files.
3. Captions - This file contains subtitles or captions for different kinds of media. There will be many captions files.

**Note that the file names may change. The folder names won't change. You must not hardcode the file names in the code.**

## PlaylistList File

This file contains the list of all playlists that the user has created. This file has data lines in comma-separated form. Example -

```
HEALTHFREAK,PODCAST,pod_health
```

Each line has data in the order -

- Playlist name - a unique name for the playlist
- MediaType - SONG, PODCAST, and SHORTCLIP are the valid media types.
- file name - Name of the file containing the list of media within a playlist.

Total Data Points: 3 fixed data points.

## Playlist File

Based on the media type, each playlist file can have different data. These are defined below -

**Playlist of Songs**

Example -

```
Mama I'm Coming Home,something,Ozzy Osbourne,rock,3,ozzy_home
```

- name - name of the song
- description - some description of the song
- Artist Name - # separated Artist Name. If a song has one artist, then one name is present (without any # in the text). If a song has more than one artist then the artist names are separated by #. For example - Ed Shereen#Taylor Swift
- Genre - genre of the song (either *POP*, *ROCK*, *JAZZ*)
- duration in mins - duration of the song in mins. **This is expected to be an integer.**
- caption filename - filename where the song lyrics (captions) can be found.

Total Data Points: 6 fixed data points.

### Playlist of Podcast

Example

```
Hitting Rock Bottom,fighting anxiety and depression,Lewis Howes#Rachel Platten,HEALTH,Mental Health
```

- name - name of the podcast
- description - some description of the podcast
- Host Name - # separated Host Names. If a podcast has one speaker, then one name is present (without any # in the text). If a podcast has more than one host then the host names are separated by #. For example - Lewis#Rachel
- Category - category of the podcast (one of these values - *HEALTH*, *EDUCATION*, *TECHNOLOGY)*
- Series name - the name of the series the podcast belongs to
- Episode number - episode number of the podcast series. **This is expected to be a positive integer(> 0).**
- duration in minutes - the duration of the song in minutes. **This is expected to be a positive integer (> 0).**
- caption filename - filename where the song lyrics (captions) can be found.

Total Data Points: 8 fixed data points.

### Playlist of ShortClip

Example -

```
Dance steps,learn to dance,Bob,1,bob_dance
```

- name - the name of the short clip
- description - some description of the short clip
- Artist Name - A short clip only has one artist and their name is mentioned here.
- duration in mins - duration of the short clip in mins. **This is expected to be an integer.**
- caption filename - filename where the captions for the short clip be found.

Total Data Points: 5 fixed data points.

## Captions File

This file only contains a bunch of lines that are free text representing captions for the media. You should maintain the ordering of the texts but don't have to parse the string into any fixed data points.

> Note that the files do not have a header describing the column names so you don't have to do special handling of the header vs the data.

When you quit the program, your program should write back any changes done to PlaylistList, and playlists back to the file/folder paths mentioned. **Please do not override the captions files.**

⚠ **IMPORTANT NOTE:** Sometimes file handling is buffered internally by the operating system as well. This means if you PrintWriter.print , you may end up with an empty file. This can be an intermittent issue. To avoid this, please use **.flush()** methods.

## Order of reading the files

1. You should start by reading the PlaylistList file present in the command line args. The command line args does not show folder names. You should take the filename append **"data/"** to it and read the file in the code. Note If no command line args are present, then nothing needs to be done.
2. Once you have iterated through all the rows of the PlaylistList file mentioned in the command line args, then you should start processing individual playlist files. For example, if the PlayListlist file contains these two data rows

```
90's Rock,SONG,90_rock
HEALTHFREAK,PODCAST,pod_health
```

this means that there should be at least two files in the **"data/playlist/" folder.** You should append the folder name to file names like 90_rock or pod_health and try reading it in the code.

3. While iterating through the playlist file, say 90_rock, you will read each individual line, and then also process the captions file associated with it. For example, if 90_rock file has the content -

```
Iris,Song by a band,Goo Goo Dolls,ROCK,5,iris_song
Mama I'm Coming Home,something,Ozzy Osbourne,rock,3,ozzy_home
Knockin' on Heaven's Door,1991 Illusion,Guns n' Roses,ROCK,4
```

You should process the line containing song Iris and then the caption file iris_song before moving to the next line.

ℹ **Note: Remember in java the file extensions doesn't matter as long as the content is text in nature. So you may or may not have any file extensions like .txt**

## Writing the files back

You must write the playlistList file back using the filename provided to you in the command line argument. If no command line argument is passed, then you must create the file using the name "**playlists.txt".**

Similarly, you must write the playlist files as per the name provided by the user or the file names that were already present in the directory "data/playlist/" when you modify these playlists. In case the user created new playlist, you must create a new file. If a media is added to the playlist then the playlist files under **"data/playlist"** folder should be updated with the contents.

⚠ **Captions files are neither created nor updated in the program.**

✕ **NOTE: Windows vs Linux FileSystems handle folder path differently, You must comply to what EdStem follows i.e. Linux style paths.**

## Exception Handling with Files

Several exceptions may happen in the program.

**Worst Case Scenario: Your program may behave incorrectly if not handled explicitly. These kind of inputs will be tested.**

1. **FileNotFoundException or IOException** - The file paths that are provided to you are not present in the directory. Or are unavailable for File read/write operations. In this case, Java raises FileNotFoundException or IOException. Your program should handle these correctly, print appropriate messages and continue processing other data. For example - this is printed when the file for PlaylistList/playlist is not found. For captions see the MediaNotFoundException below.

```
Invalid or missing file.
```

2. **InvalidLineException -** Every file expects a minimum number of fixed data points. When these data points are missing, you should raise an InvalidLineException, skip reading the line, print an error message and move on to the next line. One of the following exception messages should be printed.

```
Invalid Playlist data. Skipping this line.
Song details incomplete. Skipping this line.
Podcast details incomplete. Skipping this line.
ShortClip details incomplete. Skipping this line.
```

3. **InvalidFormatException -** Some data points are expected to be in a particular format. For example, duration in minutes or episode number can only be an integer. Also, the Genre for the song or Category for the podcast are fixed. But you may have some invalid data in the files. In such cases, you must raise an InvalidFormatException and skip reading the line, print an error message and move on to the next line. One of the following exception messages should be printed.

```
Incorrect Media Type. Skipping this line. // This is printed when media type is not SONG,PODCAST,SH
Duration in mins not in correct format. Skipping this line. // This is printed when duration in min
Incorrect Genre for Song. Skipping this line. // This is printed when genre is anything else than P
Episode number not in correct format. Skipping this line. // This is printed when episode number fo
Incorrect Category for Podcast. Skipping this line. // This is printed when category is anything el
```

4. **PlaylistFullException -** A playlist can only have 5 media (song/podcast/short clip). If you try to add more than 5 songs, then you will raise a **PlaylistFullException.** You will print a customised message with the playlist name and mediaType

```
Syntax: "Playlist "+ playlistName +" is full. You cannot add new "+ mediaType +" to this playlist."
Example: Playlist 90's Rock is full. You cannot add new song to this playlist.
```

5. **MediaNotFoundException:** While processing the playlist files, you must look up the captions file. If the caption file is not found, then throw a FileNotFoundException or IOException, print a message and continue processing further data. Similarly, when you are adding new media to the playlist, the program will prompt you to add a captions file name. If the file is not found then throw a FileNotFoundException with the below message.

```
Invalid or missing caption file.
```

but while playing the media if the captions are not found then you will raise MediaNotFoundException and print one of these messages.

```
Cannot show captions for short clip. Media not found.
Cannot show captions for podcast. Media not found.
Cannot show lyrics. Media not found.
```

# Main Menu

## Program Display Message

If no command line arguments are given, the program should show a warning message and a display message. The username is considered to be "Stranger".

```
No Playlist data found to load.
Welcome **Stranger**. Choose your music, podcasts or watch short clips.Please select one of the options
```

If the program has both command line arguments, then it should show the appropriate Data Load messages and welcome messages.

```
Data loading complete.
Welcome **Luke**. Choose your music, podcasts or watch short clips.Please select one of the options.
```

In case the program finds some errors and skips some lines while loading data, it should print those error messages as well.

```
Incorrect Media Type. Skipping this line.
Incorrect Genre for Song. Skipping this line.
Duration in mins not in correct format. Skipping this line.
Song details incomplete. Skipping this line.
Invalid or missing caption file.
Data loading complete.
Welcome **Luke**. Choose your music, podcasts or watch short clips.Please select one of the options.
```

Once the program has loaded the data files, and shown the welcome message, it should print the main menu.

```
Incorrect Media Type. Skipping this line.
Incorrect Genre for Song. Skipping this line.
Duration in mins not in correct format. Skipping this line.
Song details incomplete. Skipping this line.
Invalid or missing caption file.
Data loading complete.
Welcome Luke. Choose your music, podcasts or watch short clips.Please select one of the options.
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
```

## Option 2: View all Playlist

Once all the playlists are loaded or a new playlist is created, the user should be able to view them by choosing option 2.

```
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
2
Here are your playlists-
| #|      Type|                 Playlist Name|
----------------------------------------------
| 1|      SONG|           Top 5 International|
| 2|      SONG|                     90's Rock|
| 3|   PODCAST|                   HEALTHFREAK|
| 4| SHORTCLIP|                  DAILY Shorts|
Please select one of the options.
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
```

i Use the formatter string ***PLAYLIST_HEADER_FORMATTER*** and ***PLAYLIST_FORMATTER*** provided to you in the constants file to create the formatted table.

## Option 1: Create a new playlist

The user can create a new playlist by choosing option 1. The user can then add some media content to it. Or the user can defer it to later.

### Case 1: User creates empty playlist.

```
Please select one of the options.
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
1
Enter Playlist Name: Top Songs
Enter Playlist Type: SONG
Enter a filename to save the playlist: top_song
Add some SONG to your Playlist.
Enter A to add a SONG to the playlist or Q to quit adding: Q
Please select one of the options.
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
2
Here are your playlists-
| #|      Type|                 Playlist Name|
----------------------------------------------
| 1|      SONG|           Top 5 International|
| 2|      SONG|                     90's Rock|
| 3|   PODCAST|                   HEALTHFREAK|
| 4| SHORTCLIP|                  DAILY Shorts|
| 5|      SONG|                     Top Songs|
```

**Case 2:** The user creates a playlist and adds media to it.

```
Please select one of the options.
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
1
Enter Playlist Name: MENTALHEALTH
Enter Playlist Type: PODCAST
Enter a filename to save the playlist: mental
Add some PODCAST to your Playlist.
Enter A to add a PODCAST to the playlist or Q to quit adding: A
Enter the title: Heartbreak
Enter the description: how to cope
Enter duration in mins: 45
Enter the filename for the captions or lyrics: heartbreak
Invalid or missing caption file.
Add the category: HEALTH
Enter the host Name or Q to stop entering the host name: Somehost
Enter the host Name or Q to stop entering the host name: Someelse
Enter the host Name or Q to stop entering the host name: Q
Enter the series Name: MentalhealthSeries
Enter the episode Number: 1
Enter A to add a PODCAST to the playlist or Q to quit adding: Q
Please select one of the options.
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
```

```
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
2
Here are your playlists-
| #|      Type|                 Playlist Name|
----------------------------------------------
| 1|      SONG|           Top 5 International|
| 2|      SONG|                     90's Rock|
| 3|   PODCAST|                   HEALTHFREAK|
| 4| SHORTCLIP|                  DAILY Shorts|
| 5|      SONG|                     Top Songs|
| 6|   PODCAST|                  MENTALHEALTH|
Please select one of the options.
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
```

### How to create a playlist for songs or short clips?

The creation of the playlist is the same. Takes three inputs - playlist name, media type, and file name. However, adding media to it like songs/ podcasts/ shortclips will require different kinds of inputs. These are discussed here.

## Option 3: View the contents of a playlist

You can view the contents/media of a playlist by selecting option 3 and then selecting the playlist name.

If the playlist has some content then that will be displayed, else If there is no content then an error message `No <mediatype> in the playlist to view.` should be shown. The mediatype could be `song` , `podcast`, `shortclip`

```
Please select one of the options.
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
3
Enter Playlist Name: mentalhealth
|Id|                         Title|                  Host Name(s)|                   Description|
----------------------------------------------------------------------------------------------------
| 1|                    Heartbreak|             Somehost,Someelse|                   how to cope|
Please select one of the options.
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
3
Enter Playlist Name: Top Songs
No song in the playlist to view.
Please select one of the options.
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
```

**Note that the playlist name should match in any case.**

**Use the formatters present in the Constants file for displaying Song/Podcast/ShortClip Playlists.**

## Option 4: Removing a playlist

The user can remove a playlist by selecting option 4 and then entering the name of the playlist to be removed.

```
Please select one of the options.
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
4
Enter Playlist Name: Top Songs
Playlist removed successfully.
Please select one of the options.
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
2
Here are your playlists-
```

```
| #|      Type|                 Playlist Name|
----------------------------------------------
| 1|      SONG|           Top 5 International|
| 2|      SONG|                     90's Rock|
| 3|   PODCAST|                   HEALTHFREAK|
| 4| SHORTCLIP|                  DAILY Shorts|
| 5|   PODCAST|                  MENTALHEALTH|
Please select one of the options.
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
```

**Removing playlist doesn't mean you should delete the playlist filename from the folder. No Deletions from any folder are required.**

**Assumption: Can we have two playlist with the same name? No, such a case won't be tested.**

## Option 6: Play the contents of a playlist

When you select option 6 and enter a playlist name, you can play the media one by one in sequence. You should show a message about what you are playing and then print the captions of it. If you cannot find the captions, then you should print an error message - `Cannot show captions for <mediatype>. Media not found.` The <mediatype> is to be replaced by the actual media type. See examples below.

### Case: Playing a song playlist

```
Please select one of the options.
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
6
Enter Playlist Name: Top 5 international
-----------------------------------------------------------------------------------
Playing Song: Believer by Imagine Dragon for 3 mins.
Here are the lyrics to sing along.
You made me a, you made me a believer, believer
Pain!
You break me down and build me up, believer, believer
-----------------------------------------------------------------------------------
-----------------------------------------------------------------------------------
Playing Song: Houdini by Dua Lipa for 4 mins.
```

```
Here are the lyrics to sing along.
I'm not here for long
Catch me or I go Houdini
-----------------------------------------------------------------------------------
-----------------------------------------------------------------------------------
Playing Song: Let her go by Passenger,Ed Shereen for 4 mins.
Here are the lyrics to sing along.
Only know you love her when you let her go
And you let her go
-----------------------------------------------------------------------------------
-----------------------------------------------------------------------------------
Playing Song: Fortnight by Taylor Swift for 3 mins.
Here are the lyrics to sing along.
And for a fortnight there, we were forever
Run into you sometimes, ask about the weather
-----------------------------------------------------------------------------------
Please select one of the options.
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
```

### Case: Playing a podcast

```
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
6
Enter Playlist Name: healthfreak
-----------------------------------------------------------------------------------
Playing Podcast: Hitting Rock Bottom by Lewis Howes,Rachel Platten for 45 mins. This podcast is abo
Here are the contents of the podcast.
some text about mental health
-----------------------------------------------------------------------------------
-----------------------------------------------------------------------------------
Playing Podcast: LightSpark by Kevin Hurley for 90 mins. This podcast is about Code Story
Cannot show captions for podcast. Media not found.
-----------------------------------------------------------------------------------
Please select one of the options.
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
```

### Case: Playing a DailyShort

```
Please select one of the options.
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
6
Enter Playlist Name: daily shorts
-----------------------------------------------------------------------------------
Playing short clip: Dance steps by Bob for 1 mins.
Here are the contents of the short clip.
We learn salsa by doinng these steps.
-----------------------------------------------------------------------------------
-----------------------------------------------------------------------------------
Playing short clip: Chicken paella by Maria for 1 mins.
Cannot show captions for short clip. Media not found.
-----------------------------------------------------------------------------------
Please select one of the options.
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
```

## Option 7: Exit

The program must save the playlist data into respective files and display a message before terminating.

```
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
7
Playlist data saved.
Exiting Musify. Goodbye, Luke.
```

## Option 5: Modifying a playlist

To modify a playlist, the user can select option 5 and enter a playlist name. This will open up a sub-menu that is discussed here.

```
Please select one of the options.
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
5
Enter Playlist Name: Top 5 International
Please select one of the options.
1. View the playlist.
2. Add a new song.
3. Remove a song.
4. Exit and go back to main menu.
```

## No Playlist found

In case the user enters option 3,4,5,6 and there are no playlists in the list of playlists, then you should show an error message `No playlists found.`

```
No Playlist data found to load.
Welcome Stranger. Choose your music, podcasts or watch short clips.Please select one of the options
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
2
No playlists found.
Please select one of the options.
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
3
No playlists found.
Please select one of the options.
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
5
No playlists found.
```

```
Please select one of the options.
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
7
Exiting Musify. Goodbye, Stranger.
```

## Incorrect Playlist as input

In case you enter a playlist that is not found, you should print the error message `No such playlist found with name: Top 5`. This is valid for Option 3, 4, 5, 6.

```
Please select one of the options.
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
6
Enter Playlist Name: Top 5
No such playlist found with name: Top 5
Please select one of the options.
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
4
Enter Playlist Name: Top 5
No such playlist found with name: Top 5
Please select one of the options.
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
3
Enter Playlist Name: Top 5
No such playlist found with name: Top 5
Please select one of the options.
1. Create a new playlist.
2. View all playlist.
```

```
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
```

## Invalid Input to Main Menu

Please note that we can provide any kind of data format - integer, string, double etc. as input to your main menu. Your program should be able to handle all of these and show the message `Invalid Input`

```
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
-9.9
Invalid Input
Please select one of the options.
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
```

## Modifying Playlist SubMenu

Once the user selects option 5 and enters a playlist name, they can modify the playlist by adding or removing media or viewing the playlist. Viewing the playlist is the same as Option 3 from the main menu.

We will cover three different use cases here -

- Adding/Removing a song
- Adding/Removing a podcast
- Adding/Removing a short clip

Also while adding media, you will face these scenarios -

- While adding a song/podcast the program can expect a valid Genre/Category respectively.
- You can add a media to the playlist but if you are trying to add more than 5 media in a playlist, it will throw a PlayListFullException with the error message as shown below.
- You can add a caption filename. It tries to find the caption filename but if it cannot find one, it will show an exception message for FileNotFoundException with message `Invalid or missing caption file.`
- Incorrect media name while removing a media. You should show an error message.
- Removing media from an empty playlist.
- Viewing an empty playlist.

They are covered in the example below.

**Case: Adding/Removing a song**

- see the playlist full error message

```
Please select one of the options.
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
5
Enter Playlist Name: Top 5 International
Please select one of the options.
1. View the playlist.
2. Add a new song.
3. Remove a song.
4. Exit and go back to main menu.
1
```

```
|Id|                         Title|                   Artist Name|                   Description|
---------------------------------------------------------------------------------------------------
| 1|                      Believer|                Imagine Dragon|             Song album Evolve|
| 2|                       Houdini|                      Dua Lipa|Radical Optimism released 2024|
| 3|                    Let her go|          Passenger,Ed Shereen|                Mix by artists|
| 4|                     Fortnight|                  Taylor Swift|             Swiftie favourite|
Please select one of the options.
1. View the playlist.
2. Add a new song.
3. Remove a song.
4. Exit and go back to main menu.
2
Enter the title: Radioactive
Enter the description: some song
Enter duration in mins: 4
Enter the filename for the captions or lyrics: radioactive
Add the Genre: POP
Enter the artist Name or Q to stop entering the artist name: Imagine Dragons
Enter the artist Name or Q to stop entering the artist name: Q
Please select one of the options.
1. View the playlist.
2. Add a new song.
3. Remove a song.
4. Exit and go back to main menu.
1
|Id|                         Title|                   Artist Name|                   Description|
---------------------------------------------------------------------------------------------------
| 1|                      Believer|                Imagine Dragon|             Song album Evolve|
| 2|                       Houdini|                      Dua Lipa|Radical Optimism released 2024|
| 3|                    Let her go|          Passenger,Ed Shereen|                Mix by artists|
| 4|                     Fortnight|                  Taylor Swift|             Swiftie favourite|
| 5|                   Radioactive|               Imagine Dragons|                     some song|
Please select one of the options.
1. View the playlist.
2. Add a new song.
3. Remove a song.
4. Exit and go back to main menu.
2
Playlist Top 5 International is full. You cannot add new song to this playlist.
Please select one of the options.
1. View the playlist.
2. Add a new song.
3. Remove a song.
4. Exit and go back to main menu.
3
Enter the song to remove: radioactive
Please select one of the options.
1. View the playlist.
2. Add a new song.
3. Remove a song.
4. Exit and go back to main menu.
1
|Id|                         Title|                   Artist Name|                   Description|
---------------------------------------------------------------------------------------------------
```

```
| 1|                      Believer|                Imagine Dragon|             Song album Evolve|
| 2|                       Houdini|                      Dua Lipa|Radical Optimism released 2024|
| 3|                    Let her go|          Passenger,Ed Shereen|                Mix by artists|
| 4|                     Fortnight|                  Taylor Swift|             Swiftie favourite|
Please select one of the options.
1. View the playlist.
2. Add a new song.
3. Remove a song.
4. Exit and go back to main menu.
4
Please select one of the options.
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
```

### Case: Adding/Removing a Podcast

- see the error message for removing media that doesn't exist in the playlist
- see the error message when no media exists in the playlist anymore

```
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
5
Enter Playlist Name: healthfreak
Please select one of the options.
1. View the playlist.
2. Add a new podcast.
3. Remove a podcast.
4. Exit and go back to main menu.
1
|Id|                         Title|                  Host Name(s)|                   Description|
----------------------------------------------------------------------------------------------------
| 1|           Hitting Rock Bottom|    Lewis Howes,Rachel Platten|fighting anxiety and depression|
| 2|                    LightSpark|                  Kevin Hurley|                    Code Story|
Please select one of the options.
1. View the playlist.
2. Add a new podcast.
3. Remove a podcast.
4. Exit and go back to main menu.
3
Enter the podcast to remove: lights
No such media found with title: lights
Please select one of the options.
```

```
1. View the playlist.
2. Add a new podcast.
3. Remove a podcast.
4. Exit and go back to main menu.
3
Enter the podcast to remove: LightSpark
Please select one of the options.
1. View the playlist.
2. Add a new podcast.
3. Remove a podcast.
4. Exit and go back to main menu.
1
|Id|                         Title|                  Host Name(s)|                   Description|
----------------------------------------------------------------------------------------------------
| 1|           Hitting Rock Bottom|    Lewis Howes,Rachel Platten|fighting anxiety and depression|
Please select one of the options.
1. View the playlist.
2. Add a new podcast.
3. Remove a podcast.
4. Exit and go back to main menu.
3
Enter the podcast to remove: hitting rock bottom
Please select one of the options.
1. View the playlist.
2. Add a new podcast.
3. Remove a podcast.
4. Exit and go back to main menu.
1
No podcast in the playlist to view.
Please select one of the options.
1. View the playlist.
2. Add a new podcast.
3. Remove a podcast.
4. Exit and go back to main menu.
2
Enter the title: hitting bottom
Enter the description: fighting depression
Enter duration in mins: 90
Enter the filename for the captions or lyrics: lewis
Add the category: HEALTH
Enter the host Name or Q to stop entering the host name: Lewis Howes
Enter the host Name or Q to stop entering the host name: Rachel Platten
Enter the host Name or Q to stop entering the host name: Q
Enter the series Name: MentalHealth
Enter the episode Number: 1663
Please select one of the options.
1. View the playlist.
2. Add a new podcast.
3. Remove a podcast.
4. Exit and go back to main menu.
1
|Id|                         Title|                  Host Name(s)|                   Description|
----------------------------------------------------------------------------------------------------
| 1|                hitting bottom|    Lewis Howes,Rachel Platten|           fighting depression|
Please select one of the options.
1. View the playlist.
2. Add a new podcast.
3. Remove a podcast.
4. Exit and go back to main menu.
4
Please select one of the options.
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
```

### Case: Adding/Removing a ShortClip

- see the invalid captions file error message
- see the error message when the entire playlist is empty and the user is trying to remove it from an empty playlist
- View an empty playlist

```
Please select one of the options.
1. Create a new playlist.
2. View all playlist.
3. View contents of a playlist.
4. Remove a playlist.
5. Modify a playlist.
6. Play contents of a playlist.
7. Exit Musify.
5
Enter Playlist Name: Daily shorts
Please select one of the options.
1. View the playlist.
2. Add a new shortclip.
3. Remove a shortclip.
4. Exit and go back to main menu.
1
|Id|                         Title|                   Artist Name|                   Description|Du
---------------------------------------------------------------------------------------------------
| 1|                   Dance steps|                           Bob|                learn to dance|
| 2|                Chicken paella|                         Maria|                  spanish dish|
Please select one of the options.
1. View the playlist.
2. Add a new shortclip.
3. Remove a shortclip.
4. Exit and go back to main menu.
2
Enter the title: Chicken burrito
Enter the description: spanish dish
Enter duration in mins: 1
Enter the filename for the captions or lyrics: burrito
```

```
Invalid or missing caption file.
Enter the artist Name: someone
Please select one of the options.
1. View the playlist.
2. Add a new shortclip.
3. Remove a shortclip.
4. Exit and go back to main menu.
1
|Id|                         Title|                   Artist Name|                   Description|Du
---------------------------------------------------------------------------------------------------
| 1|                   Dance steps|                           Bob|                learn to dance|
| 2|                Chicken paella|                         Maria|                  spanish dish|
| 3|               Chicken burrito|                       someone|                  spanish dish|
Please select one of the options.
1. View the playlist.
2. Add a new shortclip.
3. Remove a shortclip.
4. Exit and go back to main menu.
3
Enter the shortclip to remove: chicken paella
Media removed successfully.
Please select one of the options.
1. View the playlist.
2. Add a new shortclip.
3. Remove a shortclip.
4. Exit and go back to main menu.
3
Enter the shortclip to remove: chicken burrito
Media removed successfully.
Please select one of the options.
1. View the playlist.
2. Add a new shortclip.
3. Remove a shortclip.
4. Exit and go back to main menu.
3
Enter the shortclip to remove: dance steps
Media removed successfully.
Please select one of the options.
1. View the playlist.
2. Add a new shortclip.
3. Remove a shortclip.
4. Exit and go back to main menu.
3
You can not remove media from an empty list.
Please select one of the options.
1. View the playlist.
2. Add a new shortclip.
3. Remove a shortclip.
4. Exit and go back to main menu.
1
No shortclip in the playlist to view.
Please select one of the options.
1. View the playlist.
2. Add a new shortclip.
```

```
3. Remove a shortclip.
4. Exit and go back to main menu.
```

⚠ **While removing a media from playlist, you must not delete any files from the folders.**

# Guidance: Object Oriented Programming

## Quick Tips

- Now that you know about UML, perhaps start designing your solutions by creating a UML first. Think about what classes need to be created. Some of them are provided to you as the scaffold. You can create other classes as well.
- The second step is to associate appropriate data with appropriate classes. Create the data fields as instance variables in those classes.
- The third step would be to create the exceptions.
- The next step is file handling. This will take the maximum amount of time.
- The next step is to create a structure for running the menu options with different Control Flows you have learned.
- Exceptions thrown in a method are not handled in the same method. You should handle them with try-catch block in some other methods where you are invoking the method that is causing the exception. For example - when you use Integer.parseInt and the parseInt method throws a NumberFormatException, you handle it in the method where you are using the Integer.parseInt.
- create packages like previous assignments to logically group entities and interfaces and exceptions as well.
- generics are optional but may gain you some bonus marks.
- Think about enums.

## Interfaces & Inheritance

- Implement the inheritance between related entities. You must have at least one inheritance hierarchy.
- Think about what are the common operations for different entities. Interfaces are implemented where unrelated/related entities can have similar behaviour but different implementations. You must create and implement at least one interface.

## File Parsing

- Use a simple Scanner and PrintWriter instead of over-complicating the reading/writing. Try to minimise duplicate code associated with reading writing.
- When you read the files, you can read the entire line and then split the data on commas. This shall provide you with different data points in an array. Check the split method in the String class here. You can use this method to perform a split on other special characters present in data like hash( # ) as well.

# Musify App

The starter code have been provided for you. Some classes are provided but packages are not created. You must create some packages and move the classes to the appropriate package except **MusifyApp.java**. **MusifyApp.java should not be moved under any package else your code won't run.** Write the rest of the code to implement your program there ▯

We also provided some tests for your code, which will run automatically every time you hit "Mark". You can run these tests as many times as you would like until the submission deadline.

⚠ **Submission will close on 18 Oct 2024 at 1700 AEDT.**

Happy coding!

## Assessment

This project is worth **40%** of the total marks for the subject. Your Java program will be assessed based on the correctness of the output as well as the quality of code implementation.

Automatic tests will be conducted on your program by compiling, running, and comparing your outputs for several test cases with generated expected outputs. The automatic test will deem your output wrong if your output does not match the expected output, even if the difference is just having extra space or missing a colon. Therefore, it is crucial that your output follows exactly the same format shown in the provided examples.

Passing the tests we provide here on edstem does not mean you will score full marks. Each submission will further be subject to additional tests using different dataset as well as manual inspection for code quality by our teaching team.

## Submission

Your submission should have

- MusifyApp.java file.
- other packages and java files in those packages.
- **a UML Diagram in PDF format mentioning** all the classes and the multiplicity/association between those classes.

✕ **You must not move this to any packages otherwise your code wont compile.**

A starter code has been provided to you here on the edstem platform, but you are welcome to develop the project outside of the edstem environment, using your own IDEs. The test files can be found here -

tests.zip

**Submission is made via edstem**, however, and will be whatever code is saved on the Code Challenge when the project deadline closes.

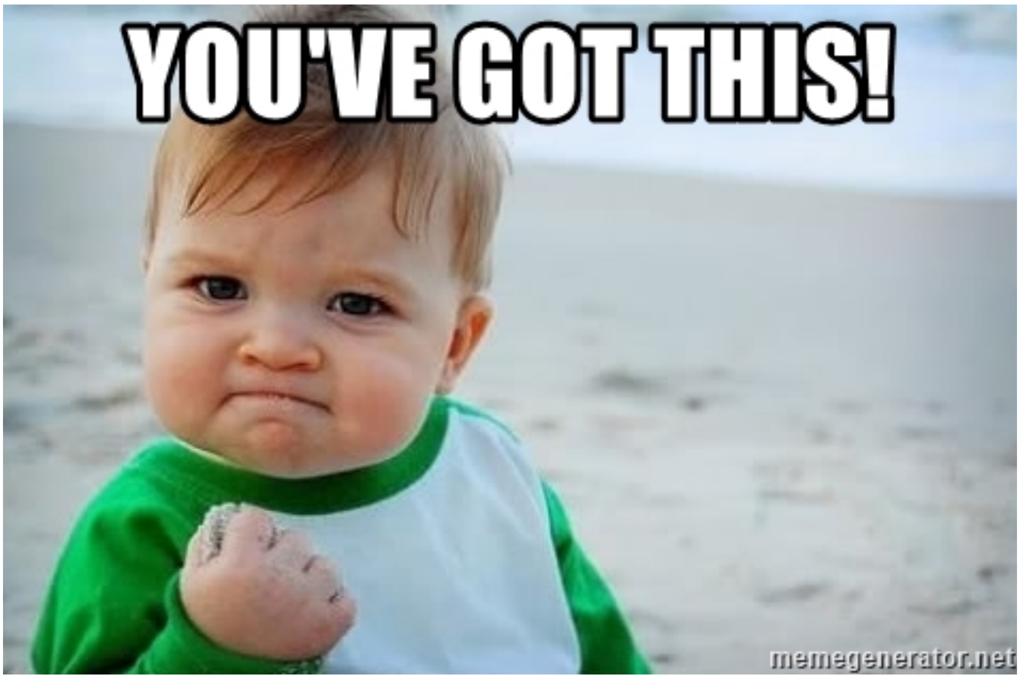
⚠ Your code MUST compile and run here on edstem. If your code does not compile we cannot mark it and you risk getting 0 marks.

*"I can't get my code to work on edstem but it worked on my local machine"* is not an acceptable excuse for late submissions. In addition, **submissions via email will not be accepted.**

Be sure to copy your code into your Code Challenge workspace well before the deadline to avoid being locked out of submissions last minute.

**Only the last version of your code before the submission deadline will be graded**. It is highly recommended that you update your code on edstem frequently and well before the submission deadline. Last-minute "*connection errors*" are not a valid excuse.

In case you need an extension due to valid reasons up to 3 days, please fill out the form mentioned in the slides here and send it to the subject coordinator's email. We will follow up with you. Ensure you have a valid reason and proper documentation with you before seeking the extension. If you seek an extension beyond 3 days, check the new Special Consideration Module on Canvas and raise your request using this portal. See more here.


YOU'VE GOT THIS!
memegenerator.net

# FAQ

1. What happens if we enter the program and then exit it without modifying any data? Will the files need to be stored?
   **Answer:** Yes
2. What happens if there are no command line arguments and then we exit the program without creating any playlist? Will the data need to be stored?
   **Answer:** Yes, in this case, you will try to store the data but you won't find anything to store. So no files will be generated. Remember to put appropriate null checks.
3. Why do I not get the error messages after the first test case passes?
   **Answer:** Usually once you save a file, your unwanted lines will be removed from the saved files and you will no longer have the error messages in the subsequent test runs.
4. Can we enter an incorrect value of Genre/Category from the command line?
   **Answer**: No, that case is not tested. But you can have these values incorrectly in the data files.
5. In displaying the playlist content the host/artist names are shown comma separated whereas in the files they are shown # separated. Is this correct behaviour?
   **Answer: Yes,** because in the file all data are comma separated. if the artist names are comma separated then you will have more than 6 or 8 data points for songs/podcast files respectively and will be confused how to process them.