Picture 'G' – FAST FOOD RESTAURANT

The code we wrote simulates the behavior of ordering food at a fast-food restaurant. we primarily have several classes (“Meal”, “Side”, and “Drink”) and a main program. Each class represents a different aspect of a fast-food restaurant, while the main program handles user input and displays the program's output.

Structure of the Classes

“Meal” Class: Represents main dishes, such as burgers or sandwiches.
It contains the following main components:
- Attributes: Including the dish name, price, and the special sauce chosen by the user.
- Methods: Including a constructor, “chooseSauce” (allows users to choose a sauce for their dish), and “describe” (provides a description of the dish, including name, sauce, and price).

“Side” Class: Represents side dishes, like fries or onion rings.
Its structure is like the “Meal” class, but it adds a boolean attribute (“spicy”) indicating whether the dish is spicy, and an “askSpicy” method that allows users to choose if they want their side dish to be spicy.

“Drink” Class: Represents optional beverages offered to customers, such as coke, coffee, and water.
It has fewer attributes, mainly including the drink's name and price, along with a “describe” method to output a description of the drink.

Structure of the “main” Function

The main program is responsible for displaying menu options, processing user choices, and calculating and displaying the total price based on the user's selections. The specific steps are as follows:

Displaying Menu Options: Use the “showMealOptions”, “showSideOptions”, and “showDrinkOptions” functions to display the options for main dishes, side dishes, and drinks, respectively.

Processing User Choices:
- The user first chooses from the main dish options, then can select a special sauce for it.
- Next, the user chooses from the side dish options
- Then, deciding if they want it to be spicy and can select a special sauce for it as well.
- Finally, the user selects from the drink options.

Additionally, we have added “None” options to each class option to allow the user to skip that option.

Calculating and Displaying Total Price: Calculate the total price based on the user's selections and display the description of chosen dishes along with the total charge.

The code also includes input validation to ensure that the user input is valid and prompts the user to retry if the input is invalid. Overall, our code implements object-oriented programming by encapsulating data and methods that operate on the data into classes. With this structure, the code not only demonstrates how classes and objects can be used to simulate real-world scenarios, but also shows how scenarios can be interactively and dynamically modified through user input.