**PROGRAM DESCRIPTION**

To design the system as discussed earlier, we identified the six objects that need to be included in the program. These objects are made more generic and their classes are declared and while execution of program objects of each classes are made. The classes defined in the program are kept in their separate headers to make code more systematic and easier to debug. The classes in the program along with their attributes are as follows:

1. Customer

***Class methods***

**checkaccount():**

This Authorizes whether the user is the registered customer of the café or not. In case it didn’t find the record notifies the user about the the error and if the details matches logs the user in the sytem.

**setdetails():**

it asks the details of the user and puts them in the record file as a new record. This method is called if the user is new to the restaurant.

**showdetails():**

It shows the current details of the user stored in the file if he is logged in to the system.

**selectmode():**

Once the user enters the system he is asked whether he is a new user or a registered one. The purpose of this method is to call the *checkaccount()* method if the user claims he is the registered one or to call the *setdetails()* method if he is new to the café and wants to get registerd.

**updaterecord():**

This method functions to update the the account balance of the user and modify the previously stored balance in the record file.

**getfilebalance():**

this method serves to get the account balance of the object as the member is private in the class and can only be used by the member functions in the class.

**Rechargebalance():**

This method is used to update the balance of the user if he recharges his account balance.

***class members***

balance(static type), onetimetotal(static type),name,age,address,gender,username,password,filebalance,recordnumber

1. Food

***Class methods***

**Showmenu()(virtual method):**

This shows the current type object menu list. It is declared as virtual function. If there is another function of the same name in the derived class then the function of the object pointed to is called.

**Foodselect()(virtual method):**

This selects the food items of the corresponding menu list being displayed.

**Updatebalance(int):**

It takes the current food item selected and updates the account balance of the user and also error handling has been included in the method using try-catch-throw and checks whether the food item price exceeds the account balance or not and throws an error message if the case is detected and provides the user an option to recharge his account.

***Class members:***

**Price(static type):**

It is of static type member variable. It has only one copy throughout its program and its value can be updated by any of the object of the this class as well its derived classes.

1. Breakfast

***Class methods***

**Showmenu():**

This shows the menu list of the breakfast items.

**Foodselect():**

This selects the food items of the list under breakfast.

1. Lunch

***Class methods***

**Showmenu():**

This shows the menu list of the lunch items.

**Foodselect():**

This selects the food items of the list under lunch.

1. Dinner

***Class methods***

**Showmenu():**

This shows the menu list of the dinner items.

**Foodselect():**

This selects the food items of the list under dinner.

1. Manipulators

***Class methods***

**Manipulators(int n) (parameterized constructor):**

It sets the character field to be wriiten based on the integer n passed to it and also sets the fill option in the unused spaces.

**operator<<(ostream&,manipulators)( friend function of the class):**

This is the friend function of the class. This has the return type of ostream class.

There is another function that is not the member function of the class but included in the header manipulators.h which is named as **sf(int)** .It sets the num variable to its desired value.