CSE201 Advanced Programming (CSE Section)
Lab Assignment 02

IIIT-Delhi. 24th August 2018. Due by 11:59pm on 25th Aug 2018

This is a take-home lab assignment. No extensions whatsoever will be provided. Any submission after the deadline will not be evaluated. If you see any ambiguity or inconsistency in a question, please seek a clarification from the teaching staff. Please read the entire text below very carefully before starting your implementation.

Plagiarism: All submitted lab assignments 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 IIITD plagiarism policy and without any relaxations: https://www.iiitd.ac.in/sites/default/files/docs/education/AcademicDishonesty.pdf

Please note that you are **not** allowed to discuss the design/solution of the lab assignment (e.g., backpack discussions, etc.). Anyone who is found doing this will be treated as plagiarism case.

Topics Covered: Interfaces and Polymorphism

Spotify is coming to India and they have contracted you to design their application to appeal to the Indian customers. They have certain requirements that need to be met based on the contract.

Their demands are as follows:

- There are two main types of users called artists and customers.
- Each artist follows certain guidelines which are as follows:
    - They can make new albums.
    - Each song played get the artist Rs. 5 and each download gets the artist Rs. 20.
    - They have unlimited streaming access on the music application.
- Each customer follows certain guidelines as follows:
    - They are chargeable based on the subscription fees that they choose.
    - Their streaming access is limited based on their subscription. Paid subscribers get the more features compared to Free users. These features include more skips, more blocks, more downloads, more favourites in a music playlist.
    - The subscriptions as defined below :
        - Free - 1 song, 0 downloads
        - Rs. 100 - max 5 songs, max 2 downloads
        - Rs. 500 - max 10 songs, max 5 downloads
        - Rs. 1000 - Unlimited access.

Following are the different kind of queries that must be handled by the system:

1. *Create new artist or customer.*

   a. Each artist and customer have a unique ID and a name.
   b. Default customer subscription is free
   c. Each query specifies whether it is an artist(A) or customer(C) followed by their name. Output their name and the system generated unique ID.

2. *Change customer's subscription*
   a. Take customer's unique ID as input and their preference of subscription based on the list provided above indexed from 1.
3. *Add new albums for the artist*
   a. Take artist's unique ID as input. After this the next input 'n' would indicate the number of songs to add. Each song will be space separated. You should output the unique ID for each song generated by the system in a single line space separated.
4. *Pay the artist if any royalties are pending*
   a. Print out the total amount that the application must pay the artist since the last time the royalties were paid. After this the artist's royalties go back to 0.
5. *Allow customer or artist to stream music*
   a. For customer take a list of songs as input which he want to stream and for each of those input also take whether they want to download of play the song. You need to calculate the fees for the customer based on their streaming sessions.
   b. The same should happen the artists but no royalties should be generated.
   c. First take the unique ID as input. The next input indicate the number of songs that the user wants to stream. Next input will be the songs where each song is specified whether the user wants to download(D) or play(P).
6. *Print the charges that customers needs to pay*
   a. Take the input as unique ID and then print out the charges for that customer based on their subscription fees.
7. *Print details of the customers or artists*.
   a. Take the input of customer ID and print their Name, unique ID and subscription ID based on the list provided before indexed from 1.
   b. Take the input of artist ID and print their Name, unique ID followed by a semicolon. Next print out the songs in a comma separated fashion.

Note: The system should not crash for any kind of invalid input. It should handle it properly, and display an appropriate error message. However, the following input format will always be followed. You should strictly follow object-oriented programming approach in your implementation as discussed in class lectures. There is no need to show us the sequence diagram.

Input Format: The first line will have an integer q for the number of queries in the application. Each of the next q lines will have variable number of space-separated inputs. The first input in the line will always be an integer specifying the query type. The rest of the inputs in the line will depend on the type of query to be performed.

Sample Case:

Input:

```
1 A A
1 C B
2 35 3
3 25
4
A B C D
5 35
3
1 D 2 P 3 D
5 25
4
1 D 2 D 3 D 4 D
4 25
6 35
7 35
7 25
```

Output:

```
A 25
B 35
A 1 B 2 C 3 D 4
45
500
B 35 3
A 25: A, B, C, D
```

For Bonus:

Bonus is to come up with your own sorting algorithm (the algorithm should be efficient in terms of complexity) and use that sorting algorithm to sort the selection of music that a user specifies based on his preference. Bonus marks will only be given if no inbuilt sorting API is used.
There will be one change in this case which is in the query 5 where once the songs are given as input there will be another line of input which specifies the user's preference. Finally you need to output this updated stream by printing the sorted songs in a space separated fashion.

Sample Bonus Case:

Input:

```
1 A A
1 C B
2 35 3
3 25
4
A B C D
```

```
5 35
3
1 D 2 P 3 D
3 2 1
5 25
4
1 D 2 D 3 D 4 D
4 3 1 2
4 25
6 35
7 35
7 25
```

Output:

```
A 25
B 35
A 1 B 2 C 3 D 4
C B A
D C A B
45
500
B 35 3
A 25: A, B, C, D
```