Object Oriented Programming

Project

Name: P V S TARAK SHREE VALLABHA

BITS ID: 2020A7PS1513P

Topic: Restaurant Management (Smart Kitchen)

Project Number: 16

# Class Descriptions

1. public class MenuItem

MenuItem class is a concrete class that is used to create menu item objects that are stored in the menu. It has name, description, price id and category attributes.

It has the getter methods to get description, name and price of the menu item. It has a setter method to set price of the menu item. It has a method void toString() which overrides the toString() method of the Object class.

1. public class OrderItem extends MenuItem

OrderItem class is a class which is used to create objects of type OrderItem which are the items ordered by the customers. It extends the MenuItem class. Since it inherits the MenuItem class, it inherits all the non-private fields and methods of the MenuItem class. In addition to them it has itemQuantity and cookState attributes.

It has getter methods for cookstate , itemQuantity and price and setter method for cookState.

It has the isServed() method which returns if the orderItem has been served. Else, it returns false.

It also has the toString() metohod which overrides the toString() method of the MenuItem class.

1. public class Menu

It is a composition of four static array lists of type MenuItem: 1) starters 2)mainCourse 3)drinks 4)desserts. These lists represent the categories of the Menu.

It has the following functions:

* void readMenu(String, ArrayList<MenuItem>, boolean) : This function is used to read the menu from the file and save it to the array list passed. The boolean value passed decides if the menu should be printed while reading. It is public and static.
* void printMenu(): This function reads the menu using the readMenu function and prints the menu on the console.This method is public and static.
* void addItem(String, double, int, String ): This function is protected and static.. Only the Admin class has access to this function. The admin can call this function by passing values to add a new MenuItem object to one of the categories of the menu. This function also uses the updateFile(String, ArrayList<MenuItem>) method to update the files where the menu is stored.
* MenuItem getItem(int): This method is public and static. It returns the MenuItem corresponding to its id.
* void ChangePrice(int, double): This method is protected and static. This method can be accessed only by the admin class which calls this method to change the price of an existing item. The method then updates the files that store the menu.
* void updateFile(String, ArrayList<MenuItem>): This method is public and static. It takes the name of the menu category and the menu list and updates the menu files.

1. public class Order extends Thread

This class represents an order at the Restaurant. An object of this class is created when one orders something at the restaurant. It is a composition of OrderItem objects that are categorized into four different array lists. Each order also has an order number and total price. It has the following methods:

* Order(int): This is the constructor which takes the order no as argument and creates and object of this class. This method is public.
* void addItem(OrderItem): This method is public. It is used to add items to the order object. This method is public.
* double getOrderPrice(): This methods returns the price of the order.
* boolean isServed(String): This method takes the category as input and checks if the corresponding category in the order has been served. This function is used in the multithreading to maintain the ordering of orders. This method is public.
* boolean isServed(ArrayList<OrderItem>): This method checks if all the items in the given array list have been served. This method is also used in multithreading for maintaining the order of Order threads. This method is public.
* void printBill(): This method is used to print the order bill. It is a public method.
* void printState(): This method is used to print the cook state of items in the order.
* void run(): This method overrides the run method of the Thread class. It is the where each order thread starts and ends. It denotes the life of the order thread.

1. public class Kitchen

This class has four synchronized public static methods:

* void starterUnit(OrderItem, int):
* void mainCourseUnit(OrderItem, int):
* void dessertUnit(OrderItem, int):
* void drinkUnit(OrderItem, int):

These methods are synchronized and ensure that only one order thread enters them at once. They then work on the OrderItem and print the orders served in a separate text file “Servings.txt”.

1. public class Admin extends Menu

The Admin class has the functionality of an admin. It has three public static methods:

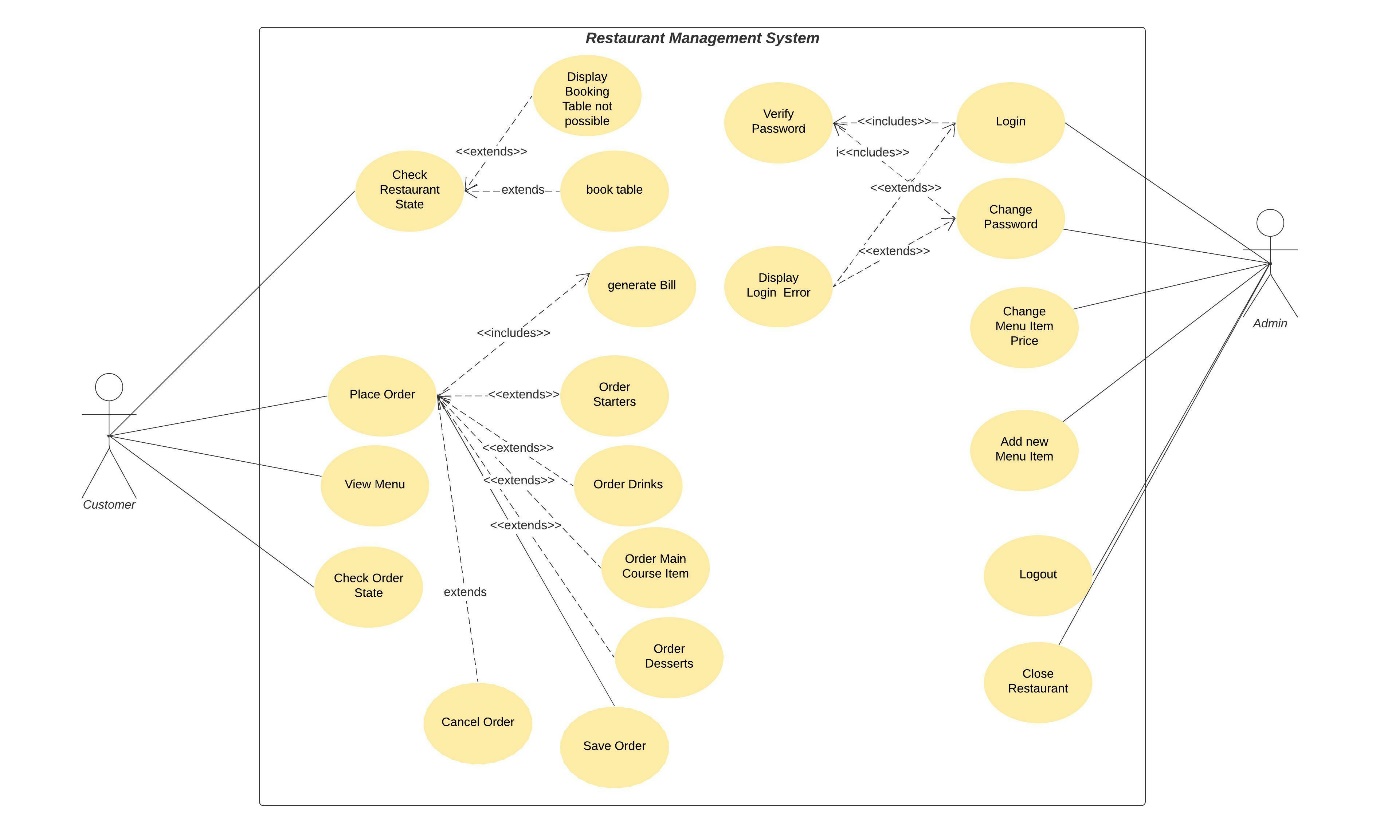
* boolean verifyPassword(String , String): It takes username and password and verifies with the existing username and password from the file base. The username is Username and the default password is Password.
* changePassword(String, String, String): It take the old username , password and the new password, verifies old username and password and sets new password by updating it in the file base.
* adminMenu(): This menu is available only for admins which the admin can use to : 1) Add Menu Item. 2)Change Price of Menu Item. 3)Change Password. 4)View Revenue. 5)Logout. 6)Close Restaurant.

1. public class Restaurant

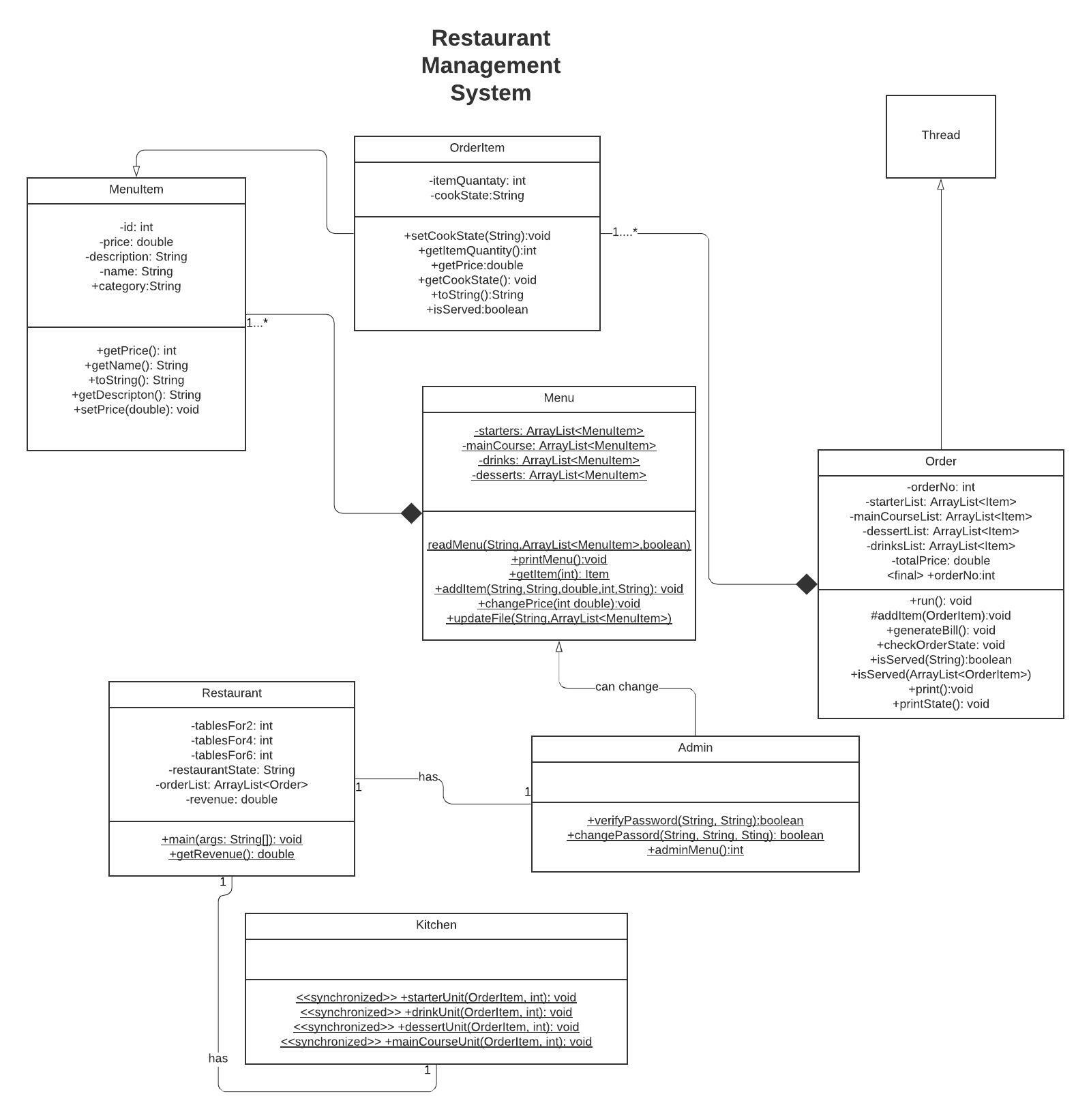
This class denotes the Restaurant. It stores the number of tables available in the restaurant, the restaurant state order list, revenue using static variables. The initial value of revenue is read from the file and updated after each order to the file “Revenue.txt”. It has the following static methods:

* boolean bookTable(): This method is private and is used to book a table.This method return a boolean value depending on whether the booking took place successfully.
* void freeTable(): This method is private and is used to free a table.
* void getRevenue(): This method reads the revenue from the file and prints it on the console.
* void updateRevenue(double): This public method update revenue after every order.
* setRestaurantState(String): This public method is used to set Restaurant state.
* void main(String[]): This method is the main method of the program. This is where the execution of the program starts. This denotes the main thread. Its functionality is displayed through the UML use case diagram.

UML Use Case Diagram



UML Class Diagram



# Analysis of Object-Oriented Programming Principles and Design Patterns

In the project that has been implemented there are four different categories of Menu Items that have different amount of cook time. So, the method used in the program was to create separate Array Lists for each category and store the different items in a menu and decide the cook time depending upon the list from which the item was taken from. For order items also the same thing was done. This kind of solution has some problems. It allows only four categories to exist. Some time there may be some problem in the kitchen, so we may want to change the cook time of only that particular item during runtime. But this design does not allow that. We may also want to add some items that are readily available during a particular and require no cooking or requires more time. But all the other items in the list will get the same properties. So, we will have to create a new category for that one dish.

Summarising the above paragraph, there will be two major problems:

1. We cannot make changes to the cooking time at run time.
2. To change the cooking time of one item, we need to change the cooking time of all other items, i.e., the items in a category are tightly coupled.
3. To add an item to which falls into one of the categories but has a different cooking time, we will have to create a new category altogether.

So, it is necessary to check whether our code is following the OOP principles and make modifications to our code:

1. Encapsulating what varies: The cook time varies for the order items. We have made different methods for cooking different categories and based upon the list, the order will go into that method. So, there is some level of encapsulation but there are still parts of the code that are repeated in the method that are similar for all the lists. They should have been encapsulated separately from cook time.
2. Favouring composition over inheritance: In our program, we have OrderItem class extending MenuItem class. This makes MenuItem class tightly coupled to OrderItem class. A better solution would be to have object of MenuItem class in the OrderItem class so that we can make use of the properties of MenuItem object without affecting it. This enables loose coupling.
3. Strive for Loose coupling between the objects that interact: The order item objects in the same menu category are stored in a common list and have the same implementation. So, there is tight coupling between them. If we want to change anything for one of the objects, it affects all the other objects in the list. So, all the OrderItem objects in a category list are tightly coupled.
4. Classes should be open for extension and closed for modification: The protected members of the Menu class can be modified by the Admin class and the category field of the MenuItem class can be modified from anywhere. There are getter and setter methods for the required private fields in classes. So, the classes are open for extension. But all the classes are not completely closed for modification.
5. Program to an Interface not an implementation: The Kitchen is a concrete class that contains different implementations of the cooking process in the form of methods. The program directly uses those methods to cook items. And the item undergoes the cooking process according to the category. We cannot change the cooking process of an object at runtime. We also cannot change the cooking process of one OrderItem object without affecting the other objects in the same category list.

*USING STRATEGY PATTERN:*

In our basic code there are four different types of order items that have different cooking times. There are data members that are common to all items. So, we encapsulate what differs from what remains same. We use an abstract class Kitchen that has methods common to the cooking process and make an abstract method cook which can be implemented by concrete classes that represent different kinds of kitchen units that extend the abstract class Kitchen. Then in each order item we store an object of the concrete classes that define cooking with reference to the variables of the supertype Kitchen. We can change the object assigned at runtime to change the cooking behaviour. And we use the cooking behaviour of the object stored. So, the code is reused since items by default get the implementation according to their category and if there are any exceptions, we can always change the object assigned to the supertype thus changing the cooking behaviour. Since the objects stored in one item can be independent of the other item, the items will be loosely coupled. In this way using strategy pattern improves our code.

# Improvements that can be done in the code

* The code does not have a feature to change the username which can be added.
* The Multithreading is not the most optimal. The multithreading design can be improved.
* The code can be optimised using design pattern and Object-Oriented Programming Principles as explained earlier.

# Software Used in Addition to Oracle

# The Console Input and output obtained (explained in the video)

Here I have pasted the copy of the text of the console input and output that was explained in the video

Welcome To the restaurant Smart Kitchen!

The Restaurant is OPEN.

Please choose one of the following options to proceed further.

To book a table enter 0

To view menu and place order enter 1

To check order state enter 2

To free table enter 3

For Admin Login enter 4

0

Available Tables:

Size 2: 1

Size 4: 0

Size 6: 0

To book a table enter the table size:

Enter 0 to exit

dsdsj

Please Enter correct Table size!

Available Tables:

Size 2: 1

Size 4: 0

Size 6: 0

To book a table enter the table size:

Enter 0 to exit

4

Please wait or try with a different table size

Welcome To the restaurant Smart Kitchen!

The Restaurant is OPEN.

Please choose one of the following options to proceed further.

To book a table enter 0

To view menu and place order enter 1

To check order state enter 2

To free table enter 3

For Admin Login enter 4

0

Available Tables:

Size 2: 1

Size 4: 0

Size 6: 0

To book a table enter the table size:

Enter 0 to exit

1

Please Enter correct Table size!

Available Tables:

Size 2: 1

Size 4: 0

Size 6: 0

To book a table enter the table size:

Enter 0 to exit

2

Table for two booked

Welcome To the restaurant Smart Kitchen!

The Restaurant is FULL.

Please choose one of the following options to proceed further.

Restaurant full . Please wait to book a table

To view menu and place order enter 1

To check order state enter 2

To free table enter 3

For Admin Login enter 4

1

Menu:

Items are in the format:

<ID> <Item Name> <ItemPrice>

Starters:

10 Starter-1 79.0

20 Starter-2 80.0

30 Starter-3 100.0

40 Starter-4 68.0

Main Course:

11 MainCourse-1 90.0

21 MainCourse-2 200.0

31 MainCourse-3 101.0

41 MainCourse-4 250.0

Desserts:

12 Dessert-1 45.0

22 Dessert-2 67.0

32 MyDesert 100.0

Drinks:

13 Drink-1 69.0

23 Drink-2 80.0

Please Enter ID of the Item to Order the Item

Please enter 0 to complete the Order

Please enter -2 to cancel the order

11

Please Enter Item Quantity

1

Menu:

Items are in the format:

<ID> <Item Name> <ItemPrice>

Starters:

10 Starter-1 79.0

20 Starter-2 80.0

30 Starter-3 100.0

40 Starter-4 68.0

Main Course:

11 MainCourse-1 90.0

21 MainCourse-2 200.0

31 MainCourse-3 101.0

41 MainCourse-4 250.0

Desserts:

12 Dessert-1 45.0

22 Dessert-2 67.0

32 MyDesert 100.0

Drinks:

13 Drink-1 69.0

23 Drink-2 80.0

Please Enter ID of the Item to Order the Item

Please enter 0 to complete the Order

Please enter -2 to cancel the order

10

Please Enter Item Quantity

2

Menu:

Items are in the format:

<ID> <Item Name> <ItemPrice>

Starters:

10 Starter-1 79.0

20 Starter-2 80.0

30 Starter-3 100.0

40 Starter-4 68.0

Main Course:

11 MainCourse-1 90.0

21 MainCourse-2 200.0

31 MainCourse-3 101.0

41 MainCourse-4 250.0

Desserts:

12 Dessert-1 45.0

22 Dessert-2 67.0

32 MyDesert 100.0

Drinks:

13 Drink-1 69.0

23 Drink-2 80.0

Please Enter ID of the Item to Order the Item

Please enter 0 to complete the Order

Please enter -2 to cancel the order

13

Please Enter Item Quantity

1

Menu:

Items are in the format:

<ID> <Item Name> <ItemPrice>

Starters:

10 Starter-1 79.0

20 Starter-2 80.0

30 Starter-3 100.0

40 Starter-4 68.0

Main Course:

11 MainCourse-1 90.0

21 MainCourse-2 200.0

31 MainCourse-3 101.0

41 MainCourse-4 250.0

Desserts:

12 Dessert-1 45.0

22 Dessert-2 67.0

32 MyDesert 100.0

Drinks:

13 Drink-1 69.0

23 Drink-2 80.0

Please Enter ID of the Item to Order the Item

Please enter 0 to complete the Order

Please enter -2 to cancel the order

12

Please Enter Item Quantity

1

Menu:

Items are in the format:

<ID> <Item Name> <ItemPrice>

Starters:

10 Starter-1 79.0

20 Starter-2 80.0

30 Starter-3 100.0

40 Starter-4 68.0

Main Course:

11 MainCourse-1 90.0

21 MainCourse-2 200.0

31 MainCourse-3 101.0

41 MainCourse-4 250.0

Desserts:

12 Dessert-1 45.0

22 Dessert-2 67.0

32 MyDesert 100.0

Drinks:

13 Drink-1 69.0

23 Drink-2 80.0

Please Enter ID of the Item to Order the Item

Please enter 0 to complete the Order

Please enter -2 to cancel the order

0

Bill:

Order Number: 1

Item Name: Starter-1 Item quantity: 2 Price: 158.0

Item Name: MainCourse-1 Item quantity: 1 Price: 90.0

Item Name: Drink-1 Item quantity: 1 Price: 69.0

Item Name: Dessert-1 Item quantity: 1 Price: 45.0

Total Price = 362.0

Welcome To the restaurant Smart Kitchen!

The Restaurant is FULL.

Please choose one of the following options to proceed further.

Restaurant full . Please wait to book a table

To view menu and place order enter 1

To check order state enter 2

To free table enter 3

For Admin Login enter 4

3

Enter table size

2

Thank you!

Welcome To the restaurant Smart Kitchen!

The Restaurant is OPEN.

Please choose one of the following options to proceed further.

To book a table enter 0

To view menu and place order enter 1

To check order state enter 2

To free table enter 3

For Admin Login enter 4

1

Menu:

Items are in the format:

<ID> <Item Name> <ItemPrice>

Starters:

10 Starter-1 79.0

20 Starter-2 80.0

30 Starter-3 100.0

40 Starter-4 68.0

Main Course:

11 MainCourse-1 90.0

21 MainCourse-2 200.0

31 MainCourse-3 101.0

41 MainCourse-4 250.0

Desserts:

12 Dessert-1 45.0

22 Dessert-2 67.0

32 MyDesert 100.0

Drinks:

13 Drink-1 69.0

23 Drink-2 80.0

Please Enter ID of the Item to Order the Item

Please enter 0 to complete the Order

Please enter -2 to cancel the order

21

Please Enter Item Quantity

2

Menu:

Items are in the format:

<ID> <Item Name> <ItemPrice>

Starters:

10 Starter-1 79.0

20 Starter-2 80.0

30 Starter-3 100.0

40 Starter-4 68.0

Main Course:

11 MainCourse-1 90.0

21 MainCourse-2 200.0

31 MainCourse-3 101.0

41 MainCourse-4 250.0

Desserts:

12 Dessert-1 45.0

22 Dessert-2 67.0

32 MyDesert 100.0

Drinks:

13 Drink-1 69.0

23 Drink-2 80.0

Please Enter ID of the Item to Order the Item

Please enter 0 to complete the Order

Please enter -2 to cancel the order

0

Bill:

Order Number: 2

Item Name: MainCourse-2 Item quantity: 2 Price: 400.0

Total Price = 400.0

Welcome To the restaurant Smart Kitchen!

The Restaurant is OPEN.

Please choose one of the following options to proceed further.

To book a table enter 0

To view menu and place order enter 1

To check order state enter 2

To free table enter 3

For Admin Login enter 4

1

Menu:

Items are in the format:

<ID> <Item Name> <ItemPrice>

Starters:

10 Starter-1 79.0

20 Starter-2 80.0

30 Starter-3 100.0

40 Starter-4 68.0

Main Course:

11 MainCourse-1 90.0

21 MainCourse-2 200.0

31 MainCourse-3 101.0

41 MainCourse-4 250.0

Desserts:

12 Dessert-1 45.0

22 Dessert-2 67.0

32 MyDesert 100.0

Drinks:

13 Drink-1 69.0

23 Drink-2 80.0

Please Enter ID of the Item to Order the Item

Please enter 0 to complete the Order

Please enter -2 to cancel the order

31

Please Enter Item Quantity

2

Menu:

Items are in the format:

<ID> <Item Name> <ItemPrice>

Starters:

10 Starter-1 79.0

20 Starter-2 80.0

30 Starter-3 100.0

40 Starter-4 68.0

Main Course:

11 MainCourse-1 90.0

21 MainCourse-2 200.0

31 MainCourse-3 101.0

41 MainCourse-4 250.0

Desserts:

12 Dessert-1 45.0

22 Dessert-2 67.0

32 MyDesert 100.0

Drinks:

13 Drink-1 69.0

23 Drink-2 80.0

Please Enter ID of the Item to Order the Item

Please enter 0 to complete the Order

Please enter -2 to cancel the order

0

Bill:

Order Number: 3

Item Name: MainCourse-3 Item quantity: 2 Price: 202.0

Total Price = 202.0

Welcome To the restaurant Smart Kitchen!

The Restaurant is OPEN.

Please choose one of the following options to proceed further.

To book a table enter 0

To view menu and place order enter 1

To check order state enter 2

To free table enter 3

For Admin Login enter 4

1

Menu:

Items are in the format:

<ID> <Item Name> <ItemPrice>

Starters:

10 Starter-1 79.0

20 Starter-2 80.0

30 Starter-3 100.0

40 Starter-4 68.0

Main Course:

11 MainCourse-1 90.0

21 MainCourse-2 200.0

31 MainCourse-3 101.0

41 MainCourse-4 250.0

Desserts:

12 Dessert-1 45.0

22 Dessert-2 67.0

32 MyDesert 100.0

Drinks:

13 Drink-1 69.0

23 Drink-2 80.0

Please Enter ID of the Item to Order the Item

Please enter 0 to complete the Order

Please enter -2 to cancel the order

10

Please Enter Item Quantity

1

Menu:

Items are in the format:

<ID> <Item Name> <ItemPrice>

Starters:

10 Starter-1 79.0

20 Starter-2 80.0

30 Starter-3 100.0

40 Starter-4 68.0

Main Course:

11 MainCourse-1 90.0

21 MainCourse-2 200.0

31 MainCourse-3 101.0

41 MainCourse-4 250.0

Desserts:

12 Dessert-1 45.0

22 Dessert-2 67.0

32 MyDesert 100.0

Drinks:

13 Drink-1 69.0

23 Drink-2 80.0

Please Enter ID of the Item to Order the Item

Please enter 0 to complete the Order

Please enter -2 to cancel the order

-2

Welcome To the restaurant Smart Kitchen!

The Restaurant is OPEN.

Please choose one of the following options to proceed further.

To book a table enter 0

To view menu and place order enter 1

To check order state enter 2

To free table enter 3

For Admin Login enter 4

2

Please Enter Order Number:

3

MainCourse-3 Served

Welcome To the restaurant Smart Kitchen!

The Restaurant is OPEN.

Please choose one of the following options to proceed further.

To book a table enter 0

To view menu and place order enter 1

To check order state enter 2

To free table enter 3

For Admin Login enter 4

4

Please Enter Username

Username

Please Enter password

Password

Admin Menu

1. Add item to Menu.

2. Change Price of Menu Item.

3. Change Password.

4. View Revenue.

5. Logout.

6. Close Restaurant.

Enter your choice.

1

Menu:

Items are in the format:

<ID> <Item Name> <ItemPrice>

Starters:

10 Starter-1 79.0

20 Starter-2 80.0

30 Starter-3 100.0

40 Starter-4 68.0

Main Course:

11 MainCourse-1 90.0

21 MainCourse-2 200.0

31 MainCourse-3 101.0

41 MainCourse-4 250.0

Desserts:

12 Dessert-1 45.0

22 Dessert-2 67.0

32 MyDesert 100.0

Drinks:

13 Drink-1 69.0

23 Drink-2 80.0

Enter Item name. The name should conatin NO spaces.

Drinknew

Choose item Category

1.Starters

2.MainCourse

3.Desserts

4.Drinks

4

Enter price

50

Enter Description

My new Drink

Admin Menu

1. Add item to Menu.

2. Change Price of Menu Item.

3. Change Password.

4. View Revenue.

5. Logout.

6. Close Restaurant.

Enter your choice.

2

Menu:

Items are in the format:

<ID> <Item Name> <ItemPrice>

Starters:

10 Starter-1 79.0

20 Starter-2 80.0

30 Starter-3 100.0

40 Starter-4 68.0

Main Course:

11 MainCourse-1 90.0

21 MainCourse-2 200.0

31 MainCourse-3 101.0

41 MainCourse-4 250.0

Desserts:

12 Dessert-1 45.0

22 Dessert-2 67.0

32 MyDesert 100.0

Drinks:

13 Drink-1 69.0

23 Drink-2 80.0

33 Drinknew 50.0

Please Enter ID of the item

41

Enter new Price

150

Admin Menu

1. Add item to Menu.

2. Change Price of Menu Item.

3. Change Password.

4. View Revenue.

5. Logout.

6. Close Restaurant.

Enter your choice.

4

Revenue is: 9288.0

Admin Menu

1. Add item to Menu.

2. Change Price of Menu Item.

3. Change Password.

4. View Revenue.

5. Logout.

6. Close Restaurant.

Enter your choice.

3

Enter your Username:

Username

Enter your current password:

Password

Enter new Password:

NewPassword

Password changed!

Admin Menu

1. Add item to Menu.

2. Change Price of Menu Item.

3. Change Password.

4. View Revenue.

5. Logout.

6. Close Restaurant.

Enter your choice.

5

Welcome To the restaurant Smart Kitchen!

The Restaurant is OPEN.

Please choose one of the following options to proceed further.

To book a table enter 0

To view menu and place order enter 1

To check order state enter 2

To free table enter 3

For Admin Login enter 4

4

Please Enter Username

Username

Please Enter password

Password

Admin Menu

1. Add item to Menu.

2. Change Price of Menu Item.

3. Change Password.

4. View Revenue.

5. Logout.

6. Close Restaurant.

Enter your choice.

6