Programming Test



Table of Contents

[Version History 2](#_Toc166234391)

[C# 3](#_Toc166234392)

[React 4](#_Toc166234393)

# Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Comment | Author |
| 0.01 | 18/11/2020 |  | Mathew Parsell |
| 0.02 | 23/01/2024 |  | Kevin Toghai |
|  |  |  |  |

# C#

1. **Question 1:** Create a class `Product` with properties `Id`, `Name`, and `Price`. Implement a method `DisplayDetails()` that prints the details of the product.
2. **Question 2:** Create a class `ShoppingCart` that manages a list of `Product` objects. Implement methods to add, remove, and display all products in the cart.
3. **Question 3:** Implement error handling for adding products in the `ShoppingCart` class. If a product with the same ID already exists, throw a custom exception `DuplicateProductException`.
4. **Question 4:** Implement a method `GetTotalPrice()` in the `ShoppingCart` class that calculates and returns the total price of all products in the cart.
5. **Question 5:** Create a class `DiscountManager` with methods to apply discounts to products based on certain criteria, such as a percentage discount for products with a price above a certain threshold.
6. **Question 6:** Use LINQ to filter and display products in the cart based on different criteria, such as price range or name.
7. **Question 7:** Create a class `Order` to represent an order containing a list of `Product` objects and their quantities. Implement methods to add products to the order, calculate the total order price, and display order details.
8. **Question 8:** Implement serialization and deserialization methods in the `ShoppingCart` class to save and load the list of products to/from a file.
9. **Question 9:** Create a console application to demonstrate the functionality of the `ShoppingCart` and `Order` classes. Allow the user to perform operations like adding products to the cart, applying discounts, creating an order, and displaying order details.
10. **Question 10 (Bonus):** Implement a feature to apply different types of discounts (e.g., percentage discount, fixed amount discount) to products in the `ShoppingCart` class.

# React

The following uses react to create an application for managing an event calendar. The application should be done within a single page and require no browser refreshing to function. The purpose of this is to gauge the comfort level using react as a front end as well as how you structure applications that you build.

You must use React and FullCalendar

The allotted time is 2 hours at the conclusion of which please zip and upload your project files to a file share and send a link through to your contact. We take no ownership of your project and you are free to use this as part of your portfolio if you wish.

React: <https://reactjs.org/>

FullCalendar: <https://fullcalendar.io/docs/react>

1. Create a form for creating calendar events. The minimum needed would include a field for title, description and datetime.
2. Using the React FullCalendar plugin render the calendar with the events entered through the form. Include the date and title as the text of the event.
3. When hovering over an event within the calendar show a tool tip that includes description that was entered for the event.
4. When an event is clicked within the calendar it should be deleted.
5. Create a search field that filters the events shown in the calendar to those that contain the entered text in either their title or description.