Data Structures (2020)

CA Exercise 1 – Restaurant Management System

The objective of this CA exercise is to create a restaurant management system in Python that makes heavy use of data structures. The system should allow the user to:

- Add a new table to the system (which can be booked by diners, etc.). The following information should be stored for each table: table number/identifier, and the number of seats at the table.
- Add a new booking to the system. A booking is made for a table for a given time slot. The following information should be stored for each booking: name of person/customer making the booking, the number of people the booking is for, the table number being booked, the time the booking is for, and the amount of time (in hours) the booking is for.
    - The system should offer a facility to search for and choose a suitable table for a booking (i.e. one that is free for the full specified time slot and has enough seats).
- Add a food/drink menu item to the system. Properties that should be stored are: name of the menu item, and the price.
    - Also provide a facility to delete/remove a menu item.
- Add a purchase to a table/booking. Purchases should have properties: menu item being purchased, and the number/quantity of that item.
    - Also provide a facility to remove/cancel purchases.
- Check-in, check-out and cancel facilities for bookings should be provided. The check-in should allow for a booking to be selected as active/current. All purchases to the assigned table would thereafter be assigned to that booking. A check-out facility would calculate the total owed for the current booking, and remove the booking from the system. A cancel facility simply removes/cancels a booking.
- View all tables and bookings. This would list all tables in the restaurant and all bookings. The items purchased at tables that are yet to check out should also be displayed.
- View menu. View all food/drink menu items and their prices.
- View total receipts. View total money received at check-out, and the total still owed for purchases at active tables.
- Reset facility. Clears all bookings, purchases and total receipts.
- Save and load the entire system data to support persistence between executions.
    - This can be done using any suitable file format (e.g. CSV, XML, binary, etc.). There is no need to use any database system beyond this.
- Other appropriate facilities to manage the restaurant as you see fit.

Notes

- This is an individual CA exercise, and students should work by themselves to complete it.
- This CA exercise is worth approximately 35% of your overall module mark.
- You should implement a suitable Tkinter graphical user interface to interact with your system.
    - The GUI does not have to be particularly fancy but should nevertheless be functional.

Indicative Marking Scheme

- Create/add facilities (table, booking, menu item, purchase) = 15%
- Delete/cancel facilities (booking, menu item, purchase) = 10%
- Booking search/table selection facility = 10%
- View all table/bookings/purchases = 10%
- View menu = 10%
- Check-in, check-out, total receipts facilities = 10%
- Reset facility = 10%
- Save/load facility = 5%
- Appropriate Tkinter user interface = 10%
- General (commenting, style, logical approach, completeness, etc.) = 10%