*1. Thread Synchronization Project 2*

**Restaurant Problem**

Open the folder BENSCHILLIBOWL and locate the main.c and BENSCHILLIBOWL.c files. You are simulating Customers and Cooks at the local Bens Chilli Bowl Restaurant. Customers and Cooks are represented as Threads using the pthread library. You are to modify these two files (and only these files), and add the implementation for each of the functions listed in the file. You can add any additional helper functions and data types as needed. Feel free to browse BENSCHILLIBOWL.h. This file defines the Order, Menu, and Restaurant structs.

**Functions to modify in main.c**

*/\* Thread funtion that represents a customer. A customer should: \*/*

*/\* - allocate space (memory) for an order. \*/*

*/\* - select a menu item. \*/*

*/\* - populate the order with their menu item and their customer ID. \*/*

*/\* - add their order to the restaurant.\*/*

**void**\* **BENSCHILLIBOWLCustomer**(**void**\* tid);

*/\* Thread function that represents a cook in the restaurant. A cook should:\*/*

*/\* - get an order from the restaurant. \*/*

*/\* - if the order is valid, it should fulfill the order, and then \*/*

*/\* free the space taken by the order. \*/*

*/\* The cook should take orders from the restaurants until it does not \*/*

*/\* receive an order.\*/*

**void**\* **BENSCHILLIBOWLCook**(**void**\* tid);

*/\* Runs when the program begins executing. This program should:\*/*

*/\* - open the restaurant (instantiate a restaurant [i.e. allocate the memory])\*/*

*/\* - create customers and cooks (Create the Threads)\*/*

*/\* - wait for all customers and cooks to be done (join)\*/*

*/\* - close the restaurant.\*/*

**int** **main**();

**Functions to modify in BENSCHILLIBOWL.c**

You are to modify the file BENSCHILLIBOWL.cBENSCHILLIBOWL.c and add the implementation for each of the functions listed below. If you have already started you will see a parameter based to all the functions of type BENSCHILLIBOWL \*mcg. All references to \*mcg should be changed to \*bcb to reference the global variable declared in main.c

/\*Select a random item **from** **the** Menu **and** return **it** \*/

MenuItem PickRandomMenuItem();

*/\*Allocate memory for the Restaurant, then create the mutex and condition variables needed to instantiate the Restaurant\*/*

BENSCHILLIBOWL\* **OpenRestaurant**(**int** max\_size, **int** expected\_num\_orders);

/\*check **that** **the** number **of** orders received **is** **equal** **to** **the** number handled (ie.fullfilled). Remember **to** deallocate your resources\*/

void CloseRestaurant(BENSCHILLIBOWL\* bcb);

*/\*add an order to the back of queue \*/*

int **AddOrder(BENSCHILLIBOWL\* bcb, Order\* order);**

/\* remove an **order** **from** the queue \*/

**Order** **\*GetOrder**(BENSCHILLIBOWL\* bcb);

*/\* Optional helper functions (you can implement if you think they would be useful)\*/*

**bool** IsFull(**BENSCHILLIBOWL\* bcb);**

/\* this methods adds **order** **to** rear of queue \*/

void AddOrderToBack(**Order** **\*\*orders**, **Order** **\*order**) {}