CS211 - Assignment 2

In this assignment, you need to write a program that allows users to make daily dinner reservations at local restaurants. The program simulates online restaurant reservations systems. This involves designing three classes: *RestaurantReservations*, *Restaurant*, and *Reservation*. All class attributes must be **private** and accessed through public member functions.

The attributes of each class are summarized below:

RestaurantReservations class has the following private attribute:

```
vector <Restaurant> restaurants;      // list of local restaurants
```

Restaurant class has the following private attributes:

```
string restaurantName;                // restaurant name
string restaurantAddress;             // restaurant address
string restaurantCity;                // restaurant city
string restaurantType;                // type of food served at restaurant
int availableSeats [4];               // array holding number of available seats at
                                      // four possible times: 5, 6, 7 and 8 pm
vector <Reservation> reservations;    // list of reservations at this restaurant
```

Reservation class has the following private attributes:

```
const long reservationNum;            // reservation number which is set
                                      // to the value of nextReservationNum
string contactName;                   // name of person making reservation
string contactPhone;                  // person phone number
int groupSize;                        // number of persons in reservation
int reservationTime;                  // reservation time which could be 5, 6, 7 or 8
static long nextReservationNum;       // initialize it to 100 and increment it by 10
                                      // as you create a new Reservation object
```

For each class, you need to write an appropriate constructor, accessor and mutator functions **as needed**. The member functions for the *RestaurantReservations* class are summarized below:

- *void RestaurantReservations::ProcessTransactionFile(string fileName)*
  This function opens the transaction file, named *fileName* and processes its lines one by one. Each line contains a specific command along with its required data. The commands and their formats are:

  - CreateNewRestaurant rName rAddress rCity rType rCapacity
  - FindTable rCity rType rGroup rTime
  - FindTableAtRestaurant rName rGroup

- MakeReservation rName cName cPhone rGroup rTime
- PrintAllRestaurants
- PrintRestaurantReservations rName

A sample transaction file is shown later. If the file could not be opened, you should print an appropriate error message.

- *void RestaurantReservations::CreateNewRestaurant (string rName, string rAddress, string rCity, string rType, int rCapacity)*
  This function creates a new Restaurant object by calling the appropriate constructor and adds it to the restaurants vector. The reservations vector is initially empty and is updated later in the makeReservation method. The array availableSeats's elements are all set to the value of *rCapacity*. The remaining restaurant attributes are set to the argument values *rName, rAddress, rCity, and rType.*

- *void RestaurantReservations::PrintAllRestaurants()*
  This function prints summary information of all restaurants in the restaurants vector. You should print the restaurant name, address, type, and capacity in a neat table format. For example:

```
Restaurant          Address, City             Type         Capacity
-------------------------------------------------------------------
LaPizzeria          4365NobleDr, SanDiego     italian      20
…
```

- *void RestaurantReservations::FindTable(string rCity, string rType, int rGroup, int rTime)*
  This function searches for all restaurants in a specific city, *rCity*, of a specific type, *rType*, and having at a specific time, *rTime*, enough seats for a given group of people, *rGroup*. If such restaurants are found, the function prints the names of all such restaurants. For example:

```
FindTable SanDiego italian 10 5
You may reserve a table for 10 at 5 pm at:
LaPizzeria
Civico
```

  Otherwise, it prints an appropriate error message.

```
FindTable SanDiego italian 10 5
No restaurant can accommodate such a group at this time, check another time
```

- *void RestaurantReservations::FindTableAtRestaurant (string rName, int rGroup)*
  This function finds all possible availabilities for a given group of people, rG*roup*, at a specific restaurant whose name is *rName*. In other words, it prints the possible times at which such a group can reserve a table at this restaurant. For example:

```
FindTableAtRestaurant LaPizzeria 2
```

```
You can reserve a table for 2 at LaPizzeria at 7:00 pm, 8:00 pm
```

If the restaurant name is invalid or the restaurant can not accommodate such a group at any time, this function prints an appropriate error message.

```
FindTableAtRestaurant LaPizzeria 8
LaPizzeria does not have such availability
```

- *void RestaurantReservations::MakeReservation (string rName, string cName, string cPhone, int rGroup, int rTime)*
  This function creates a new Reservation object and adds it to the reservations vector of the restaurant whose name is *rName*. You can use the push_back function to add the new object at then of the vector. The customer name, phone, group and time should be set to *cName, cPhone, rGroup,* and *rTime* respectively. reservationNum for the new object should be set to the value of nextReservationNum and then nextReservationNum should be incremented by 10. The function also updates the available seats array at that specific time. Note that the first array element corresponds to the available seats at 5:00 pm. The second corresponds to the availabilities at 6:00 pm and so on.

- *void RestaurantReservations::PrintRestaurantReservations (string rName)*
  This function prints the details of all reservations at the restaurant whose name is *rName*. Details include the following information: reservation number, customer name, phone number, group size and reservation time. For example:

```
PrintRestaurantReservations Barrio

Reservation   Contact       Phone         Group  Time
-----------------------------------------------------
170           Jenna         633-782-9848  4      8:00 PM
180           Davina        315-380-9848  20     7:00 PM
190           Pete          215-380-9845  6      7:00 PM
```

Note: Any member function that does not modify the attributes must be made constant.

After designing your classes (and breaking up your classes into .cpp and .h files), write a main program that instantiates an object of type *RestaurantReservations* and calls a method to read a transaction file and process its commands. Your main should look like this:

```
int main()
{
        RestaurantReservations openTable;
        openTable.ProcessTransactionFile("TransactionFile.txt");

        return 0;
```

}

Then test your code by including the following commands in your transaction file:

```
CreateNewRestaurant LaPizzeria 4356NobelDr SanDiego italian 10
CreateNewRestaurant Civico 8765RegentsRd SanDiego italian 20
CreateNewRestaurant OnBorder 101DoveSt SanDiego mexican 20
CreateNewRestaurant Ortega 3465RegentsRd SanDiego mexican 30
CreateNewRestaurant OlivioRistorante 4320CaminoDr Carlsbad italian 15
CreateNewRestaurant MariaRistorante 6534SpringburstDr Carlsbad italian 10
CreateNewRestaurant Agave 8764CreekViewBlvd Carlsbad mexican 30
CreateNewRestaurant Barrio 5Broadway Carlsbad mexican 30
PrintAllRestaurants
FindTable SanDiego italian 10 5
MakeReservation LaPizzeria John 858-334-3334 10 5
FindTable SanDiego italian 15 5
MakeReservation Civico Jamie 316-324-1234 15 5
FindTable SanDiego italian 10 5
FindTable SanDiego italian 10 6
MakeReservation LaPizzeria Andrea 513-984-7878 10 6
FindTable SanDiego italian 5 7
MakeReservation LaPizzeria Amy 838-234-3111 5 7
FindTable SanDiego mexican 10 8
MakeReservation OnBorder Bill 222-532-3148 10 8
FindTable SanDiego mexican 10 8
MakeReservation OnBorder Andre 325-632-9148 10 8
FindTable SanDiego mexican 2 8
MakeReservation Ortega Kathy 617-682-5148 2 8
FindTable Carlsbad mexican 4 8
MakeReservation Barrio Jenna 633-782-9848 4 8
FindTable Carlsbad mexican 20 7
MakeReservation Barrio Davina 315-380-9848 20 7
MakeReservation Barrio Pete 215-380-9845 6 7
FindTable Carlsbad mexican 5 7
MakeReservation Agave Joe 399-300-1848 5 7
FindTable Carlsbad italian 5 6
MakeReservation OlivioRistorante Courtney 229-456-1865 5 6
MakeReservation OlivioRistorante Patrick 665-102-0876 5 6
MakeReservation MariaRistorante Nancy 218-396-2349 5 6
FindTableAtRestaurant LaPizzeria 2
MakeReservation LaPizzeria Carolyn 613-084-7898 2 7
FindTableAtRestaurant LaPizzeria 3
MakeReservation LaPizzeria Edward 313-284-0087 3 7
FindTableAtRestaurant LaPizzeria 3
MakeReservation LaPizzeria Michael 215-487-1082 3 8
FindTableAtRestaurant Agave 26
MakeReservation Agave Joe 615-233-1065 26 5
FindTableAtRestaurant LaPizzeria 8
PrintRestaurantReservations LaPizzeria
PrintRestaurantReservations Civico
PrintRestaurantReservations Ortega
PrintRestaurantReservations OlivioRistorante
PrintRestaurantReservations Agave
PrintRestaurantReservations Barrio
PrintRestaurantReservations Season23
```

Once done, you need to submit the following:

1. A hard copy (printed copy) of the source codes (RestaurantReservations.h, Restaurant.h, Reservation.h, RestaurantReservations.cpp, Restaurant.cpp,

Reservation.cpp, and main.cpp) and the output run placed in a folder. You should hand in the folder during the first 5 minutes of the class when the homework is due. You lose 10% of the grade if you do not provide a complete folder.

2. An electronic copy of your source code (no output). You should place all source codes and executable file in a zipped folder and name it based on your first/last name and the assignment number. For example, JoeSmith-A1.zip. You should submit the zipped folder online through cougar website.