

Coursework Title: **Seat 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	<b>Computer Studies</b>	731	0151 231 2279	d.reilly@ljmu.ac.uk
Dr Denis Reilly	<b>Computer Networks</b>	731	0151 231 2279	d.reilly@ljmu.ac.uk
Dr Gabor Kecskemeti	<b>Computer Science</b>	701	0151 231 2855	G.Kecskemeti@ljmu.ac.uk
Dr David Lamb	<b>Software Engineering</b>	711	0151 231 2636	d.j.lamb@ljmu.ac.uk
Dr Bo Zhou	<b>Computer Security</b>	719	0151 231 2122	b.zhou@ljmu.ac.uk
Dr Sorren Hanvey Dr Dr Michael Mackay	<b>Computer Forensics</b>	645	0151 904 6160	S.C.Hanvey@ljmu.ac.uk
	<b>MultiMedia Computing</b>			
	<b>Data Science</b>	721	0151 231 2276	M.I.Mackay@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: **18<sup>th</sup> November 2019**

Hand-In Date: **13<sup>th</sup> December 2019 by 5PM**

Hand-In Method: **Google Classroom**

FeedBack Date: **14<sup>th</sup> January 2020**

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 seat booking system is required to help users reserve a seat based on the following requirements :

- First or Standard Class.
- Window and/or Aisle.
- With or Without Table.
- Seat Price.

When a suitable match is found, the user (*identified by their eMail*) should be able to reserve the seat. Seat reservations may also be cancelled. The seat data should be modelled in a file **M:\data\seats.txt** that will contain the seat data and any reservations, though initially every seat 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 **seats.txt** file (i.e. Appendix A) is provided on **Google Classroom** in the **Assignments** section.

You should download the file to a new folder called **data** on your **M:** drive.

The format of each seats's data is as follows :

`seatNum seatClass isWindow isAisle isTable seatPrice eMail`

See Appendix A for further explanation of the file `seats.txt`

You are thusly required to produce a design including an analysis of the specification, 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 seat, cancelling a seat and viewing seat 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 seats match the guest's seat requirements*) the application offers the “next best match” (*i.e. we can match 3 of your 5 seat requirements*).

---

What you should hand in:

A word processed report (*not exceeding 2 pages*) containing your design materials.

The properly formatted & commented code in a ZIP file. Specifically, a single Eclipse project directory complete with all Java (`.java`) source code.

You should also submit any files required for your programming code's data storage (*i.e. the contents of your **M:\data\** directory*).

---

Marking Scheme/Assessment Criteria:

Assessment	Assessment Criteria	% weighting for part
1	Design	10
2	<div>Solution's Functionality<ul style="list-style-type: none"><li>• Seat Reserving, Cancelling &amp; Viewing30</li><li>• Data Structures &amp; Storage20</li><li>• Use of Methods10</li><li>• Use of Objects10</li></ul></div>	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. As a last resort, functionality could reside the “main” method / function only. 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 6<sup>th</sup> & 7<sup>th</sup> floors of the Byrom Street Campus as well as the 1<sup>st</sup> 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/appeals-and-complaints>).

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/appeals-and-complaints>
- Information on study skills:  
<https://www.ljmu.ac.uk/students/supporting-your-study>

---

#### Appendix A

1A STD true false false 23.50 free

1B STD false true false 23.50 free

1D STD true true false 27.50 free

2A STD true false true 24.50 free

2B STD false true true 24.50 free

2D STD true true true 28.50 free

3A STD true false true 24.50 free

3B STD false true true 24.50 free

3D STD true true true 28.50 free

4A STD true false false 23.50 free  
4B STD false true false 23.50 free  
4D STD true true false 27.50 free  
5A 1ST true true true 48.50 free  
5C 1ST false true true 44.50 free  
5D 1ST true false true 44.50 free  
6A 1ST true true true 48.50 free  
6C 1ST false true true 44.50 free  
6D 1ST true false true 44.50 free

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

1A STD true false false 23.50 free  
1B STD false true false 23.50 free  
1D STD true true false 27.50 free  
2A STD true false true 24.50 joe.bloggs@ljmu.ac.uk  
2B STD false true true 24.50 free  
2D STD true true true 28.50 free  
3A STD true false true 24.50 free  
3B STD false true true 24.50 free  
3D STD true true true 28.50 free  
4A STD true false false 23.50 sue.white@gmail.com  
4B STD false true false 23.50 free  
4D STD true true false 27.50 free  
5A 1ST true true true 48.50 free  
5C 1ST false true true 44.50 free  
5D 1ST true false true 44.50 free  
6A 1ST true true true 48.50 joe.soap@persil.com  
6C 1ST false true true 44.50 free  
6D 1ST true false true 44.50 free

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

---

## Appendix B

-- Seat Booking System --

-- MAIN MENU --

1 - Reserve Seat

2 - Cancel Seat

3 - View Seat Reservations

Q - Quit

Pick :