Challenge

Implement Unit Testing Using Mocha and Chai

Challenge

- Challenge: Enable Testing in Ticket Booking Bill Calculator

Context

Creative IT Solutions frontend developer team has developed JavaScript solution to calculate total ticket booking amount for MovieZone cinemas.

However, they now want to enable automated unit testing to test the functions defined in the solution to ensure the code meets the stated requirements and is bug free.

Also, while planning test cases, few additional test cases are identified. This is done to ensure the code provides appropriate messages when the functions are provided with invalid inputs.

For these additional test cases, the developer team will have to refactor the code and provide error handling solutions that satisfy the test criteria.

Challenge: Enable Testing in Ticket Booking Bill Calculator

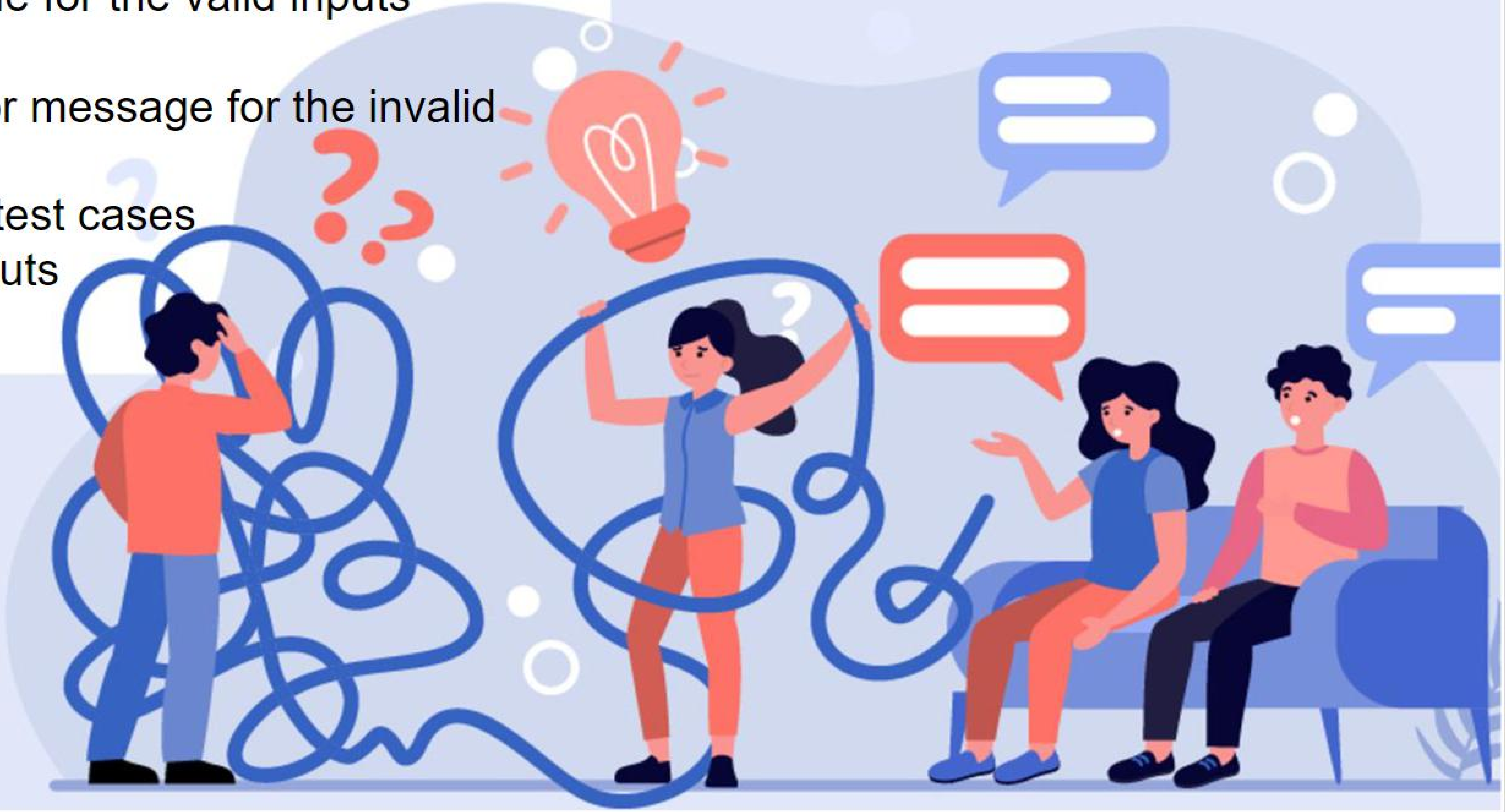
Write unit test cases to test the functions in the solution provided for calculating total bill amount.

The test cases should test whether

- the desired functions exist
- the functions return expected value for the valid inputs provided
- the functions return expected error message for the invalid inputs provided

Refactor the code to ensure that the test cases written to test functions for invalid inputs pass.

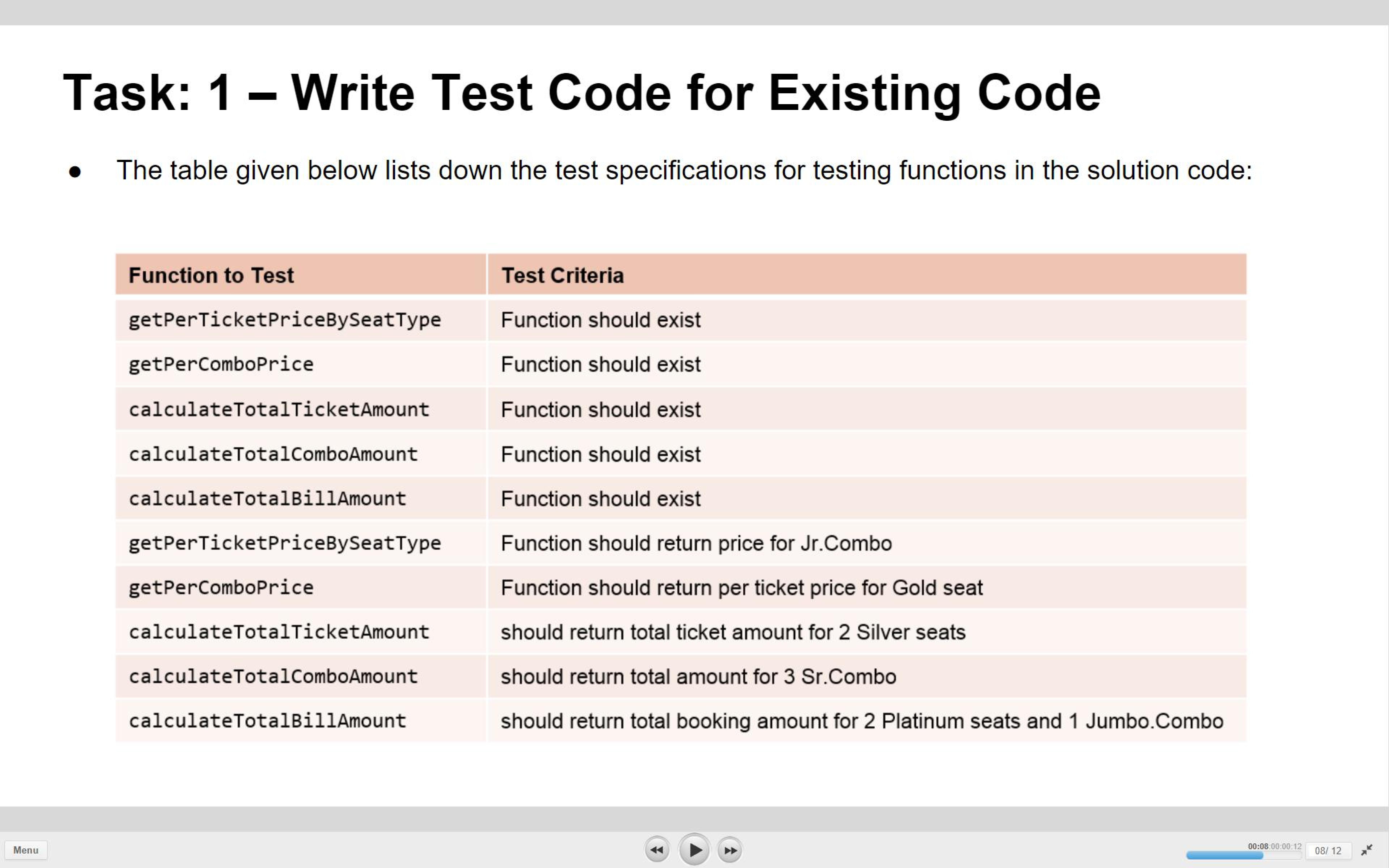
CHALLENGE



About Solution Code

- The table given below provides list of details of functions defined in the solution code:

Function Name	Objective	Valid Inputs
`getPerTicketPriceBySeatType`	Accepts 1 string value as seat type and returns ticket price for one seat based on the seat type value	Silver / "Gold" / "Platinum"
`getPerComboPrice`	Accepts 1 string value as combo type and returns price for one combo based on the combo type value	Jr.Combo / "Sr.Combo" / "Jumbo.Combo"
`calculateTotalTicketAmount`	Accepts 1 string value as seat type and 1 number value as ticket count and returns total ticket amount	1. Seat Type should be "Silver" / "Gold" / "Platinum", 2. Ticket Count should be Greater Than Zero
`calculateTotalComboAmount`	Accepts 1 string value as combo type and 1 number value as combo count and returns total combo amount	1. Combo Type should be "Jr.Combo" / "Sr.Combo" / "Jumbo.Combo", 2. Combo Count should be Greater Than Zero
`calculateTotalBillAmount`	Accepts 1 string value as seat type, 1 number value as ticket count, 1 string value as combo type and 1 number value as combo count and returns total bill amount	1. Seat Type should be "Silver" / "Gold" / "Platinum", 2. Ticket Count should be Greater Than Zero, 3. Combo Type should be "Jr.Combo" / "Sr.Combo" / "Jumbo.Combo", 4. Combo Count should be Greater Than Zero



Task: 1 – Write Test Code for Existing Code

- The table given below lists down the test specifications for testing functions in the solution code:

Function to Test	Test Criteria
`getPerTicketPriceBySeatType`	Function should exist
`getPerComboPrice`	Function should exist
`calculateTotalTicketAmount`	Function should exist
`calculateTotalComboAmount`	Function should exist
`calculateTotalBillAmount`	Function should exist
`getPerTicketPriceBySeatType`	Function should return price for Jr.Combo
`getPerComboPrice`	Function should return per ticket price for Gold seat
`calculateTotalTicketAmount`	should return total ticket amount for 2 Silver seats
`calculateTotalComboAmount`	should return total amount for 3 Sr.Combo
`calculateTotalBillAmount`	should return total booking amount for 2 Platinum seats and 1 Jumbo.Combo

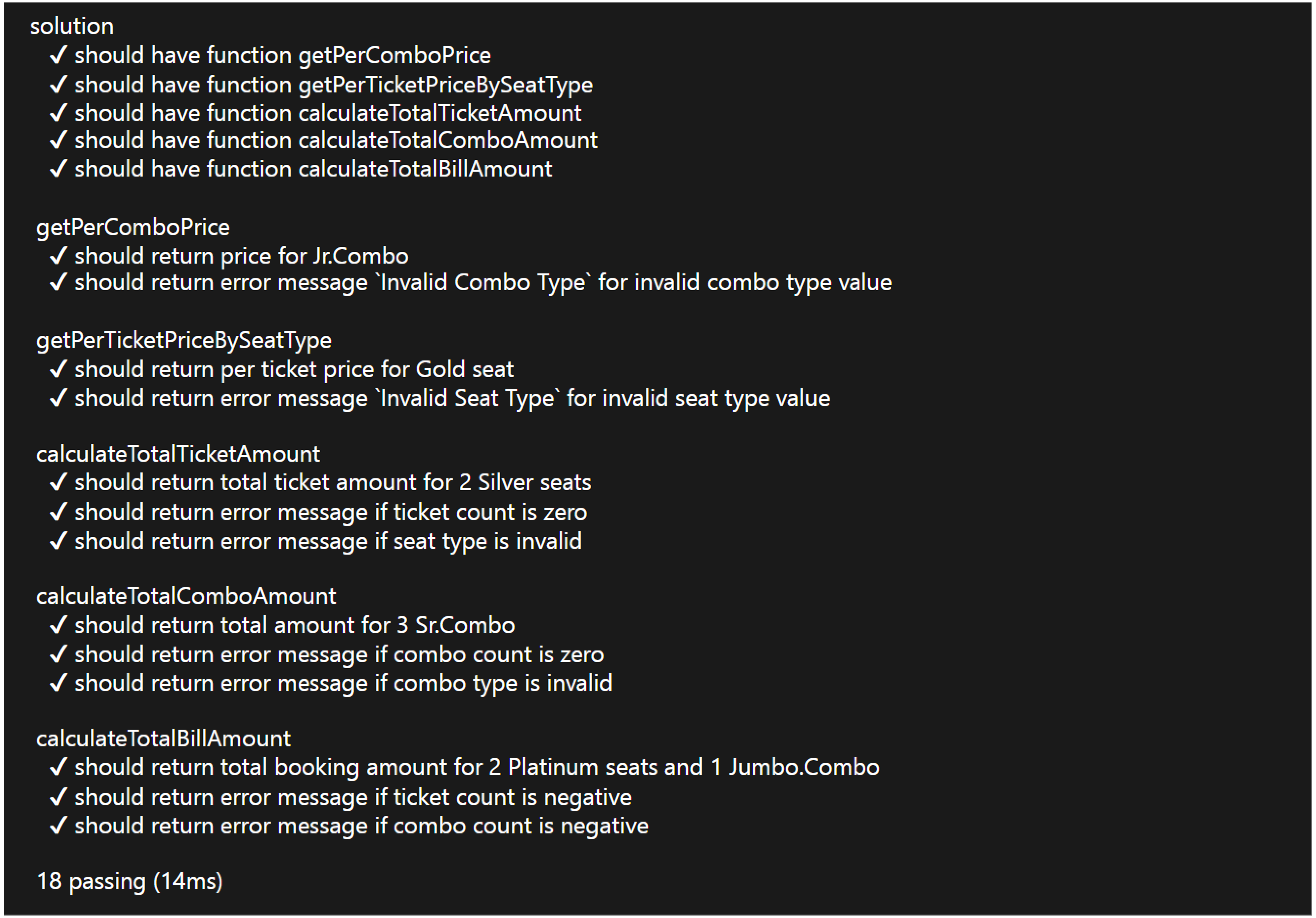
Task: 2 – Write Test Code and Refactor Code to Handle Errors

- The table given below lists down the specifications for testing these functions to ensure they return appropriate error messages when they are provided with invalid inputs:

Function to Test	Test Criteria
`getPerTicketPriceBySeatType`	Function should return error message `Invalid Seat Type` for invalid seat type value
`getPerComboPrice`	Function should return error message `Invalid Combo Type` for invalid combo type value
`calculateTotalTicketAmount`	Function should return error message if ticket count is zero
`calculateTotalTicketAmount`	Function should return error message if seat type is invalid
`calculateTotalComboAmount`	Function should return error message if combo count is zero
`calculateTotalComboAmount`	Function should return error message if combo type is invalid
`calculateTotalBillAmount`	Function should return error message if ticket count is negative
`calculateTotalBillAmount`	Function should return error message if combo count is negative

- Refactor the solution code to ensure the above listed test cases pass.

Expected Output



```
solution
  ✓ should have function getPerComboPrice
  ✓ should have function getPerTicketPriceBySeatType
  ✓ should have function calculateTotalTicketAmount
  ✓ should have function calculateTotalComboAmount
  ✓ should have function calculateTotalBillAmount

 getPerComboPrice
  ✓ should return price for Jr.Combo
  ✓ should return error message `Invalid Combo Type` for invalid combo type value

 getPerTicketPriceBySeatType
  ✓ should return per ticket price for Gold seat
  ✓ should return error message `Invalid Seat Type` for invalid seat type value

 calculateTotalTicketAmount
  ✓ should return total ticket amount for 2 Silver seats
  ✓ should return error message if ticket count is zero
  ✓ should return error message if seat type is invalid

 calculateTotalComboAmount
  ✓ should return total amount for 3 Sr.Combo
  ✓ should return error message if combo count is zero
  ✓ should return error message if combo type is invalid

 calculateTotalBillAmount
  ✓ should return total booking amount for 2 Platinum seats and 1 Jumbo.Combo
  ✓ should return error message if ticket count is negative
  ✓ should return error message if combo count is negative

 18 passing (14ms)
```