

COIT11134 Object Oriented Programming (T1 2019)

Assignment Item 2 – Dynamic Data Structures & File Processing

Due date:	Week 10 Friday (24-May-2019) 11:59 PM AEST	ASSESSMENT
Weighting:	20%	2
Length:	NA	

Objectives

- Continue to develop a Java GUI application using more than one Java class
- Read and/or write from/to text files
- Use search algorithms and/or sort algorithms
- Use ArrayList or LinkedList or any other dynamic data structure

Assignment Details (Version 1.0 last updated on 17/03/2019)

Background

Friendly Student Accommodation (FSA) is a real estate agent that rents individual rooms and whole properties to students. They manage various types of rental properties, ranging from wooden houses to apartment units in city skyrises. You have been hired by FSA to develop a Java GUI application to help manage their business.

This proposed software will allow FSA staff to enter the details of newly acquired rental properties into the application. FSA staff will be able to manage these rental properties, such as changing their rental prices, the inclusions that come with these rental properties, and so on. The application will also create periodic backups of the properties stored within.

There are 2 categories of rentals that FSA manages.

1. Room Rental
 - a. These are rooms inside whole properties that are rented to individual students.
 - b. For example, a house has 5 rooms within and they can be rented to 5 different students separately.
2. Whole Rental
 - a. These are rental properties that are rented as a whole to a student.

Develop a Java GUI application using Swing components to implement the application.

You are allowed to continue work on this assignment by yourself or in pairs.

Requirements

We are moving onto Phase 2, which requires changes to be made to what we have now.

Phase 1 (Assignment 1) - Complete

1. Enter 1 rental property into the application
2. Edit and update the details of this rental property
3. Display a list of previously entered rental properties in the system (should only display the 1 entered earlier)
4. Exit the software safely

Phase 2 (Assignment 2)

1. Enter the more than 1 rental property into the application
2. Retrieve the details of rental properties previously entered
3. Edit and update the details of rental properties previously entered
4. Deleting rental properties from the application
5. Display a list of available rental properties in the system
6. Output the list of available rental properties into a file

Assignment 2 Tasks

For Assignment 2, you need to enter more than 1 rental property into the application.

1. Enter more than 1 rental property into the application

To be able to enter and store more than 1 rental property, you will need to use a dynamic data structure to store them. An **ArrayList** or **LinkedList** of **Rental** objects will work for this phase.

Back in Phase 1, you created an object (for Room or Whole property respectively) to store the details that the user entered into the application. What you should do for this phase is to take this same object and add it to your ArrayList or LinkedList.

When the user enters another Room or Whole property into your application, construct another and add it to the same list.

2. Retrieve the details of rental properties previously entered

This is similar to the previous task. In Phase 1, you extracted the stored details from the 1 object you had in the application.

For this phase, you will need to:

- Look into your list
- Find the required rental property object
- (Same as back in Phase 1) extract the stored details within and display on your GUI

There are several ways that you can design this into your GUI.

For all the following options, you will need to use the GUI interface you developed back in Phase 1 for editing the details of the 1 entered rental property.

GUI Option 1 (related to Task 3)

- Disable editing on all the fields.
- Add the following navigation buttons to the buttons to your screen
 - o [Next] (or >)
 - o [Previous] (or <)
- When this screen first loads, the details of the first Room or Whole property will be loaded into the fields.
- When the user clicks the [Next] button, the details of the next property in your list will be loaded into the fields.
- When the user clicks the [Previous] button, the details of the previous property in your list will be loaded into the fields.

Something to think about

This option is easy to use but it may take a long time to get to a specific rental property, especially if there are many rental properties

You can add [First] & [Last] buttons to speed this up

GUI Option 2 (related to Task 3)

- The same as GUI Option 1 but instead of using navigation buttons, use one of the following
 - o Use a **JComboBox** to select which rental property you want to view
 - The combo box should be populated with the RentalID of each entered rental property
 - This option is easy to navigate directly to the rental property you want to view
 - Might potentially be a problem if you have too many rental properties entered into the application
 - o Use a **JTextField**. Let users enter the RentalID of the rental property they want to view. Your application will then search the rental properties in the list and display the details if found
 - Easiest to implement, GUI-wise
 - Requires additional code for searching
 - Users might enter incorrect or non-existing RentalID
 - How will you manage this?

GUI Option 3

- Come up with your own design.
- Please remember to check with your tutor before attempting this option.

3. Edit and update the details of rental properties previously entered

This is basically an extension of Task 2.

If you went with GUI Option 1 or 2, you can add an [Edit] and [Save] button to the same GUI interface.

When users click the [Edit] button, the relevant GUI fields will have their editing functionality turned on. Users can then make changes and click [Save] to save the changes.

If you went with any other GUI design, you can still use this design above if it suits. If you are going to use any other design, remember to make it easy for the user to understand and use.

NOTE: Do not forget to perform input validation again before committing the changes permanently.

4. Deleting rental properties from the application

This requirement can be implemented similarly to Task 3.

You can add a [Delete] button to your view/editing screen to allow users to delete the current rental property from your application.

Deletion confirmation

Deleting data is usually considered a major action in most applications. Make sure you implement some form of confirmation by the user before permanently deleting the relevant data from the application.

A simple confirmation dialog box works well enough.

You can also consider other creative ways to verify the user's choice. For example, you can ask the user to re-enter the RentalID of the rental property and verify it before deletion.

5. Display a list of available rental properties in the system

This task requires an update to the same function we completed in Phase 1, except instead of displaying the details of 1 rental property, the application will now need to display the details of all the rental properties stored within.

There is no need to sort the output in any order at this stage.

Something to consider

What about the future? Will it make sense to add some sort of sorting to the list of properties?

6. Output the list of available rental properties into a file

Every so often, FSA needs to print out their list of available rental properties. To make things easier for this stage of the development, you were asked to output the list of rental properties in the application to a text file.

The file is to be named **FSA_Availability.csv**

Data structure of the output file (FSA_Availability.csv)

- Each line represents 1 rental property
- On each line, the individual bits of data are separated using a comma (,)
- This type of file is called a Comma-Separated Value (CSV) file

Example of how the stored data in this file might look like

```
R1251,Room 1 123 George Street. Brisbane City QLD 4000,180.00,true,A lovely furnished room in a city apartment,true,true
W3352,57 Canary Lane Ascot QLD 4007,350.50,false,A perfect little home for a family,4,2,true,true
W2251,124a Henson Drive Emerton NSW 2770,375.00,true,Home overlooking the sea,3,2,true,false
```

NOTE: One way to test if your file was correctly created is to open it using a spreadsheet application (such as MS Excel). If all the details above are nicely displayed within individual cells, then your file was created correctly. Check if your tutor if you are unsure.

To implement this function, for each rental property in your list

- Extract all the relevant details
- Combine them into a single String but insert a comma (,) between each bit of information
 - o You will need to remove any existing commas in your data to avoid corrupting your file
 - o Another way is to avoid saving any commas into your list in the first place
- Write the newly formed String into the file

File reading/writing was covered back in Week 11 in the pre-requisite course COIT11222 Programming Fundamentals. Additional material will be posted for your reference. Please check the Assignment 2 section on your course website regularly for updates.

7. Assignment Report (same as Assignment 1)

You will need to write a report that includes the following:

- Cover page that shows the unit code and name, assignment number, your name and student ID number
- **“Screen Shots”** section
 - o Includes screen shots of your running application.
 - o All relevant screens, error messages, and dialogs should be included.
 - o Each screenshot should have a description (caption) of the respective screenshot written below.
- **“Reflection”** section
 - o In this section, you need to write about the following (paragraphs or in a list)
 - o What difficulties did you encounter while working on this assignment?
Eg. Took too long to revise file reading/writing
 - o If you were able to complete all the required tasks
 - What do you think contributed to your success?
 - Is there anything you can do better in future?
 - o If you were not able to complete all the required tasks
 - Which sections were you not able to complete?
 - What reasons do you think prevented you from completing them?
 - How can you avoid these in reasons in future?
- **“Task Breakdown”** section (only needed if assignment was done in a pair)
 - o Use a table or a list to show how you distributed the work among yourselves
- Name this report “ReportAssign2.docx”

8. Submission

The assignment must be submitted in the following form:

- A zip file containing all the Java source code files. If done using NetBeans, a zip of the entire project folder.
- ReportAssign2.docx (a word file containing a cover page, screen shots showing various inputs, error messages and outputs from your application program when it is run).
- If assignment is completed as a pair, both students must submit the same files.

9. Assignment 2- Marking criteria

Code Quality	3
Meaningful variable/method names and constants using camel notation	0.5
Indentation and structured code for easy reading	0.5
Class & method headers	0.5
Decision and loops given appropriate comments	0.5
Comments in general	0.5
Use of methods to avoid redundant code	0.5
Functional Requirements	
1. Enter more than 1 rental property into the application	3.5
ArrayList or LinkedList created to store Rental objects	1.5
Successfully entered and saved new rental property into application (new object created for each new rental property entered)	2
2. Retrieve the details of rental properties previously entered	2.5
Suitable, easy to use GUI interface to retrieve and navigate through the previously entered rental properties	1
All rental properties can be displayed	1.5
3. Edit and update the details of rental properties previously entered	3
Suitable GUI interface to edit and update the selected rental property	1
Rental property successfully updated	1
Data is validated before saving	1
4. Deleting rental properties from the application	2
Rental property successfully deleted from the list	1
User confirmation before deletion works	0.5
GUI updated correctly after selected rental property deleted from list. (eg. entry removed from JComboBox)	0.5
5. Display a list of available rental properties in the system	1
All rental properties displayed	1
6. Output the list of available rental properties into a file	3
Output file was named "FSA_Availability.csv"	0.5
All rental properties saved into the file	1
Each line represents a single rental property	0.5
The data on each line is comma separated (valid CSV file)	0.5
File processing code has no issues (used try/catch correctly, file properly closed, etc)	0.5
7. Report	2
Cover page provided	0.25
Screen shots with all tests and error messages (deduct 0.5 if screenshots do not come with captions)	1
Reflection (difficulties, reflection on success, reflection for failure)	0.75
Task Breakdown (if pair work, deduct 1 mark if not included)	-
Sub-Total	20
Penalties (enter negative marks)	
Does not compile and run (deduct up to 2 marks)	
Part marks might be given upon code inspection	
Plagiarism	
Late submission: 5% per day (1 mark / day or part of a day)	
Total Marks	20