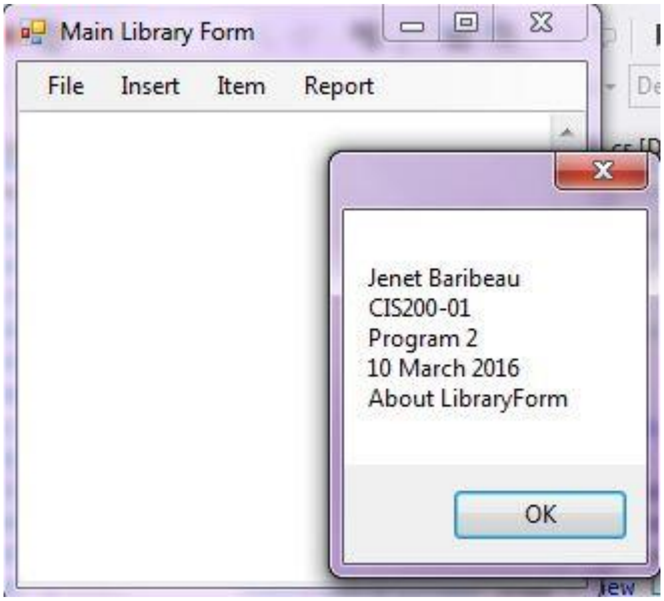
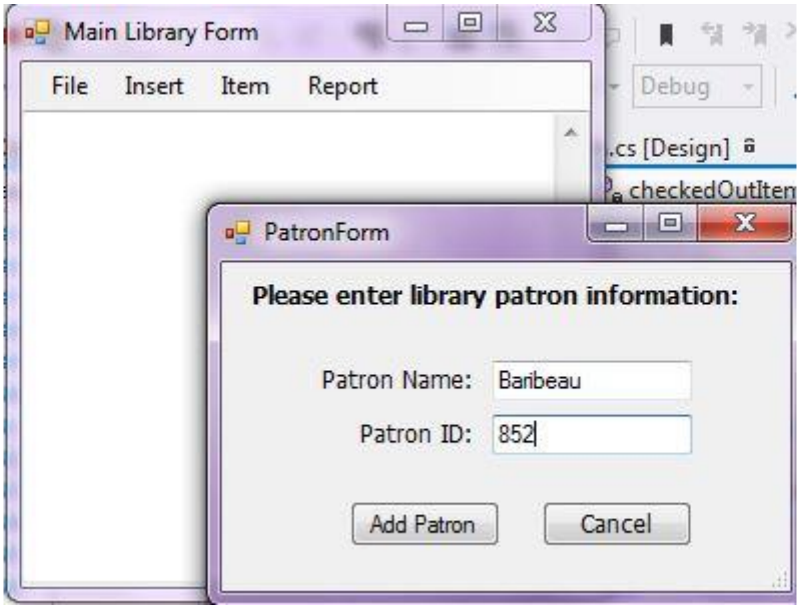
```
//Jenet Baribeau
//CIS200-01
//10 March 2016

//Program 2
// File: MainForm.cs
// This file operates a basic Library that takes responsibility for the operations
//of the different forms in order to check out, report, add patrons and items as well as
return items.
// classes in the same namespace (LibraryItems)
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace LibraryItems
{
    public partial class MainForm : Form
    {
        private Library mainLib = new Library();//initialize library prior to form
loading.

        public MainForm()
        {
            InitializeComponent();

            //Test Data
            mainLib.AddLibraryBook("The Wright Guide to C#", "UofL Press", 2010, 11,
"ZZ25 3G", "Andrew Wright");
            mainLib.AddLibraryBook("Harriet Pooter", "Stealer Books", 2000, 21, "AG773
ZF", "IP Thief");
            mainLib.AddLibraryBook("The Color Mauve", "Beautiful Books Ltd.", 1985, 14,
"JJ438 4T", "Mary Smith");
            mainLib.AddPatron("John Voight", "55555");
            mainLib.AddPatron("Brad Pitt", "77777");
            mainLib.AddPatron("Ryan Reynolds", "99999");
        }

        //Precondition: About item initiated from menu
        //Postcondition: Show information about the program
        private void aboutToolStripMenuItem_Click(object sender, EventArgs e)
        {
            MessageBox.Show(String.Format("Jenet Baribeau{0}CIS200-01{0}Program 2{0}10
March 2016{0}About LibraryForm",
                System.Environment.NewLine));
        }
        //Precondition: Exit item initiated from menu.
        //Postcondition: Application is closed.
        private void exitToolStripMenuItem_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }
        //Precondition: Patron List chosen from Report item on Menu Tool
        //Postcondition: List of patrons displayed in reportTxtbx in MainForm.
        private void patronListToolStripMenuItem_Click(object sender, EventArgs e)
        {
            reportTxtbx.Clear();

            StringBuilder report = new StringBuilder();//Stringbuilder used to hold data 
as it formats
            report.Append(String.Format("Patron Report {0}", 
System.Environment.NewLine));
            report.Append(String.Format("-------------{0}", System.Environment.NewLine));
            report.Append(String.Format("Count:     {0,4:d}{1}{1}", 
mainLib._patrons.Count(),System.Environment.NewLine));

            foreach(LibraryPatron p in mainLib._patrons)
            {
                report.Append(p.ToString());
                report.Append(System.Environment.NewLine);
                report.Append(System.Environment.NewLine);
            }

            reportTxtbx.Text = report.ToString();

        }
        // Precondition:  The Patron Menu item has been clicked
        // Postcondition: The PatronForm has been displayed prompting for
        //                a patron and patronID. If user submits, this will add
        //                a patron to the library.
        private void patronToolStripMenuItem_Click(object sender, EventArgs e)
        {
            PatronForm patronForm = new PatronForm();
            DialogResult result;

            result = patronForm.ShowDialog();

            if (result == DialogResult.OK)
            {
                LibraryPatron newPatron = new LibraryPatron(patronForm.PatronName, 
patronForm.PatronID);
                mainLib._patrons.Add(newPatron);
            }
        }
        // Precondition:  The Book Menu item has been clicked
        // Postcondition: The BookForm has been displayed prompting for
        //                book information to be entered. If user submits, this will add
        //                a book to the library.
        private void bookToolStripMenuItem_Click(object sender, EventArgs e)
        {
            BookForm bookForm = new BookForm();
            DialogResult result;

            result = bookForm.ShowDialog();

            if (result == DialogResult.OK)
            {
                try
                {
                    mainLib.AddLibraryBook(bookForm.Title, bookForm.Publisher, 
int.Parse(bookForm.CopyrightYear),
                            int.Parse(bookForm.LoanPeriod), bookForm.CallNumber,
bookForm.Author);
                }
                catch(Exception)
                {
                    MessageBox.Show("Please re-evaluate entries.");
                }
            }
        }
        // Precondition:  The Report, Item list has been clicked
        // Postcondition: The report box shows all the books added to the list.
        private void itemListToolStripMenuItem_Click(object sender, EventArgs e)
        {
            reportTxtbx.Clear();

            StringBuilder report = new StringBuilder();//Stringbuilder used to hold data
as it formats
            report.Append(String.Format("Book Report {0}", System.Environment.NewLine));
            report.Append(String.Format("-----------{0}", System.Environment.NewLine));
            report.Append(String.Format("Count:     {0,4:d}{1}{1}",
mainLib._items.Count(), System.Environment.NewLine));

            foreach (LibraryBook b in mainLib._items)
            {
                report.Append(b.ToString());
                report.Append(System.Environment.NewLine);
                report.Append(System.Environment.NewLine);
            }

            reportTxtbx.Text = report.ToString();
        }
        //Precondition: Report, check out  menu item was clicked
        //Postcondition: The list of checked out items is shown in the ReportTxtbx
        private void checkedOutItemsToolStripMenuItem_Click(object sender, EventArgs e)
        {
            reportTxtbx.Clear();//Clear the textbox


            StringBuilder ckOutReport = new StringBuilder();

            ckOutReport.Append(String.Format("Checked Out Report {0}",
System.Environment.NewLine));
            ckOutReport.Append(String.Format("------------------{0}",
System.Environment.NewLine));
            ckOutReport.Append(String.Format("Count:     {0,4:d}{1}{1}",
mainLib._items.Count(), System.Environment.NewLine));
             List<LibraryItem> item = new List<LibraryItem>();
            foreach (LibraryBook b in mainLib._items)
            {
                ckOutReport.Append(b.ToString());
                ckOutReport.Append(System.Environment.NewLine);
                ckOutReport.Append(System.Environment.NewLine);
            }

            reportTxtbx.Text = ckOutReport.ToString();

        }
        private void checkOutToolStripMenuItem_Click(object sender, EventArgs e)
        {
            List<LibraryItem> notCkOutList;
            List<int> notCkOutIndex;

            notCkOutList = new List<LibraryItem>();
            notCkOutIndex = new List<int>();

            for (int i = 0; i < mainLib._items.Count(); ++i)
                if (!mainLib._items[i].IsCheckedOut())
                {
                    notCkOutList.Add(mainLib._items[i]);
                    notCkOutIndex.Add(i);
                }
            if ((notCkOutList.Count() == 0) || (mainLib._patrons.Count() == 0))
                MessageBox.Show("Please choose both items and patrons for check out.");
            else
            {
                CheckOutForm checkOutForm = new CheckOutForm(notCkOutList,
mainLib._patrons);
                DialogResult result = checkOutForm.ShowDialog();

                if(result == DialogResult.OK)
                    try
                    {
                        int itemIndex;
                        itemIndex = notCkOutIndex[checkOutForm.ItemsIndex];
                        mainLib.CheckOut(itemIndex, checkOutForm.PatronsIndex);
                    }
                  catch (ArgumentOutOfRangeException)
                    {
                        MessageBox.Show("Please refer to the Check out List.");
                    }
            }
        }

        private void returnToolStripMenuItem_Click(object sender, EventArgs e)
        {
            List<LibraryItem> checkedOutList;
            List<int> checkedOutIndex;
            checkedOutList = new List<LibraryItem>();
            checkedOutIndex = new List<int>();

            for (int i = 0; i < mainLib._items.Count(); ++i)
                if (mainLib._items[i].IsCheckedOut())
                {
                    checkedOutList.Add(mainLib._items[i]);
                    checkedOutIndex.Add(i);
                }
            if (checkedOutList.Count() == 0)
            {
                MessageBox.Show("No items to return");
            }
            else
            {
                ReturnForm returnForm = new ReturnForm(checkedOutList);
                DialogResult result = returnForm.ShowDialog();
```

```
if (result == DialogResult.OK)
{
    int itemsIndex;
    itemsIndex = checkedOutIndex[returnForm.ItemsIndex];
    mainLib.ReturnToShelf(itemsIndex);
```



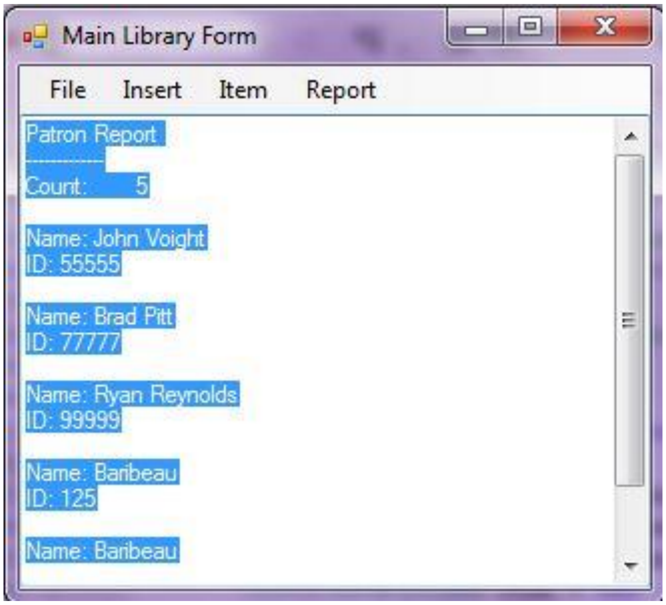
Jenet Baribeau
CIS200-01
Program 2
10 March 2016
About LibraryForm



Please enter library patron information:

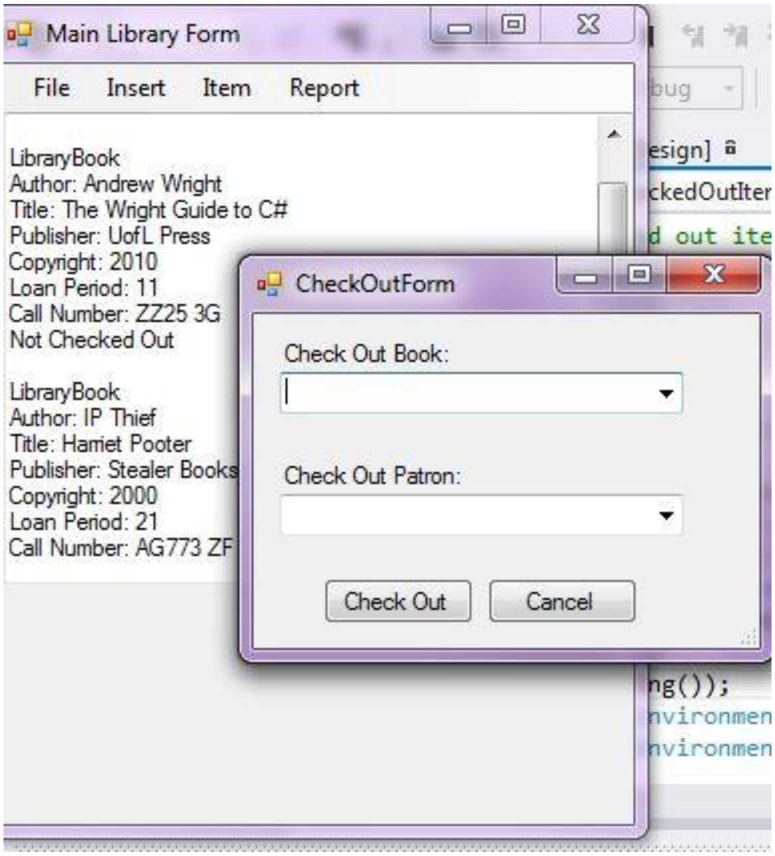
Patron Name: Baribeau

Patron ID: 852

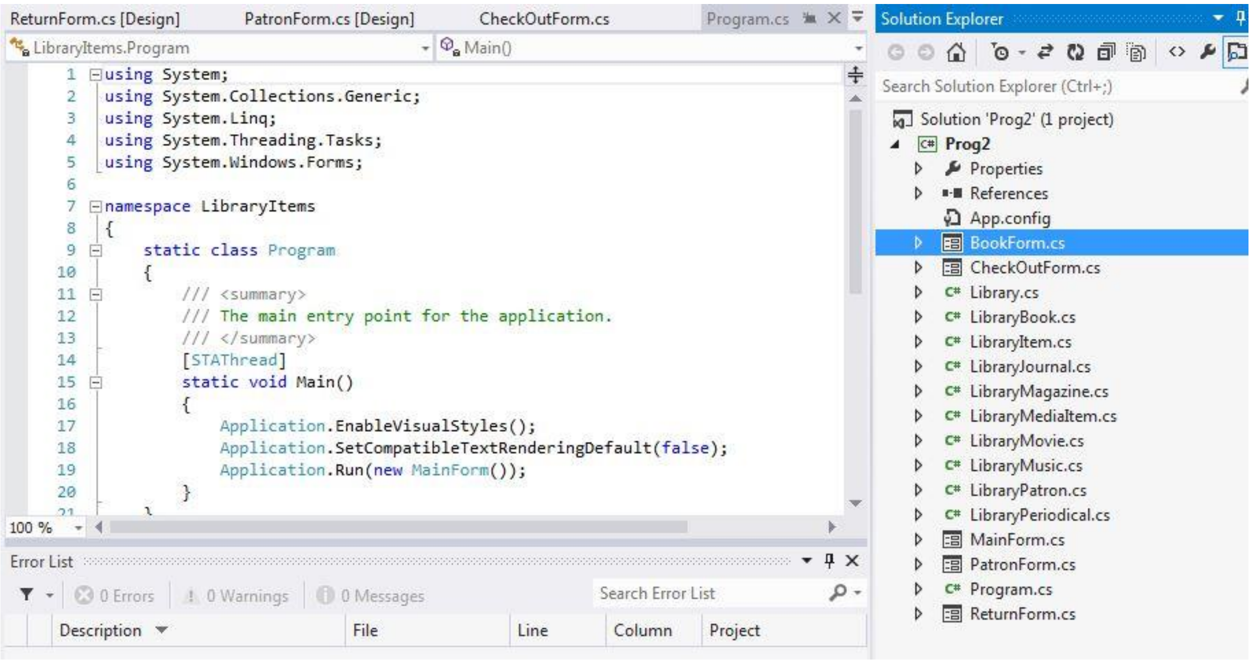


```
Patron Report
------------
Count:        5

Name: John Voight
ID: 55555

Name: Brad Pitt
ID: 77777

Name: Ryan Reynolds
ID: 99999

Name: Baribeau
ID: 125

Name: Baribeau
```

```
}
```



```
LibraryBook
Author: Andrew Wright
Title: The Wright Guide to C#
Publisher: UofL Press
Copyright: 2010
Loan Period: 11
Call Number: ZZ25 3G
Not Checked Out

LibraryBook
Author: IP Thief
Title: Harriet Pooter
Publisher: Stealer Books
Copyright: 2000
Loan Period: 21
Call Number: AG773 ZF
```



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace LibraryItems
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new MainForm());
        }
    }
```