

**Further Programming**  
**COSC2391**  
**Assignment 1: Online Marketplace for Holiday Rentals**

<b>Assessment Type</b>	<b>Individual assignment;</b> no group work. Submit online via Canvas → Assignments → Assignment 1. Marks are awarded for meeting requirements as closely as possible according to assignment specifications and the supplied rubric. Clarifications or updates may be made via announcements.
<b>Due Date</b>	<b>Week 5, Sunday 3<sup>rd</sup> April 11:59pm.</b> Late submission penalty will be applied unless special consideration has been granted. <b>There will be one milestone, video demo check, in week 3, Friday 18<sup>th</sup> March 11:59pm.</b> You will describe and explain the key concepts adopted in your code and demonstrate code execution in the milestone.
<b>Marks</b>	15 marks out of 100 for assignment 1. 1 milestone of 4 marks in addition to the 15 marks.

### 1. Overview

**NOTE:** Carefully read this document. In addition, regularly follow the Canvas assignment discussion board for assignment related clarifications and discussion.

In this assignment you are required to work individually to implement a basic Java program, an online marketplace for lodging homestays for holiday rentals, named Melbnb. You will practice your knowledge of Java basics, object-oriented (OO) concepts, Java Collections Framework, exceptions, file input/output, and unit testing. You will use Java SE 8 or later.

### Task specification

The system keeps a list of properties where users can choose from. It allows the user to book a property for holiday rentals in Melbourne. The list of properties is provided in Melbnb.csv file (a csv file is a comma-separated values file). The first row in Melbnb.csv file contains headers, and the rest of the rows contain the property information by different hosts. The header row and an example record are shown below.

Property	Location	Description	Type of place	Host	Maximum no. of guests	Rating (out of 5)	Price per night (AUD)	Service fee per night (AUD)	Cleaning fee (AUD)	Weekly discount (%)
Single room next to Carlton Gardens	Carlton South	A nice private room with everything you need. Explore the city and enjoy gorgeous views from our lovely house.	Private room	Carmen	2	3.67	40	12	18	10

Cleaning fee is one-off. Weekly discount is only applied to the price per night if the period of the stay is equal to or over 7 nights. There is no discount for service fee per night and cleaning fee. For example, if a user books a property for 7 nights, the total payment with discount is calculated below:

$$(\text{Price per night} * \frac{100 - \text{Weekly discount}}{100} + \text{Service fee per night}) * 7 + \text{Cleaning fee}$$

The console-based program provides the following options:

- The user can search a property by location (e.g., Carlton South). The program should list all properties with the matching input. If the user types south or South, the menu should show all properties in the locations that contain south/South (i.e., Carlton South, Southbank, South Yarra). The user can then choose one from the list.
- The user can browse properties by type of place (e.g., Entire place). Once the type of place is selected, the user can choose one from the list of properties under that type.

- The user can filter properties by minimum rating. For example, if the user types 3, the menu should show all properties with the rating equal to or above 3. The user can then choose one from the list.
- Once a property is selected, the user will be asked to provide check-in and checkout dates.
- Once the dates are provided, the program will print the information of the selected property (it can be assumed a property is always available during the selected period). The information should include the name of the property, host, location, type of place, description, maximum number of guests, rating, total price without discount for the period of stay with price breakdown (i.e., price per night \* number of nights), total price with discount for the period of stay with price breakdown (i.e., discounted price per night \* number of nights) if discount is applied, total service fee with fee breakdown (i.e., service fee per night \* number of nights), cleaning fee, and total payment.
- The program will ask the user to reserve the property. If the user chooses not to proceed, the program will go to the main menu (i.e., top-level menu). If the user chooses to reserve, the program will ask the user to provide personal information. The personal information includes given name, surname, email address, and number of guests for the stay. Note that the number of guests provided should be smaller than the maximum number of guests.
- Once the user provides his/her personal information, the program will ask the user to confirm and pay. If the user chooses not to proceed, the program will go to the main menu (i.e., top-level menu). If the user chooses to pay, the program will print the reservation details and exit after printing the details (it can be assumed the payment is completed via an external payment portal). Reservation details include a confirmation message (e.g., "Your trip is booked. A receipt has been sent to your email. Your host will contact you before your trip. Enjoy your stay!"), name of the user, email address used for booking, name of the booked property, name of the host, number of guests coming, check-in and checkout dates, and total payment.
- The user can choose to exit the program. The program does not keep the previous incomplete reservation.

## 2. Assessment criteria

This assessment will determine your ability to implement Object-Oriented Java code according to the specifications shown below. In addition to functional correctness (i.e., getting your code to work) you will also be assessed on code quality. Specifically:

- You are required to demonstrate your understanding of basic object-oriented programming principles, including encapsulation and abstraction.
- You are required to use well-tested Java collections.
- You are required to validate all user inputs and catch and handle all exceptions.
- You are required to demonstrate that you have done adequate unit testing using the JUnit framework.
- You are required to read all system data from the given file and parse the information correctly.
- You are required to write properly documented code.

## 3. Learning Outcomes

This assessment is relevant to the following Learning Outcomes:

- CLO1: Explain the purpose of OO design and apply the following OO concepts in Java code: inheritance, polymorphism, abstract classes, interfaces, and generics.
- CLO2: Describe the Java Collections Framework (JCF) and apply this framework in Java code.
- CLO4: Demonstrate proficiency using an integrated development environment such as Eclipse for project management, coding and debugging.

## 4. Assessment details

In this assignment, you will incrementally build the basic Java program. The assignment is structured into 1 milestone with video demo in week 3 (4 marks in addition to the 15 marks for the assignment) and final submission in week 5.

### Part A

You are required to implement the functions mentioned in the task specification in Section 1 - Overview. You can start this part from week 1 and work on it until the due date.

- You will incorporate basic object-oriented concepts (e.g., abstraction and encapsulation). In particular, you will define several classes to support the OO implementation. You will also need to choose appropriate data

structures to store relevant information (e.g., a list of properties in Melbnb). You MUST use Java Collections because the list is growable.

- Your program should perform basic input validation (e.g., out-of-bound menu option, empty input, invalid check-in date/checkout date, invalid email address) and provide clear console messages for invalid inputs and further actions.
- The summary should be well-formatted (HINT: you can use System.out.printf method for formatting).
- You can hard code the property information at this point. You MUST use the information provided in the file, for testing purposes; please do not create new property information.
- You will need to document the code properly by following Code Conventions for the Java Programming Language by Oracle: <https://www.oracle.com/java/technologies/javase/codeconventions-introduction.html>

Following is a sample interaction with the online marketplace Melbnb. You do not have to follow this precisely, but it illustrates the required functionality. Text in **bold and green** would be input from the user:

Welcome to Melbnb!

-----  
> Select from main menu

- 1) Search by location  
2) Browse by type of place  
3) Filter by rating  
4) Exit

Please select: **1**

Please provide a location: **south**

-----  
> Select from matching list

- 1) Private room in the heart of Southbank  
2) Spacious bedroom in a cosy apartment in South Yarra  
3) Single room next to Carlton Gardens  
4) Go to main menu

Please select: **4**

-----  
> Select from main menu

- 1) Search by location  
2) Browse by type of place  
3) Filter by rating  
4) Exit

Please select: **2**

-----  
> Browse by type of place

- 1) Private room  
2) Entire place  
3) Shared room  
4) Go to main menu

Please select: **2**

-----  
> Select from entire place list

- 1) Studio close to Melbourne CBD  
2) 1-bedroom CBD view suite near Melbourne Central and RMIT  
3) Stylish two bedroom in CBD  
4) Sky high studio with amazing views  
5) Go to main menu

Please select: **5**  
-----

> Select from main menu

- 1) Search by location
- 2) Browse by type of place
- 3) Filter by rating
- 4) Exit

Please select: 3

Please provide the minimum rating: 4.2

> Select from matching list

- 1) Private room in the heart of Southbank
- 2) Spacious bedroom in a cosy apartment in South Yarra
- 3) Studio close to Melbourne CBD
- 4) 1-bedroom CBD view suite near Melbourne Central and RMIT
- 5) Sky high studio with amazing views
- 6) Budget accommodation bunk beds
- 7) Go to main menu

Please select: 1

> Provide dates

Please provide check-in date (dd/mm/yyyy): 12/12/2020

Please provide checkout date (dd/mm/yyyy): 30/12/2020

> Show property details

Property: Private room in the heart of Southbank hosted by Shelley

Type of place: Private room

Location: Southbank

Rating: 4.50

Description: The apartment is situated in the heart of Southbank with an easy access to shops and cafes. It has a warm and spacious living room with an amazing view of the gardens.

Number of guests: 1

Price: \$756.00 (\$42.00 \* 18 nights)

Discounted price: \$718.20 (\$39.90 \* 18 nights)

Service fee: \$180.00 (\$10.00 \* 18 nights)

Cleaning fee: \$11.00

Total: \$909.20

Would you like to reserve the property (Y/N)? Y

> Provide personal information

Please provide your given name: Mengmeng

Please provide your surname: Ge

Please provide your email address: mengmeng.ge@rmit.edu.au

Please provide number of guests: 1

Confirm and pay (Y/N): Y

> Congratulations! Your trip is booked. A receipt has been sent to your email.

Details of your trip are shown below.

Your host will contact you before your trip. Enjoy your stay!

Name: Mengmeng Ge

Email: mengmeng.ge@rmit.edu.au

Your stay: Private room in the heart of Southbank hosted

	by Shelley
Who's coming:	1 guest
Check-in date:	2020-12-12
Checkout date:	2020-12-30
Total payment:	\$909.20

## Part B

You can start this part from week 3 and work on it until the due date.

- You are required to write unit tests for all classes that include functions using JUnit framework. Your unit tests should have a good coverage of the typical use cases of your classes (e.g., calculating the total payment) and test all reasonable corner cases (e.g., handling invalid method arguments). You do not need to write unit tests for getters and setters.

## Part C

You can start this part from week 4 and work on it until the due date.

- You are required to read all information from the given file (HINT: you can use Scanner class and its relevant methods to read from the csv file and parse the information). For simplicity, you only need to extract property records from the file (i.e., 5 used as the maximum rating and 7 used as the minimum number of days for weekly discount can be hard coded).
- You are required to catch and handle all exceptions in your program. For example, if you use Scanner class to read the csv file, you will need to catch and handle the *FileNotFoundException* specified by Scanner class. Avoid throwing an unspecific Exception; throw a specific exception (e.g., throw a *NumberFormatException* instead of an *IllegalArgumentException*).

## 5. Submission

You will submit all the relevant material as listed below via Canvas.

- You must have Main.java file which includes the main method. This file serves as the entry point of your program.
- You must include a README with instructions on how to compile and run your program from the command line. You will be given a sample console program and a document showing instructions on how to compile and run the sample program from the command line on Canvas. You must give similar instructions (as shown in step 2 in the given document) in your README.
- You must submit a zip file of your project. Make sure you zip the top-level project folder.
- You must not submit compiled files (\*.class files) and any IDE related files (e.g., .settings folder generated by Eclipse). Please check your submission carefully, make sure all java files have been included.
- You can submit your assignment as many times as you would like prior to the due date. The latest submission will be graded.

After the due date, you will have 5 business days to submit your assignment as a late submission. Late submissions will incur a penalty of 10% of the full assignment 1 marks per day. After these five days, Canvas will be closed, and you will lose ALL assignment 1 marks.

## 6. Academic integrity and plagiarism (standard warning)

Academic integrity is about honest presentation of your academic work. It means acknowledging the work of others while developing your own insights, knowledge and ideas. You should take extreme care that you have:

- Acknowledged words, data, diagrams, models, frameworks and/or ideas of others you have quoted (i.e. directly copied), summarised, paraphrased, discussed or mentioned in your assessment through the appropriate referencing methods,
- Provided a reference list of the publication details so your reader can locate the source if necessary. This includes material taken from Internet sites.

If you do not acknowledge the sources of your material, you may be accused of plagiarism because you have passed off the work and ideas of another person without appropriate referencing, as if they were your own.

RMIT University treats plagiarism as a very serious offence constituting misconduct. Plagiarism covers a variety of inappropriate behaviours, including:

- Failure to properly document a source
- Copyright material from the internet or databases
- Collusion between students

For further information on our policies and procedures, please refer to <https://www.rmit.edu.au/students/student-essentials/rights-and-responsibilities/academic-integrity>

## 7. Marking Guidelines

- Functionality (7/15)
- Unit testing (2/15)
- Exception handling (1/15)
- Object oriented design and choice of data structures (2/15)
- Source code quality (2/15)
- Video demo in the final submission (1/15)
- Milestone check in week 3 (4/4)

**Functionality:** Menu selection and user input operations correctly implemented; basic input validation performed; correct calculation of payment; full property information and reservation details well-formatted and printed; all information from the given file parsed correctly; clear and meaningful messages after all operations and for all exceptions.

**Unit testing:** Unit tests with a good coverage of the typical use cases of classes and testing all reasonable corner cases.

**Exception handling:** All exceptions caught and handled; using specific exception classes.

**Object oriented design and choice of data structures:** Appropriate choice of data structures and class designs.

**Source code quality:** Adequately documented and properly indented code; appropriate class/method/variable names.

**Video demo in the final submission:** Clearly perform the demo of all functions implemented and unit tests. You must show your face while you demo. This is for the purpose of academic integrity. The demo must be within 4 minutes. You must use Microsoft Streams to create and share the demo. Instructions will be provided in a separate document.

**Milestone check:** Video demo in week 3; clearly describe class design (i.e., classes and significant methods); clearly explain choice of data structures in JCF; complete implementation of menu selection options and relevant input validation (Part A of the assignment specification) and demonstrate the code execution. You must show your face while you demo. This is for the purpose of academic integrity. The demo must be within 5 minutes. You must use Microsoft Streams to create and share the demo. Instructions will be provided in a separate document.