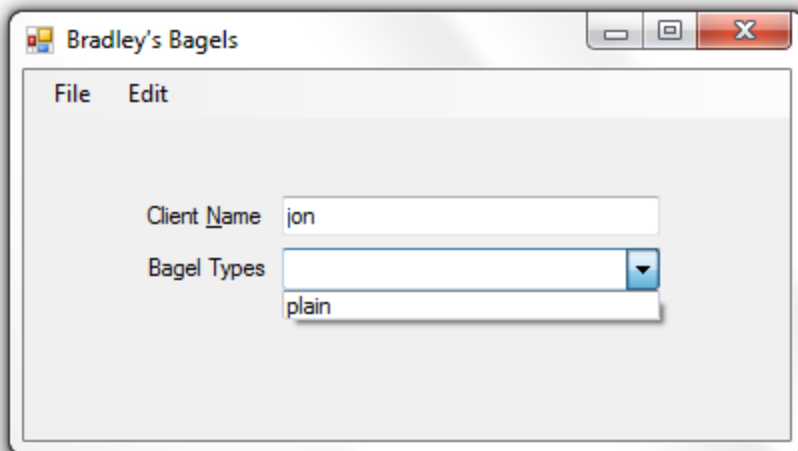
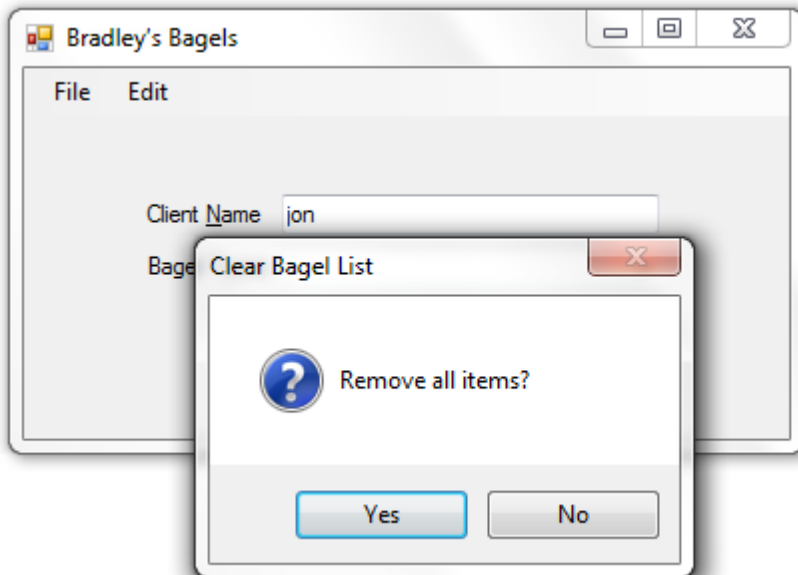
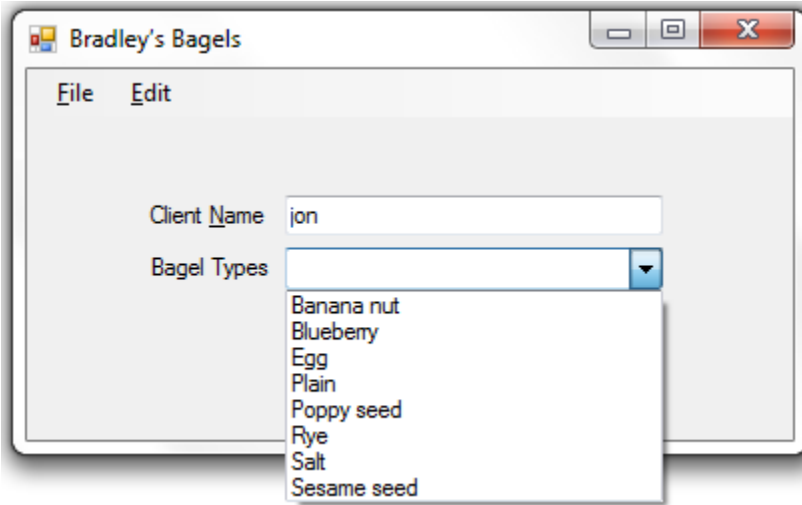
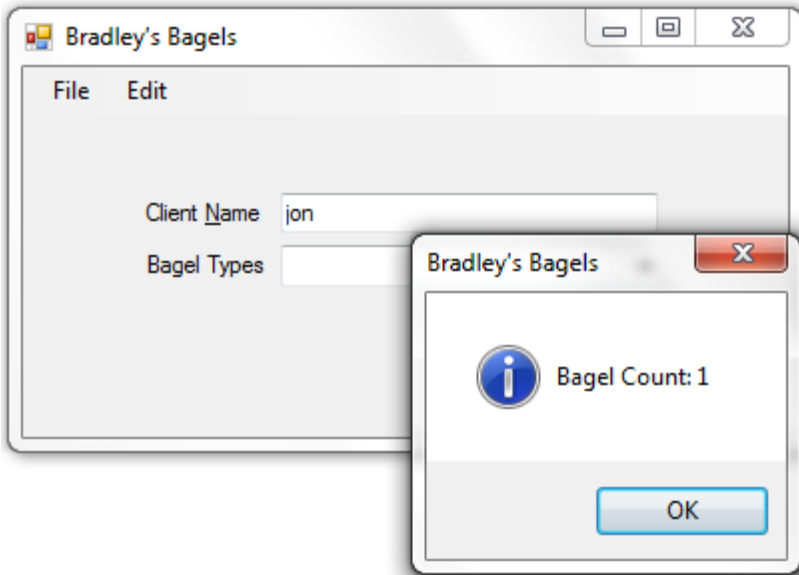


# Lab 7 by Bryant Tunbutr





## Source code for bagelsForm.cs

```
/*
 * Project:           BTunbutrLab7
 * Programmer:        Bradley/Millspaugh
 * Date:              October 18
 * Description:        Maintain a list of bagels, print the selected
 *                     bagel or print a list of all bagels.
 * I certify that the code below is my own work.
 */

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace Ch07HandsOn
{
    public partial class bagelsForm : Form
    {
        public bagelsForm()
        {
            InitializeComponent();
        }

        private void printSelectedBagelToolStripMenuItem_Click(object sender,
            EventArgs e)
        {
            // Print preview for selected Bagel.

            if (bagelsComboBox.SelectedIndex == -1)
            {
                MessageBox.Show("Select a bagel to print.", "No Selection");
            }
            else
            {
                printPreviewDialog1.Document = printSelectedPrintDocument;
                printPreviewDialog1.ShowDialog();
            }
        }

        private void printAllBagelsToolStripMenuItem_Click(object sender,
            EventArgs e)
        {
            // Print preview for all Bagels.

            printPreviewDialog1.Document = printAllPrintDocument;
            printPreviewDialog1.ShowDialog();
        }

        private void exitToolStripMenuItem_Click(object sender, EventArgs e)

```

```

{
    // Close the application.

    this.Close();
}

private void addABagelToolStripMenuItem_Click(object sender,
EventArgs e)
{
    // Add a new Bagel.
    int indexInteger = 0;
    bool itemFoundBoolean = false;

    // Use a loop to test for duplicates.
    if (bagelsComboBox.Text != string.Empty)
    {
        while (indexInteger < bagelsComboBox.Items.Count &&
!itemFoundBoolean)
        {
            if (bagelsComboBox.Items[indexInteger++].ToString().ToUpper()
== bagelsComboBox.Text.ToUpper())
            {
                MessageBox.Show("Duplicate class name.", "Bagel Not
Added",
                MessageBoxButtons.OK, MessageBoxIcon.Information);
                itemFoundBoolean = true;
            }
        }
        if (!itemFoundBoolean)
        {
            // Add to the list.
            bagelsComboBox.Items.Add(bagelsComboBox.Text);
            bagelsComboBox.Text = string.Empty;
        }
    }
    else
    {
        MessageBox.Show("Enter the new bagel name.",
            "No Bagel Name Entered", MessageBoxButtons.OK,
            MessageBoxIcon.Exclamation);
    }
}

private void removeABagelToolStripMenuItem_Click(object sender,
EventArgs e)
{
    // Remove the selected Bagel.

    if (bagelsComboBox.SelectedIndex != -1)
    {
        bagelsComboBox.Items.RemoveAt(bagelsComboBox.SelectedIndex);
    }
    else
    {
        MessageBox.Show("Select a bagel to remove.", "No Selection",
            MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}

```

```

    }

    private void clearTheBagelListToolStripMenuItem_Click(object sender,
        EventArgs e)
    {
        // Verify and then remove the Bagel.
        DialogResult confirmDialogResult = MessageBox.Show("Remove all
items?",
        "Clear Bagel List", MessageBoxButtons.YesNo,
        MessageBoxIcon.Question);
        if (confirmDialogResult == DialogResult.Yes)
        {
            bagelsComboBox.Items.Clear();
        }
    }

    private void displayBagelCountToolStripMenuItem_Click(object sender,
        EventArgs e)
    {
        // Display the number of Bagel.

        MessageBox.Show("Bagel Count: " +
            bagelsComboBox.Items.Count.ToString(),
            "Bradley's Bagels", MessageBoxButtons.OK,
            MessageBoxIcon.Information);
    }

    private void printAllPrintDocument_PrintPage(object sender,
        System.Drawing.Printing.PrintPageEventArgs e)
    {
        // Print a list of all current courses.
        Font printFont = new Font("Arial", 10);
        Font headingFont = new Font("Arial", 14, FontStyle.Bold);
        float verticalPrintPositionFloat = e.MarginBounds.Top;
        float horizontalPrintPositionFloat = e.MarginBounds.Left;
        float lineHeightFloat = printFont.GetHeight();

        // Print heading.
        e.Graphics.DrawString("Current Bagel List As of "
            + DateTime.Now.ToShortDateString(), headingFont,
            Brushes.Black, horizontalPrintPositionFloat,
            verticalPrintPositionFloat);
        verticalPrintPositionFloat += 2 * lineHeightFloat;

        // Loop through the list to print all courses.
        for (int indexInteger = 0; indexInteger < bagelsComboBox.Items.Count;
            indexInteger++)
        {
            e.Graphics.DrawString(bagelsComboBox.Items[indexInteger].ToString(),
                printFont, Brushes.Black, horizontalPrintPositionFloat,
                verticalPrintPositionFloat);
            verticalPrintPositionFloat += lineHeightFloat;
        }
    }

    private void printSelectedPrintDocument_PrintPage(object sender,

```

```
System.Drawing.Printing.PrintPageEventArgs e)
{
    // Set up output for selected item.
    Font printFont = new Font("Arial", 12);

    // Print heading.
    e.Graphics.DrawString("Selected Bagel:", printFont, Brushes.Black,
        100, 100);
    e.Graphics.DrawString(nameTextBox.Text + " \t" +
        bagelsComboBox.SelectedItem.ToString(), printFont, Brushes.Black,
        100, 140);
}
}
```

If a drop-down combo box is not available, can we use a list box for this exercise? Explain why or why not.

Yes because there are only a few items. If the menu had 100 items, a drop-down combo box would be much better.