**Module 04 – Bob’s Restaurant**

Tim Mastarone

Rasmussen University

Microsoft C# Programming

Instructor: Jim Barringer

Module 04 - Lab 2

April 27, 2024

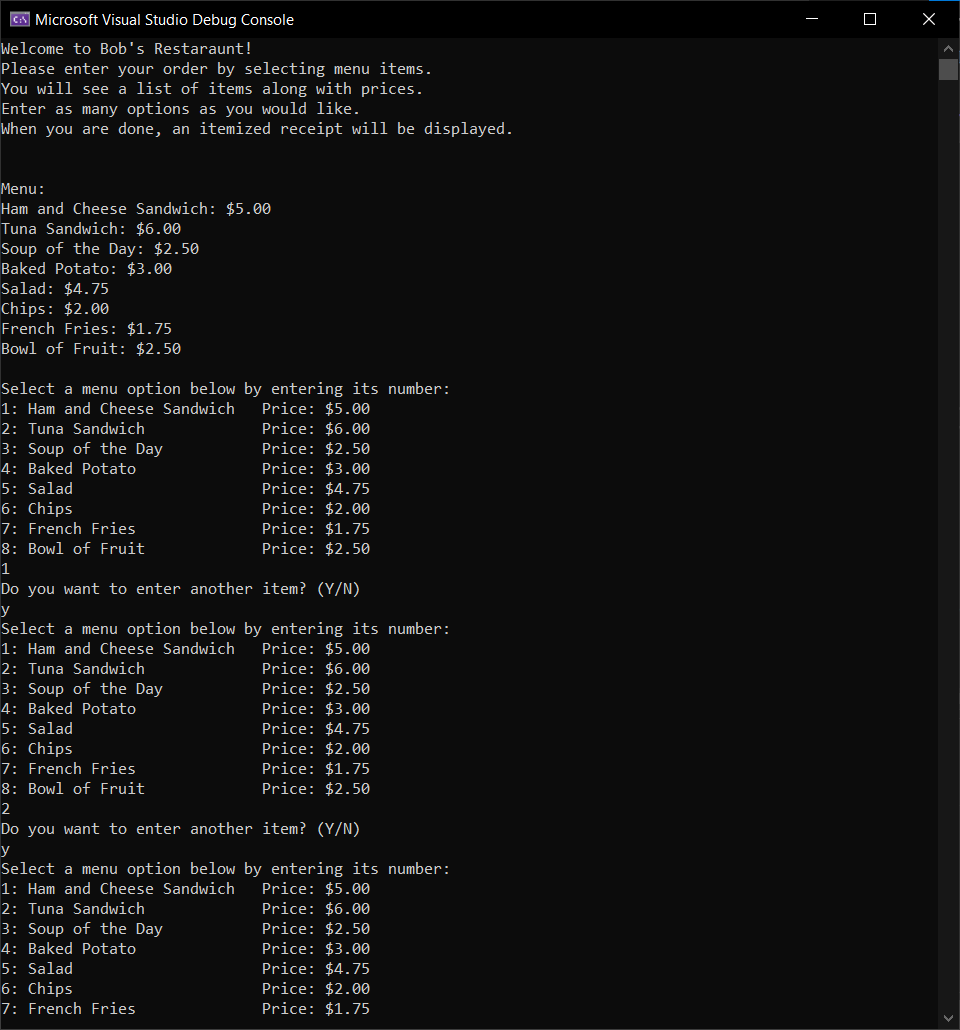
The Bob’s Restaurant application shows the user a menu of food options and allows them to select as many as they choose. The items are chosen by entering a number (1-8). When all items have been entered the user enters ‘N’ and the check/receipt is displayed. The receipt shows each item and its prices. The check shows the tax amount and the total with tax included.

The snip below is a user entering 2 items:

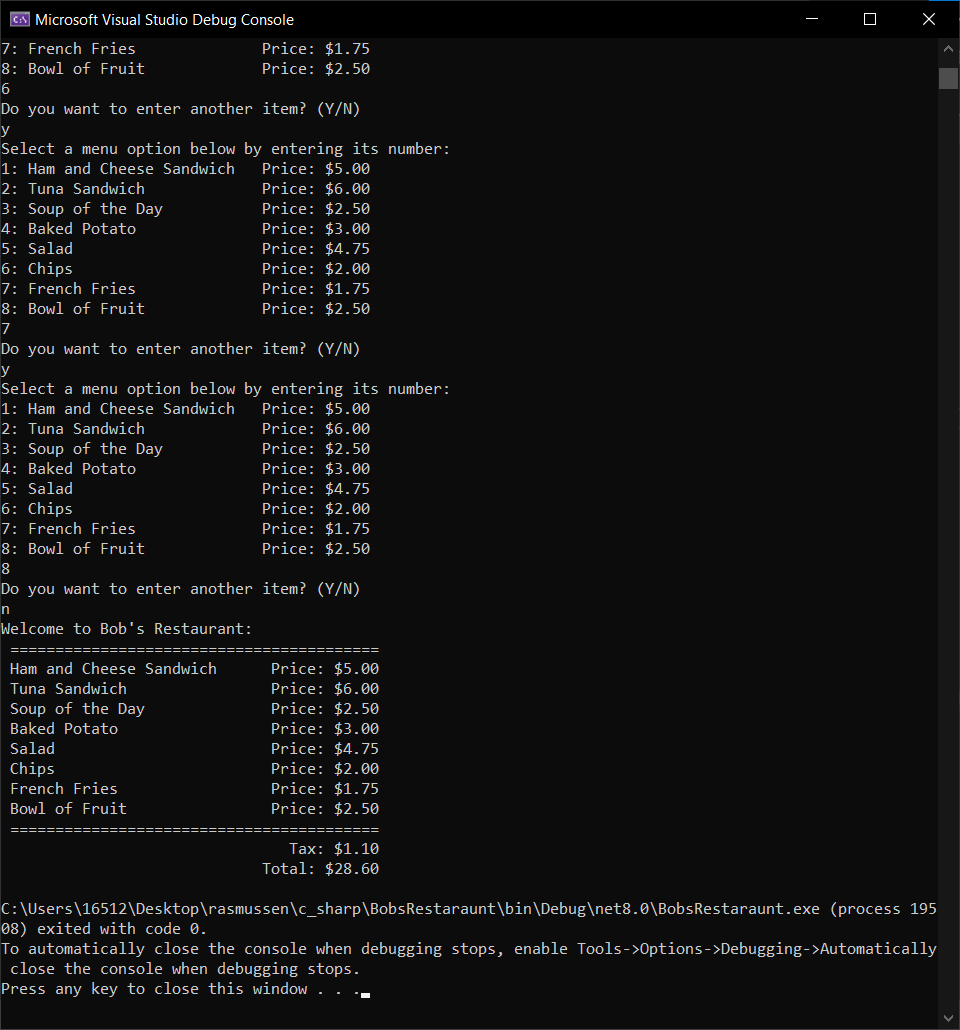
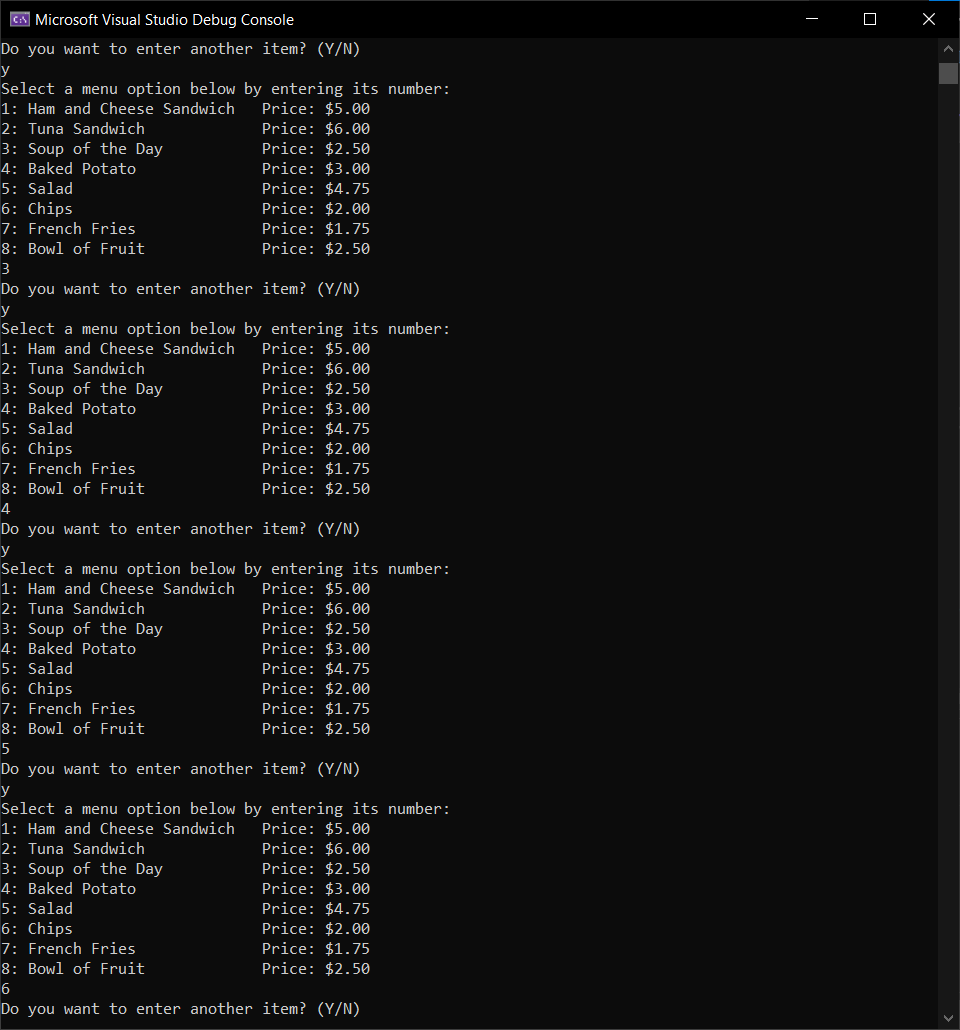
A screenshot of a computer

Description automatically generated

The user can choose to continue to enter more items. The snip below shows a user entering 1 of everything on the menu and ending the loop to see the total:



Continued on the next page.



If the user enters a number that is not on the menu, they see a message and they are allowed to enter a new one. If the user enters a string or nothing, they are shown a different message and also allowed to enter a new option.

A screenshot of a computer

Description automatically generated

The source code for this project is public at:  
<https://github.com/timmasta/rasmussen/blob/main/c_sharp/BobsRestaraunt>   
The code below is from my Program.cs file that contains all functionality of the program:

//Program.cs

// Define the struct menuItemType

public struct menuItemType

{

public string menuItem;

public double menuPrice;

}

class Program

{

// Define const arrays for menu items and prices

static string[] menuItems = { "Ham and Cheese Sandwich", "Tuna Sandwich", "Soup of the Day", "Baked Potato", "Salad", "Chips", "French Fries", "Bowl of Fruit" };

// Prices corresponding to the menu items -- Ensure the prices are in the same order as the menu items

static double[] menuPrices = { 5.00, 6.00, 2.50, 3.00, 4.75, 2.00, 1.75, 2.50 };

static menuItemType[] getData(string[] menuItems, double[] menuPrices)

{

int size = Math.Min(menuItems.Length, menuPrices.Length);//get the smaller number so we do not go past the bounds of either array

menuItemType[] tempMenuList = new menuItemType[size];

// Populate the menuList array

for (int i = 0; i < size; i++)

{

tempMenuList[i].menuItem = menuItems[i];

tempMenuList[i].menuPrice = menuPrices[i];

}

return tempMenuList;

}

static void printGreeting()

{

Console.WriteLine("Welcome to Bob's Restaurant!");

Console.WriteLine("Please enter your order by selecting menu items.");

Console.WriteLine("You will see a list of items along with prices.");

Console.WriteLine("Enter as many options as you would like.");

Console.WriteLine("When you are done, an itemized receipt will be displayed.");

Console.WriteLine("");

}

static void showMenu(string[] menuItems, double[] menuPrices)

{

int size = Math.Min(menuItems.Length, menuPrices.Length);//get the smaller number so we do not go past the bounds of either array

Console.WriteLine("Select a menu option below by entering its number:");

// Find the length of the longest menu item

int maxMenuItemLength = 0;

for (int i = 0; i < size; i++)

{

if (menuItems[i].Length > maxMenuItemLength)

{

maxMenuItemLength = menuItems[i].Length;

}

}

// Calculate the width of the first column

int firstColumnWidth = maxMenuItemLength + 6; // Add extra spacing

// Print menu options with two columns

for (int i = 0; i < size; i++)

{

//format the string to print for each line including menu number, food, and price

string menuItemText = $"{(i + 1).ToString()}: {menuItems[i]}";

string priceText = $"Price: ${menuPrices[i].ToString("F2")}";

string spacing = new string(' ', firstColumnWidth - menuItemText.Length);

Console.WriteLine($"{menuItemText}{spacing}{priceText}");

}

}

static List<int> getUserOrder()

{

double orderTotal = 0;

List<int> choicesList = []; // Store user choices (array indices) in a list for use in the receipt

string? userInput;

int userOption;

while (true)

{

showMenu(menuItems, menuPrices);

userInput = Console.ReadLine();

if (int.TryParse(userInput, out userOption))

{

if (userOption > 0 && userOption < menuItems.Length + 1)

{

orderTotal += menuPrices[userOption - 1];

choicesList.Add(userOption - 1); // Add user choice index to the list, subtract one because we are using this for indexing

Console.WriteLine("Do you want to enter another item? (Y/N)");

userInput = Console.ReadLine();

if (userInput != null && userInput.ToLower() == "n")

{

return choicesList;

}

else

{

continue;

}

}

else

{

Console.WriteLine($"Please enter a valid menu option (1-{menuItems.Length}).");

}

}

else

{

Console.WriteLine("Please enter a valid menu option.");

}

}

}

static double calculateTotal(List<int> items)

{

double itemsTotal = 0;

foreach (int item in items)

{

itemsTotal += menuPrices[item];

}

return itemsTotal;

}

static double calculateTax(double subtotal)

{

return (subtotal \* .04);

}

static void printReceipt(List<int> items)

{

//set up the spacing for the columns

// Find the length of the longest menu item

int maxMenuItemLength = 0;

for (int i = 0; i < items.Count; i++)

{

if ((menuItems[i].Length) > maxMenuItemLength)

{

maxMenuItemLength = menuItems[i].Length;

}

}

// Calculate the width of the first column

int firstColumnWidth = maxMenuItemLength + 6; // Add extra spacing

// Print menu options with two columns

for (int i = 0; i < items.Count; i++)

{

//format the string to print for each line including menu number, food, and price

string menuItemText = $"{menuItems[items[i]]}";//reference the array with the saved index from get user order

string priceText = $"Price: ${menuPrices[items[i]].ToString("F2")}";

string spacing = new string(' ', firstColumnWidth - menuItemText.Length);

Console.WriteLine($" {menuItemText}{spacing}{priceText}");

}

}

static void printCheck(double subtotal, List<int> userChoices)

{

Console.WriteLine("Welcome to Bob's Restaurant:");

Console.WriteLine(" =========================================");

printReceipt(userChoices);//shows an itemized receipts with prices

Console.WriteLine(" =========================================");

double tax = calculateTax(subtotal);

string taxString = $" Tax: ${tax.ToString("F2")}";

string totalString = $" Total: ${(subtotal + tax).ToString("F2")}";

// Calculate spacing dynamically based on the length of the tax and total strings

int taxSpacingLength = 42 - taxString.Length;

int totalSpacingLength = 42 - totalString.Length;

string taxSpacing = new string(' ', Math.Max(taxSpacingLength, 0));

string totalSpacing = new string(' ', Math.Max(totalSpacingLength, 0));

Console.WriteLine($"{taxSpacing}{taxString}");

Console.WriteLine($"{totalSpacing}{totalString}");

}

static void Main(string[] args)

{

// Define an array of menuItemType and fill it with the GetData function

menuItemType[] menuList = getData(menuItems, menuPrices);

//Display the welcome message

printGreeting();

Console.WriteLine("");//add new line padding

// Print out the menu items and prices

Console.WriteLine("Menu:");

foreach (var item in menuList)

{

Console.WriteLine($"{item.menuItem}: ${item.menuPrice.ToString("F2")}");

}

Console.WriteLine("");//add new line padding

//getUserOder displays options and returns a list of integers representing user choices

List<int> choicesList = getUserOrder();

double subtotal = calculateTotal(choicesList);

printCheck(subtotal, choicesList);

}

}