Cinnamon Cinemas Movie Theatre Seating Challenge - The Brief

Cue the popcorn! It’s time for a movie! Welcome to the Cinnamon Cinemas Movie Theatre Seating Challenge!

Your Task

Setting the Scene

You have a special task from Cinnamon Cinemas to develop a program to allocate seats to customers purchasing tickets for a movie theatre.

Cinnamon Cinemas provided some criteria for how the program should be built.

Cinnamon Cinemas' Criteria

Well-designed and tested code representing real-world scenarios

Evidence of Test-Driven Development being applied

Clear communication and documentation

Preparing your Starter Project

Before you begin, ensure you have set up a console application project with a testing framework, so you are able to write code and run tests.

Cinnamon Cinemas Business Requirements

. The Cinnamon Cinemas Movie Theatre has 15 seats, arranged in 3 rows of 5

- Rows are assigned a letter from A to C
- Seats are assigned a number from 1 to 5

User Story

As a Cinnamon Cinemas Movie Theatre Manager

I want to allocate seats to customers

So that I can control reserved seating for the theatre

- **GIVEN** a customer wants to request some tickets
- **WHEN** they request a number of seats between 1 and 3 for a movie
- **THEN** the customer should be allocated the required number of seats from the available seats on the seating plan
- **AND** the seats should be recorded as allocated

Acceptance Criteria and Assumptions

- Write a program that allocates seats based on a random integer “number of seats” between 1 and 3

- Your program should allocate the required number of seats from the available seats starting from seat A1 and filling the auditorium from left to right, front to back
- All of the seats are available for sale when the program starts
- The program should continue to allocate a random number of seats until it finds there are not enough seats left to complete the request
- Once there are not enough seats available to be allocated then the program can halt

Your Solution

Feel free to implement an approach that you feel comfortable with to receive input into your program e.g. feeding input values into unit tests; input via a console application; supplying input via a file etc.

We would like you to apply Test-Driven Development (TDD) to test-drive your solution.

We would like to see production-quality code, this means you have thought carefully about your code design and that your code is clean and well-tested.

We'd love to see good unit test coverage and all unit tests passing.

Top Tips

- Sketch / plan out your ideas first, we recommend starting of by modelling what you need using the Unified Modelling Language (UML).
- Commit into your Github repository frequently and with descriptive commit messages.
- Write a descriptive README to document the key features of your solution, your assumptions, approaches and future thoughts.
- You must demonstrate a Test-Driven Development approach - RED-GREEN-REFACTOR. Each commit on your Github repository should contain unit tests that match up to the feature you are focusing on for each commit.

How do I make a submission?

You can choose either **Option 1** or **Option 2**:

Option 1 Please attach a link to your GitHub repository containing your final code solution. If you are working in a pair, please ensure you attach the link to the Github repository you are working from. If you are doing option 1, it doesn't have to be a complete solution. We expect to see two or three starter test scenarios to see how you might start off your solution using a TDD approach. If you wish to finish the whole solution, that's entirely up to you - we will provide you with feedback on what you submit.

OR

Option 2 Please upload a 10 minute screenshare recording in .mp4 format of yourself doing the challenge on your chosen Code Editor/IDE. You can record yourself on a single person zoom meeting for the video aspect. Feel free to finish off the challenge after your recording if you wish.

You must talk through your thinking process on the recording while you're coding; think of it like a live commentary of your reasoning on 'why' and 'how' you're going about solving the challenge. This is a common ask during tech interviews.

This brief was inspired by a real 1 hour live coding tech task for a digital ticketing platform.