Faculty *of* Engineering & Technology

School *of* Computer Science and Mathematics

Coursework Title: **Room Booking System**

Module Name: **Introduction to Programming**

Module Code: **4100COMP**

Level: **4**

Credit Rating: **20**

Weighting: **60%**

Maximum Mark Available: **100**

Lecturer: **Dr. Bo Zhou**

Contact: *If you have any issues with this coursework you may contact your lecturer as listed below:*

Lecturer	**Course Taught**	**Room, Phone & eMail**		
Dr Denis Reilly	**Computer Studies**	731	0151 231 2279	d.reilly@ljmu.ac.uk
Dr Denis Reilly	**Computer Networks**	731	0151 231 2279	d.reilly@ljmu.ac.uk
Dr Gabor Kecskemeti	**Computer Science**	701	0151 231 2855	G.Kecskemeti@ljmu.ac.uk
Dr David Lamb	**Software Engineering**	711	0151 231 2636	d.j.lamb@ljmu.ac.uk
Dr Bo Zhou Dr Philip Kendrick	**Computer Security**	719 645	0151 231 2122 0151 231 8067	b.zhou@ljmu.ac.uk P.G.Kendrick@ljmu.ac.uk
Dr Sorren Hanvey Dr Dr Michael Mackay	**Computer Forensics**	645 721	0151 904 6160 015 1231 2276	S.C.Hanvey@ljmu.ac.uk M.I.Mackay@ljmu.ac.uk
Mr Andy Laws	**MultiMedia Computing**	604a	0151 231 2270	A.Laws@ljmu.ac.uk
Mr Andy Laws	**Data Science**	604a	0151 231 2270	A.Laws@ljmu.ac.uk

If there are any outstanding issues you may contact the module co-ordinator whose contact details are:

eMail: **b.zhou@ljmu.ac.uk**

Room: **719**

Hand-Out Date: **19th November 2020**

Hand-In Date: **16th December 2020 by 5PM**

Hand-In Method: **Canvas**

FeedBack Date: **15th January 2021**

FeedBack Method: **eMail**

Programme(s): **CS, CSc, SE, MC, CF, CSe, CN, DS**

Introduction:

This coursework is to be attempted ***individually***.

You are required to apply basic problem solving skills in the design *of* a computing solution. Once a suitable design has been produced, you will employ the programming skills taught throughout the module to develop robust programming code that utilizes appropriate data structures and storage. The coursework is supported by tutorial sessions up to the submission date.

Learning Outcome(s) Being Assessed:

1. *<not assessed in this coursework>*
2. *<not assessed in this coursework>*
3. Evaluate alternatives and make sound judgements regarding programming solutions.
4. Investigate integrated development environments & application programming interfaces.
5. Demonstrate basic knowledge *of* the object oriented programming paradigm.

Details *of* Task:

A new room booking system is required to help managers reserve a room based on the following requirements :

- Single, Double or Suite.
- Room Price.
- With or Without Balcony.
- With or Without Lounge.

When a suitable match is found, the guests (*identified by their eMail*) should be able to reserve the room. Room reservations may also be cancelled. The room data should be stored in a file ***rooms.txt*** that will contain the room data and any reservations, though initially every room should be unreserved. Upon application launch, the data should be loaded into appropriate data structures and upon application exit, the data should be saved back to the file.

*NOTE : The **rooms.txt** file (i.e. Appendix A) is provided on **Canvas** in the **Assignments** section.*
It contains seventeen sample rooms. The format of each room's data is as follows :

roomNum roomType roomPrice hasBalcony hasLounge eMail

You are thusly required to produce a design including an analysis *of* the specification, and UML (Unified Modelling Language) class diagrams. You should provide justifications for the design decisions you make.

You are then required to produce a console application (*using Java*) that is driven by a repeating main menu (*i.e. Appendix B*) with appropriate instructions and guidance throughout. Menu options should include reserving a room, cancelling a room and viewing room reservations.

Additional credit may be gained by implementing extra functionality :

- Performing Validation (*i.e. fallacious input should be rejected*).
- Before rejecting a reservation *(i.e. no rooms match the guest's room requirements)* the application offers the "next best match" (*i.e. we can match 2 of your 4 room requirements*).

What you should hand in:

1. A word processed report (*not exceeding 2 pages*) containing your design materials.
2. An electronic copy of all the properly formatted & commented source code (.java) files.

You should submit a **Single Zip File** which contains the **Word Processed Report** and the **Source Code Files**. Submit the zip file via the Assignment Handler in Canvas. Please check the document "How to create and submit a single zip file for assignment.pdf" on Canvas if you are unsure about how to do it.

Marking Scheme/Assessment Criteria:

Assessment	Assessment Criteria	% weighting *for* part
1	Design	10
2	Solution's Functionality • *Room Reserving, Cancelling & Viewing* 30 • *Data Structures & Storage* 20 • *Use of Methods* 10 • *Use of Objects* 10	70
3	Best Practice	10
4	Commenting	10

Guidelines:

- A best practice solution should employ robust OO programming code. In the event that you struggle with OO, you should still produce a design, but based around procedural programming in which you decompose the complete task into sub tasks. Your programming code should then be based on this design, which might use static methods in Java. Note that while a best practice solution is necessary for maximum marks, some marks are still available for partial solutions and you should attempt as much as you can, even if you do not get your programming code functioning fully.

- It is not necessary to develop your system outside the specification above, marks are available for answering the question, the whole question and nothing but the question, that said see below regarding best practice.

- The code should employ best practice (*i.e. indentation, spacing & camelCase*).
- The code should also be suitably commented (*i.e. non self-explanatory*).

Resources Required:

You may use the computing labs on the 6th & 7th floors *of* the Byrom Street Campus as well as the 1st floor *of* the Henry Cotton Campus.

You should make use *of* these specific tools & resources:

- Eclipse.
- Microsoft Visio 2013 / 2016.
- Lecture Materials.
- The Internet.
- Malik's Java Book.

Extenuating Circumstances:

If something serious happens that means that you will not be able to complete this assignment, you need to contact the module leader as soon as possible. There are a number *of* things that can be done to help, such as extensions, waivers and alternative assessments, but we can only arrange this if you tell us. To ensure that the system is not abused, you will need to provide some evidence *of* the problem.

More guidance is available at:

https://www.ljmu.ac.uk/about-us/public-information/student-regulations/guidance-policy-and-process

Any coursework submitted late without the prior agreement *of* the module leader will receive 0 marks.

Academic Misconduct:

The University defines Academic Misconduct as 'any case *of* deliberate, premeditated cheating, collusion, plagiarism or falsification *of* information, in an attempt to deceive and gain an unfair advantage in assessment'.

This includes attempting to gain marks as part *of* a team without making a contribution. The Faculty takes Academic Misconduct very seriously and any suspected cases will be investigated through the University's standard policy (*https://www.ljmu.ac.uk/about-us/public-information/student-regulations*).

If you are found guilty, you may be expelled from the University with no award.

It is your responsibility to ensure that you understand what constitutes Academic Misconduct and to ensure that you do not break the rules. If you are unclear about what is required, please ask.

For more information you are directed to following the University web pages:

- Information regarding academic misconduct:
  https://www.ljmu.ac.uk/about-us/public-information/student-regulations
- Information on study skills:
  https://www.ljmu.ac.uk/microsites/library/skills-ljmu

Appendix A

101 Single 23.50 false false free
102 Single 23.50 false false free
103 Double 27.50 false false free
104 Double 27.50 false false free
105 Double 27.50 false false free
201 Single 23.50 true false free
202 Single 23.50 true false free
203 Double 27.50 true false free
204 Double 27.50 true false free
205 Double 27.50 true false free
301 Single 33.50 true true free
302 Single 33.50 true true free
303 Suite 60.00 true true free

304 Suite 60.00 true true free
305 Suite 60.00 true true free
306 Double 37.50 true true free
307 Double 37.50 true true free

Note that in the original file above the field 'free' indicates that a room is not booked, an email would appear in this field when a room is booked. So for example, the stored data would appear as below after several rooms are booked.

101 Single 23.50 false false free
102 Single 23.50 false false fred.bloggs@ljmu.ac.uk
103 Double 27.50 false false free
104 Double 27.50 false false free
105 Double 27.50 false false free
201 Single 23.50 true false sue.white@gmail.com
202 Single 23.50 true false free
203 Double 27.50 true false free
204 Double 27.50 true false free
205 Double 27.50 true false free
301 Single 33.50 true true free
302 Single 33.50 true true free
303 Suite 60.00 true true free
304 Suite 60.00 true true free
305 Suite 60.00 true true sam.soap@persil.com
306 Double 37.50 true true free
307 Double 37.50 true true free

This data would be saved back to the rooms.txt file when the application exits.

Appendix B

- - Room Booking System - -

- - MAIN MENU - -
1 - Reserve Room
2 - Cancel Room
3 - View Room Reservations
Q - Quit
Pick :