```cpp
/*
        Lab 1
        CS2
        Updated by: FIXME
        Date: FIXME

        Breakfast billing system for a restaurant.

        This program allows restaurant customers to select breakfast items and prints the bill.
*/
#include <iostream>
#include <fstream>
#include <iomanip>
#include <string>

using namespace std;

// max no. of menu items that can be stored by the program
const int NO_OF_ITEMS = 8;

// data structure to hold each menu item info
struct menuItemType
{
       string itemName;
       double itemPrice;
};

// initialize menu list with some default values
void initMenuList(menuItemType menuList[], int menuListSize);
// function that reads menu data into menuList array
void readMenuData(ifstream& fin, menuItemType menuList[], int menuListSize);
// function that displays the menu
void showMenu(menuItemType menuList[], int menuListSize);
// function to print check after menu items are selected
void printCheck(menuItemType menuList[], int menuListSize,
       int selectedList[], int selectedListLength);
// function to select items from the menu list
void makeSelection(menuItemType menuList[], int menuListSize,
       int selectedList[], int& selectedListLength);
// function that returns true if the itemNo is in the selectedList otherwise false.
// this function avoids user repeating the menu item
bool isItemSelected(int selectedList[], int selectedListLength, int itemNo);

int main()
{
       menuItemType menuList[NO_OF_ITEMS];
       int choiceList[NO_OF_ITEMS]; // array to keep track of choices based on item number
       int choiceListLength; // variable to keep track of no. of choices
       ifstream fin;
       cout << fixed << showpoint << setprecision(2);
       // open menu.txt file to read menu data from
       fin.open("menu.txt");
       if (!fin)
       {
              cout << "Input file does not exist. Program Terminates!"
                     << endl;
              cin.get();
              return 1;
       }
       initMenuList(menuList, NO_OF_ITEMS);
       // FIXME
       // Call readMenuData function passing proper arguments
       readMenuData(fin, menuList, NO_OF_ITEMS);
       fin.close(); //close input file
```

```
        showMenu(menuList, NO_OF_ITEMS);
        makeSelection(menuList, NO_OF_ITEMS,
                choiceList, choiceListLength);
        printCheck(menuList, NO_OF_ITEMS,
                choiceList, choiceListLength);
        // pause the program until enter his hit
        cin.get();
        cin.get();
        return 0;
}

void initMenuList(menuItemType menuList[], int menuListSize)
{
        for (int i = 0; i < menuListSize; ++i)
        {
                // FIXME
                // Initialize each menu item's name to A Tasty Item
                // Initialize each menu item's price to 0;
                menuList[i].itemName = "A Tasty Item";
                menuList[i].itemPrice = 0;
        }
}

void readMenuData(ifstream& fin, menuItemType menuList[], int menuListSize)
{
        char ch;
        for (int i = 0; i < menuListSize; i++)
        {
                // FIXME
                // 1. Read each item name into itemName field of menuList array
                // 2. Read each item price into itemPrice field of menuList array
                getline(fin, menuList[i].itemName);
                fin >> menuList[i].itemPrice;
                fin.get(ch); // read and ignore \n char
        }
}

void showMenu(menuItemType menuList[], int menuListSize)
{
        cout << "Welcome to Papi Joey's Kitchen" << endl;
        cout << "----Today's Menu----" << endl;

        for (int i = 0; i < menuListSize; i++)
        {
                cout << i + 1 << ": " << left << setw(15) << menuList[i].itemName
                        << right << " $" << menuList[i].itemPrice << endl;
        }
        cout << endl;
}

void printCheck(menuItemType menuList[], int menuListSize,
        int selectedList[], int selectedListLength)
{
        int i;
        double salesTax;
        double amountDue = 0;

        cout << "   Papi Joey's Kitchen" << endl;
        cout << "       Guest Check    " << endl;
        cout << setw(25) << setfill('=') << endl;
        cout << setfill(' ') << endl;
        for (i = 0; i < selectedListLength; i++)
        {
                cout << left << setw(15) << menuList[selectedList[i]].itemName
```

```
                        << right << " $" << setw(4) << menuList[selectedList[i]].itemPrice << endl;
                amountDue += menuList[selectedList[i]].itemPrice;
        }
        cout << endl;
        salesTax = amountDue * .07;
        cout << left << setw(15) << "Tax " << right << " $"
                << salesTax << endl;
        amountDue = amountDue + salesTax;
        cout << left << setw(15) << "Amount Due " << right
                << " $" << amountDue << endl << endl;
        cout << setw(25) << setfill('=') << ' ' << endl;
        cout << setfill(' ') << endl;
        cout << "     Thank you! " << endl;
        // FIXME
        // 1. Prompt user to enter output filename to write receipt data.
        // 2. Create the file and write the check info (as you see on the common output) into the file.
        // 3. Provide feedback to the user, once done.
        // 4.. Close the file.
}

void makeSelection(menuItemType menuList[], int menuListSize, int selectedList[],
 int& selectedListLength)
{
        int selectionNo = 0;
        int itemNo;
        char response;
        selectedListLength = 0;

        cout << "You can make up to " << menuListSize
                << " single order selections" << endl;

        cout << "Do you want to make selection Y/y (Yes), N/n (No): ";
        cin >> response;
        cout << endl;

        while ((response == 'Y' || response == 'y') &&
                selectedListLength < 8)
        {
                cout << "Enter item number: ";
                cin >> itemNo;
                cout << endl;

                if (!isItemSelected(selectedList, selectedListLength, itemNo))
                        selectedList[selectedListLength++] = itemNo - 1;
                else
                        cout << "Item already selected" << endl;

                cout << "Select another item Y/y (Yes), N/n (No): ";
                cin >> response;
                cout << endl;
        }
}

bool isItemSelected(int selectedList[], int selectedListLength, int itemNo)
{
        /*
        FIXME
        Algorithm steps:
        1. Go through each item in selectedList and check if itemNo is in there.
        2. If it is, return true, otherwise false
        */
        bool found = false;

        for (int i = 0; i < selectedListLength; i++)
        if (selectedList[i] == itemNo)
        {
                found = true;
                break;
        }
        return found;
}
```