The University of Lahore, CS & IT Department

Object Oriented Programming

CampusBites Project

Start Date: 09/12/2024 **Total Marks: 125**

Due Date: 05/01/2025 **Program:** BSCS

Instructions

1. Understanding the problems is part of the project. So, no query, please.
2. You will get zero marks if found any type of plagiarism or AI generated code.
3. No submission after due date.
4. Submit a zip file containing your project files, including all .cpp files, header files, and any CSV or TXT files used.
5. There must not be any Syntax Error in Code. If your code fail to execute during evaluation you will automatically lose 50 % marks.

CampusBites is a busy cafeteria located on a college campus. With a diverse menu, including various cuisines and dietary options, the cafeteria caters to hundreds of students, faculty, and staff members daily. The cafeteria management aims to improve the efficiency of order processing and enhance customer satisfaction, leading to increased revenue and a positive dining experience.

1. Functionalities:

Menu Management: The cafeteria manager, Alice, uses the system to add, modify, and display menu items. She categorizes menu items based on types (e.g., vegetarian, vegan, gluten-free) for easy access. The system allows her to adjust prices, update item availability, and track sales data for each menu item.

Customer Registration & Loyalty Program: Whenever a new customer arrives, the cafeteria staff registers them using the system. The customer provides their name, contact details, and dietary preferences such as vegetarian, vegan, or gluten-free. The system assigns them a unique customer ID and records their order history. Frequent customers can enroll in a loyalty program, earning loyalty points based on their total spending, for example, 1 point for every $50 spent. These points can be accumulated and automatically applied as discounts on future orders, encouraging customers to return and continue their patronage.

Order Placement & Customization: When a customer approaches the counter to place an order, the cafeteria staff, Bob, uses the system to create a new order. Bob can search the menu by categories like "Vegetarian" or "Gluten-Free" or use keywords to find items quickly. He adds items to the order, allows customizations (e.g., toppings, sides), and manages quantities.

Order Processing & Payment: After finalizing the order, Bob confirms it, and the system calculates the total price, applying any applicable discounts from loyalty points. The system tracks payment status and generates an order receipt for the customer. Bob marks the order as completed after receiving payment.

Inventory Management: The system keeps track of inventory levels for each menu item. When an order is placed, the system automatically updates the stock count, notifying the staff when certain items are running low.

Sales Reporting & Analysis: At the end of each day, Alice generates a detailed sales report. The report includes information on total sales, the number of orders processed, revenue earned, and the most popular menu items. She can also generate monthly reports to track trends and identify areas for improvement.

Customer Service & Personalization: The system enables staff members to access customer details, including their order history and preferences. This information allows them to provide personalized recommendations and offer a better dining experience.

Conclusion:

The Cafeteria Order Management System (COMS) significantly improves the efficiency and management of CampusBites cafeteria. Utilizing Object-Oriented Programming (OOP) principles in C++, the system provides a modular and maintainable codebase. With COMS in place, Alice and Bob can focus more on providing quality service to their customers, leading to increased customer satisfaction, enhanced revenue, and better cafeteria management. The system's loyalty program and personalized services further strengthen the cafeteria's relationship with its patrons, fostering a positive dining environment on the college campus.

Sample CampusBites Cafeteria Menu

Vegetarian:

1. Veggie Burger - $8.99 (Available: 25)
2. Veggie Wrap - $7.49 (Available: 20)
3. Grilled Cheese Sandwich - $5.99 (Available: 30)
4. Vegetable Stir Fry - $9.49 (Available: 18)

Vegan:

5. Vegan Buddha Bowl - $10.99 (Available: 15)
6. Vegan Tacos - $8.49 (Available: 20)
7. Vegan Quinoa Salad - $9.29 (Available: 12)
8. Vegan Chili - $7.99 (Available: 25)

Gluten-Free:

9. Gluten-Free Margherita Pizza - $11.49 (Available: 10)
10. Grilled Chicken Salad (Gluten-Free) - $10.99 (Available: 18)
11. Gluten-Free Pancakes - $6.99 (Available: 15)
12. Gluten-Free Pasta Primavera - $12.49 (Available: 14)

Specialty:

13. Classic Cheeseburger - $9.99 (Available: 20)
14. BBQ Pulled Pork Sandwich - $9.49 (Available: 15)
15. Chicken Caesar Wrap - $8.99 (Available: 25)
16. Grilled Salmon Plate - $13.49 (Available: 10)

Drinks:

17. Iced Latte - $4.29 (Available: 30)
18. Fresh Fruit Smoothie - $5.99 (Available: 20)
19. Cappuccino - $3.49 (Available: 25)
20. Lemonade - $2.99 (Available: 35)

Rubrics for marking

1. **OOP Principles (25 points):**
   - Encapsulation (5 points): Assessing the use of access modifiers (private, protected, public) to encapsulate data and methods within classes.
   - Inheritance (5 points): Evaluating the use of inheritance to establish a relationship between classes and promote code reusability.
   - Polymorphism (5 points): Checking if polymorphism (e.g., function overloading, virtual functions) is appropriately employed to handle different data types and behaviors.
   - Abstraction (5 points): Reviewing how classes and functions abstract the essential characteristics of objects and operations.
2. **Menu Management (20 points):**
   - Adding/Updating Menu Items (5 points): Assessing the ability to add and update menu items with their corresponding prices and availability status.
   - Categorization (5 points): Evaluating the implementation of menu item categorization for easy access and user-friendliness.
   - Displaying Menu (5 points): Checking the display of the menu, ensuring it presents items and their details clearly.
   - Removing Items (5 points): Reviewing the functionality to remove items from the menu when they are discontinued.
3. **Customer Management (20 points):**
   - Customer Registration (5 points): Assessing the registration process for new customers, including the assignment of unique customer IDs.
   - Customer Information Update (5 points): Checking the ability to update customer details, such as contact information and dietary preferences.
   - Loyalty Program (5 points): Evaluating the loyalty program implementation and proper application of discounts based on loyalty points.
   - Order History (5 points): Reviewing the tracking of customer order history for personalized services.
4. **Order Processing (20 points):**
   - Order Placement & Customization (5 points): Assessing the functionality to place orders, add items, and apply customizations (e.g., toppings, sides).
   - Order Calculation (5 points): Checking the accurate calculation of total order price, considering prices and applicable discounts.
   - Inventory Management (5 points): Evaluating the system's ability to track inventory levels and update stock counts after order placement.
   - Order Status & Payment (5 points): Reviewing the order status handling and payment tracking.
5. **Sales Reporting & Analysis using File Handling (15 points):**
   - Daily Sales Report (5 points): Assessing the generation of daily sales reports, including information on total sales, orders processed, and revenue earned.
   - Monthly Sales Report (5 points): Evaluating the generation of monthly sales reports to track trends and identify areas for improvement.

- Popular Menu Items (5 points): Checking the identification of the most popular menu items based on sales data.

6. **User Interface & Experience (25 points):**
   - Ease of Use (5points): Evaluating the system's user-friendliness and ease of navigation through the console interface.
   - Error Handling (5 points): Assessing how the system handles user inputs and provides appropriate error messages.
   - User Inputs (15 points): System must take input from the user. No Hardcode Data.