

# Title: Seat/Client Assignment- Class Project

2<sup>nd</sup> PASS – POC

## Table of Contents

Goal .....	2
Initial Core Features .....	2
Rationale .....	2
INPUTS .....	2
1) Internal List of Clients .....	2
2) Room Matrix .....	3
PROCESSING.....	3
OUTPUTS.....	3
1) Seating Matrix (a draft layout to explain).....	3
2) Clients List and their seating assignment (in console).....	4
Project Criteria .....	4
Project Folder.....	4
Hints and Advice .....	6

## Goal

To create an application which assigned clients to seats similar to a booking system for a movie theatre.

In this second pass, you will create a very basic scaffolding where we can associate clients loaded into a JList with a 10x10 matrix of seats which uses Swing GUI.

***To be delivered by July 29<sup>th</sup>, 2019***

## Initial Core Features

- contain a list of clients
- contain an internal matrix of a room which defines usable areas (where seating is possible)
  - Can book/unbook any given client to a seat
  - Can provide in the console, a report of the clients and their seating arrangement

## Rationale

### INPUTS

#### 1) Internal List of Clients

id	lastname	firstname
1	Mellor	Alexander
2	Schalk	Alexander
3	Storey	Alice
4	Montgomery	Andrew
5	Leighton	Angela
6	Torres	Brian Jesus Cedillo
7	Catzin	Chris
8	De Hoyos	Daniel
9	Loughran	David
10	Sobers	Deron F.
11	Hunter	Devlin Chase
12	Buchanan	Elbert
13	Silva	Issic
14	Guevara	Joel Reyes
15	Mills	John
16	Patiag	Jonathan
17	Chacko	Joseph
18	Williams	Lonnie
19	John	Marilyn
20	Kirk	Robert
21	Mao	Hannah Mao
22	Hand	Trey
23	Chmielewski	Victoria
24	Ubak	Yvonne
25	Gauthier	Claude

## 2) Room Matrix

Row/Column Array of objects which will contain the following info

- Initials of Client: "" ← starts as empty
- Client ID: "" ← starts as empty
- row/column reference (rows are letters A to ...) and columns are numbers 1 to ...

IF the Client ID and Initials of Client are both empty, the 'seat' is considered "vacant"

Note: obviously, you have a "Client" object and more than likely a "Seat" object.

Your seat object should be smart enough to know how to return "vacant" vs "id: xx/initials"

Your seat object should also be able to toggle a button between two states such as "book" and "unbook".

Finally, you will need to learn how to dynamically manage a JList to ensure that any client "booked" are removed from the JList, but when "unbook", that client returns to it.

### Room Matrix Size

10 rows, 10 columns

## PROCESSING

## OUTPUTS

### 1) Seating Matrix (a draft layout to explain)

"My 2 <sup>nd</sup> Matrix Demo"							
JList with JScrollPane of Clients		Seat 1	Seat 2	Seat 3	Seat 4	Seat 5	to Seat 10
	Row A	id: 25/CG "unbook"	vacant "book"	vacant "book"	vacant "book"	vacant "book"	vacant "book"
	Row B	vacant "book"	vacant "book"	vacant "book"	vacant "book"	vacant "book"	id: 01/AM "unbook"
	Row C	vacant "book"	vacant "book"	vacant "book"	vacant "book"	vacant "book"	vacant "book"
	To Row J	vacant "book"	vacant "book"	vacant "book"	vacant "book"	vacant "book"	vacant "book"
"reset entire matrix" and "print list of assigned clients in console"							

Behavior: Select a client, click the book button

It will assign the client, it will also remove the client from the JList

You cannot book another client in a seat, you can “unbook” a client and that will return the client to the JList and return the seat as “vacant” and “book”.

You will also add a “reset entire matrix” button, which will unbook all and ensure the JList is back with a full list of clients.

You will also add a “print list of assigned clients in console” button and it will generate a list as per bottom.

## 2) Clients List and their seating assignment (in console)

Client	ID	Seat
Claude Gauthier	01	A-1
...		

## Project Criteria

- Must be worked alone, no outside help, this is going to be work that will be evaluated for your MID and FINAL terms
- Demonstrate pragmatism, for example:
  - use an iterative approach in your development process, create versions/subversions of your app in stage
  - Use what you have mastered as a coder, keep it simple, when a version works, if you feel it could be better, then document and identify areas your code can be improved, perhaps with more advanced version of Java concepts and syntax and then, create a new version of your code
- By demonstrating pragmatism, you will deliver faster, albeit not always want you WISH you would have coded, but at least, it will work exactly as per minimum specifications. Your hiring manager can be made aware of your progress and help guide you towards your development, but only if you are able to document and clearly explain your thought process, your development strategy and the end goal you are trying to achieve.
- KEEP IT SIMPLE

***Note: those who do provide help, you may be risking your own career over this. Do not help. It is up to all employees who require help, to see the manager's attention. If you are caught, there will be consequences, again, beware.***

## Project Folder

**Folder: javatraining/preterm\_swing/ ← folder**

if you have any iteration, organize your code as per version, perhaps folder version naming convention for version of full projects such as version01, version02.. etc.. at the room of the preterm folder, have a **README.md** file which clearly indicates each version and the features you've added and any other comment which is relevant to the project's evolution. “missing features, bugs to fix, etc..”



## Hints and Advice

You will need to research on how to dynamically add/remove to a JList.

Basically, this is managing a JList dynamically in Java.

I would suggest you create mini projects to determine how best to do this.

You will use the web to find how to do this.

Many keywords can be used:

*“Dynamically manage a JList in Swing” or “java JList add/remove items dynamically”*

To repeat: use an iterative approach in your development process. Keep your “main” as clean as possible. Don’t try to solve it all. Instead, build on a logic of your objectives, which is what your main and all the functions you would create should do.

Keep it simple and get it solve!