Introduction To Programming
Tutorial 8

(See Canvas→Assignments for due dates and marks)

Note: Do not include your name, student ID or any personally identifiable info in your submission as the submission may be used for peer reviews; your submission will not be lost as Canvas keeps track of these internally.

Please follow all of the steps below in the given sequence:

1. Read all unread announcements and unread replies to announcements under Canvas→Announcements.

2.1 Do any missed tutorials before going further.

2.2 Watch any unwatched recordings of the compulsory **Weekly Live Lecture** and any important videos in the Extra Videos Playlist.

2.3 If you need help in addition to what has been shown in the compulsory weekly live lecture, you are also expected to speak to your **group tutor via discussion forums** and attend/watch their live sessions. Please note that group tutors cannot debug your assessment code on your behalf as debugging is a part of every programming assessment.

2.4 **If you still have have any unresolved questions** or **if you need further feedback**, post the relevant parts of your submitted work in a new post under Canvas→Discussions→Tutorial discussions and ask from your group tutor. E.g. you can ask "*In the live lesson Gayan did ___ with ___. I didn't do ___ so should I be doing this as well?*", etc. Please note that the university requires teaching to be conducted in an equitable manner so your tutors will require you to post questions in the discussion forums.

3. Check any available feedback of your previous submissions and if you have any unresolved questions or if you need further feedback, post the relevant parts of your submitted work in a new post under Canvas→Discussions→Tutorial discussions and ask from your tutor. E.g. you can ask "*Gayan showed _____ but I did mine like ______, so which is the better approach and why?*", etc. Please note that the university requires teaching to be conducted in an equitable manner so please only use email for matters such as special consideration.

4. Follow the materials under Canvas→Modules→Week 8…

5.1. With the help of your group tutors via the Canvas→Discussion forums and by using good Object Oriented (OO) coding practices (also required for Assignment 3), convert the MusicLibraryW7.java (from IIE7 or your latest version) to an OO application as per the following description:

Song.java: A *Song* object cannot be created without a *song title* and a String containing the *location* of the song in the computer (e.g. "c:/song.mp3"). It also has the relevant accessor (get) methods but no mutator (set) methods. A Song must not have any methods that will display messages. Only its constructor and the above two methods can be accessed by other classes. Write the code for this in a new class named **Song.java**.

MusicLibrary.java: A *MusicLibrary* object cannot be created without a *maximum number of songs*. A *MusicLibrary* object also has an array of *Song* objects (the size of this array can be set to the *maximum number of songs specified*) and the *number of currently added songs*. It must not have the separate arrays songTitles and songLocations anymore. Move all of the code from the main method to the constructor first then refactor the code in to different methods so that you have at least one method per menu item. Aside from the methods, no other component of this class should be accessible by other classes. As it is the application class, the *MusicLibrary* has a main method and this method must contain only the following line (then the return statement):

```
MusicLibrary ml=new MusicLibrary(100);
```

In the above, the 100 refers to the maximum number of songs.

Remember, when doing OO in Intro To Prog: Follow the guidelines given in Canvas→Assignments→Assignment 3 (guidelines). You must not create anything *static* other than *public static void main*. All object member variables must be *private* and they must only be initialised inside the constructor (including the creation of arrays). Any references to member variables from within methods should start with "`this.`". There is no need for private methods. You will need to write a constructor for each class and you must prevent the creation of objects that are invalid (e.g. don't create a second constructor that will allow you to create a *Song* object with no song title or location). Use only standard arrays in Java (e.g. avoid ArrayList, etc.). You must not use break, continue, System.exit(). Use only while loops for repetition. When possible, you must use concepts covered in standard class materials over others.

5.2. Add comments in the style required by Assignment 2/3. See rubric section in Assignment 2/3 PDF.

Submission Checklist (Compulsory):

1. Format your code (e.g. Eclipse→Source→Format).
2. It is OK if your independent investigative efforts before the lesson are not fully functional but a program with even one red dot under Eclipse (programs with syntax errors) are not runnable/testable Java and cannot be marked. Remove any parts of the code that result in syntax errors and explain the issues in code comments instead. Do not keep code that is commented out.
3. Go to Canvas→Assignments→**Independent Investigative Effort 8** and select 'submit assignment'.
4. Select to attach files from your computer, navigate to your Eclipse workspace folder→Project folder→src folder and select the final version of your **`MusicLibrary.java and Song.java`** file. Please **attach and submit both files in one go**; the university considers only the last submission as your final submission.
5. Verify your submitted file as shown during the week 1 chat session.

Having trouble with usernames, passwords, access, etc.? Please call the RMIT IT Service and Support Centre for quick help on 03-9925 8888 and remember to ask for a reference number and pass it on to your instructor.

Need extensions or special consideration? Please follow the RMIT Special Consideration page