```csharp
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10
11
12 //Programmer: Aaron Yoon
13 //Date: 12/2/22
14 //Project: Taco truck simulation
15
16
17
18 namespace Taco_Truck_Tuesday
19 {
20     public partial class Form1 : Form
21     {
22         //declare manager info variables
23         int totalSales = 0;
24         double totalIncome = 0;
25         double queuedPrice = 0;
26
27
28         //declare the prices
29
30         //shells
31         static double softShellPrice = 2;
32         static double hardShellPrice = 2.5;
33         static double doritoShellPrice = 3;
34
35         //meats
36         static double beefMeatPrice = 1.5;
37         static double chickenMeatPrice = 2.25;
38         static double porkMeatPrice = 1;
39
40         //toppings
41         static double cheeseTopPrice = 0.5;
42         static double guacTopPrice = 0.75;
43         static double salsaTopPrice = 0.25;
44         static double onionTopPrice = 0.3;
45         static double tomatoTopPrice = 0.6;
46
47         double tax = 1.0775;
48         public Form1()
49         {
```

```csharp
50              InitializeComponent();
51          }
52
53          private void pictureBox4_Click(object sender, EventArgs e)
54          {
55
56          }
57
58          private void btnQuit_Click(object sender, EventArgs e)
59          {
60              this.Close();
61          }
62
63          private void btnComplete_Click(object sender, EventArgs e)
64          {
65              //adds total sales and adds the order price to the total
                    income and makes the average sale variable
66              totalSales += 1;
67              totalIncome += queuedPrice;
68              double avgSale = totalIncome / totalSales;
69              //lblOutput.Text = queuedPrice.ToString("C");
70
71              DialogResult resultAnswer = new DialogResult();
72
73              resultAnswer = MessageBox.Show("Complete Order?", "FINAL
                    ORDER", MessageBoxButtons.YesNo);
74
75
76
77              if(resultAnswer == DialogResult.Yes)
78              {
79                  MessageBox.Show("Your total is: " + queuedPrice.ToString
                        ("C"), "SUMMARY", MessageBoxButtons.OK);
80                  clear();
81              }
82
83
84
85          }
86
87          private double getShell()
88          {
89              //gets the price of what shell is selected
90              if (radShellSoft.Checked)
91                  return softShellPrice;
92              else if (radShellHard.Checked)
93                  return hardShellPrice;
94              else
95                  return doritoShellPrice;
```

```
 96            }
 97
 98            private string getShellName()
 99            {
100                //returns what shell is ordered to make outputs easier
101                if (radShellSoft.Checked)
102                    return "Soft Shell";
103                else if (radShellHard.Checked)
104                    return "Hard Shell";
105                else
106                    return "Dorito Shell";
107            }
108
109            private double getMeat()
110            {
111                //gets the price of what meat is selected
112                if (radMeatBeef.Checked)
113                    return beefMeatPrice;
114                else if (radMeatChicken.Checked)
115                    return chickenMeatPrice;
116                else
117                    return porkMeatPrice;
118            }
119
120            private string getMeatName()
121            {
122                //returns what meat is chosen in a string to make outputs    ⤴
                       easier
123                if (radMeatBeef.Checked)
124                    return "Beef";
125                else if (radMeatChicken.Checked)
126                    return "Chicken";
127                else
128                    return "Pork";
129            }
130            private double getToppings()
131            {
132                //gets the total price of all the toppings selected
133                double toppingPrice = 0;
134                if (chkCheese.Checked)
135                    toppingPrice += cheeseTopPrice;
136                if (chkGuac.Checked)
137                    toppingPrice += guacTopPrice;
138                if (chkOnion.Checked)
139                    toppingPrice += onionTopPrice;
140                if (chkSalsa.Checked)
141                    toppingPrice += salsaTopPrice;
142                if (chkTomato.Checked)
143                    toppingPrice += tomatoTopPrice;
```

```
144
145             return toppingPrice;
146
147         }
148
149      private string getToppingNames()
150      {
151         //returns the string of what toppings are ordered to make    ⇁
                outputs easier
152         string toppings = "";
153
154         if (chkCheese.Checked)
155             toppings += "Cheese ";
156         if (chkGuac.Checked)
157             toppings += "Guacamole ";
158         if (chkOnion.Checked)
159             toppings += "Onion ";
160         if (chkSalsa.Checked)
161             toppings += "Salsa ";
162         if (chkTomato.Checked)
163             toppings += "Tomato ";
164
165         return toppings;
166      }
167
168      private string[] getOrder(string quantity1, string shell1, string   ⇁
           meat1, string toppings1)
169      {
170         //makes an array based on the parameters and returns it
171         string[] orderList =
172         {
173             quantity1,
174             shell1,
175             meat1,
176             toppings1
177         };
178
179         return orderList;
180      }
181
182      private double getPrice()
183      {
184         //gets the price of the order with credit being an option
185         if (chkCredit.Checked)
186             return ((getMeat() + getShell() + getToppings()) * tax *   ⇁
                (int.Parse(txtQuantity.Text)));
187         else
188             return ((getMeat() + getShell() + getToppings()) *         ⇁
                (int.Parse(txtQuantity.Text)));
```

```csharp
189            }
190
191        private DialogResult displayOrder(string[] order)
192        {
193            //displays the order in a messagebox
194            return MessageBox.Show(order[0] + " " + order[1] + " " + order
                    [2] + " Tacos With " + order[3] + "\n"
195                + "for " + txtFirstName.Text + " " + txtLastName.Text +
                    "\n"
196                + getPrice().ToString("C") + "\n"
197                + txtOrderNum.Text,
198                "Order",
199                MessageBoxButtons.OKCancel
200                );
201        }
202        private void btnSummary_Click(object sender, EventArgs e)
203        {
204            //shows all the stats of everything
205            MessageBox.Show("Earned: " + totalIncome.ToString("C") + "\n"
                    +
206                "Total Customers: " + totalSales.ToString() + "\n" +
207                "Average Income: " + (totalIncome / totalSales).ToString
                    ("C")
208                );
209        }
210
211        private void btnAdd_Click(object sender, EventArgs e)
212        {
213            //make the array variable
214            string[] order1 = getOrder(txtQuantity.Text, getShellName(),
                    getMeatName(), getToppingNames());
215            //displays the order
216            DialogResult resultAnswer = new DialogResult();
217            resultAnswer = displayOrder(order1);
218
219            queuedPrice += getPrice();
220
221
222            if (resultAnswer == DialogResult.OK)
223            {
224                txtFirstName.Enabled = false;
225                txtLastName.Enabled = false;
226                txtOrderNum.Enabled = false;
227            }
228        }
229
230        private void Form1_Load(object sender, EventArgs e)
231        {
232
```

```csharp
233            }
234
235        private void btnClear_Click(object sender, EventArgs e)
236        {
237            //calls the clear method when btnClear is pressed
238            clear();
239        }
240
241        private void clear()
242        {
243            //clears everything
244            txtFirstName.Enabled = true;
245            txtLastName.Enabled = true;
246            txtOrderNum.Enabled = true;
247            txtFirstName.Text = null;
248            txtLastName.Text = null;
249            txtOrderNum.Text = null;
250            txtQuantity.Text = null;
251
252            chkCheese.Checked = false;
253            chkGuac.Checked = false;
254            chkOnion.Checked = false;
255            chkSalsa.Checked = false;
256            chkTomato.Checked = false;
257
258            radMeatBeef.Checked = false;
259            radMeatChicken.Checked = false;
260            radMeatPork.Checked = false;
261            radShellDorito.Checked = false;
262            radShellHard.Checked = false;
263            radShellSoft.Checked = false;
264
265            queuedPrice = 0;
266        }
267    }
268 }
269
```