

Introduction To Programming

Tutorial 9

(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 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 9](#)...

5.1 Complete this exercise with the help of your group tutor as it will help you come up with an idea for an Object Oriented (OO) application for Assignment 3. However, your tutor will not be able to give you ideas for OO application on your behalf; They can only give guidance on general OO concepts. In Assignment 3, you only get marks for the design that is your own (e.g. number and choice of classes, how they and their objects relate to each other, what operations they perform and what member variables they contain, etc.)

Write the answers to parts a, b, c, d, e, f, in the submission page itself (no file uploads for this week). Do not repeat the question text as it could give you a “similarity” rating to other students’ work. Just write the part number then your answer (sample answers shown in italics; yours must be different).

a) State the hypothetical name of your application. This would be your application class and you must state its name. E.g. *MusicLibrary*

b) What other things need to be there be in the system that you have multiples of? These are your objects. E.g. *array of songs in the MusicLibrary*

c) Which of these objects would be of the same type? In IIE8 every song had a song title and a location and therefore every song in the song array came from the same Song class. List the classes that you have identified, in addition to the application class. E.g. *Song*

d). For each class above, specify what object member variables would be there. E.g. the *Song class can have members String name, String location, int rating, int numberOfTimesPlayed* so that every Song object will have a unique set of values for each of these variables. (Note: All member variables must be non-static explicitly **private** in Intro To Prog).

e) From the member variables that you have identified in part d, which of these are mandatory when creating an object? In IIE8 we said that Song cannot be created without a *title and a location* (the other member variables such as rating and numberOfTimesPlayed are not that important to specify at the time of creating a Song object). These would be in your constructor’s parameters/arguments. For each class identified above, state the constructor’s parameters.

f) State at least one other important method that must be there in the application that is not a simple get/set (accessor/mutator) method. E.g. a Song object has the *getSongTitle* method and the *setSongTitle* method but these are too obvious. Instead we are interested in methods like *playSong* or *findSongs*, etc.

Note: Even if you have not thought everything through, something planned in writing is better than just an idea in your head. Your answers to a-f need not be final but you must show effort in these.

Continued on next page...

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.

6. If you have not submitted your final version of A3, add comments explaining your plan. Note that this will not be marked but it is to help you progress.

Submission Checklist:

1. Go to Canvas→Assignments→**Independent Investigative Effort 9** and select ‘submit assignment’.
2. Write the answer to this exercise in the submission page itself (**ensure answers are numbered and formatted appropriately**). **Do not submit any Java code** (just explanations).
3. Submit your work and check to see if there is a high “similarity rating”. If it does, modify your submission and explain why it might show a high similarity rating. Note that a high similarity does not automatically mean any issue but these should still be addressed.

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 details and process below:

<https://www.rmit.edu.au/students/student-essentials/assessment-and-exams/assessment/special-consideration>