Advanced Programming Round

Duration : 3 hours

General Instructions

- Understand the question clearly before writing the program. Ask the invigilator for clarification if needed.
- Use a logical approach and coding standards for evaluation.
- Test the program with required sample inputs.
- Choose Java, C, C++, or C# for the program.
- You can use any standard IDE/Compiler/Interpreter or online compiler.

Food Delivery Application

Objective:

Develop a command-line Food Delivery Application in either Java or C++ that manages restaurants, menus, delivery agents, and customer orders, including calculating delivery distances and assigning agents based on proximity.

Requirements Overview:

1. **Restaurant Inputs:**
   - **Restaurants:** Each with a unique ID, name, and location using Cartesian coordinates (x, y).
   - **Menu with Price:** Each restaurant has a menu with multiple items, each with a unique ID, name, and price.
2. **Delivery Inputs:**
   - **Delivery Agents:** Each agent has a unique ID, name, current location (x, y), and status (available or busy).
   - **Delivery Cost:** Calculated at a rate of 10 INR per km, using Cartesian distance.

3. *Core Use Cases:*
   - **Populate Static Data:** Preload the application with a list of restaurants with menus and delivery agents with their locations and their initial status.
4. **Signup/Login:**
   - Implement a basic signup and login system for Admin, Customers, and Delivery agents. Use simple text-based input to store and verify user credentials.
   - Get the current location of the customer while signing up and use that location as a delivery location.
5. **List Restaurants and Menu:**
   - Display a list of restaurants to the customer.
   - Allow customers to select a restaurant and view its menu.
6. **Add Items to Cart and Proceed to Booking:**
   - Allow customers to add menu items to a cart.
   - Display the total price of items in the cart and proceed to check out
7. **Generate Bill with Delivery Cost:**
   - Calculate the total bill, including the delivery cost, based on the distance between the restaurant and the customer's location using the Cartesian distance formula : {Distance} = $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$
   - Display the total cost to the customer before confirming the booking.
8. **Assign Delivery Agent Based on Proximity:**
   - Calculate the nearest available delivery agent to the restaurant using their last drop location.
   - Assign the agent to the order and update their status to busy
9. **Block Booking When Delivery Agents Are Full**
   - If no delivery agents are available, block new bookings and inform the customer.
10. **Delivery Agent Status Reset:**
   - Allow the Delivery agent to login and reset his status (e.g., when an order is completed update the status).
11. **List Restaurants Based on Proximity**
   - List restaurants for customers based on proximity to their location (nearest first).
12. **Admin Functionality:**
   - Admin users can add or modify static data (restaurants, menu items, delivery agents).
   - Admin and customers can view all or their own orders and order status.

Sample Static Data

Restaurant	Location	Menu	Price
A2B	(1,10)	Idly	40 INR
		Dosa	80 INR
		Poori	70 INR
Sangeetha	(-10,5)	Mini Idly	30 INR
		Pongal	50 INR
		Vada	20 INR
Buhari	(-6,-10)	Fired Rice	150 INR
		Biriyani	300 INR
		Noodles	180 INR

Delivery Agent	Location	Status
Kumar	(5,6)	Available
Praveen	(1,-8)	Busy
Sathish	(-5,-5)	Available

Graphical Representation