BIS 628 – Application Development

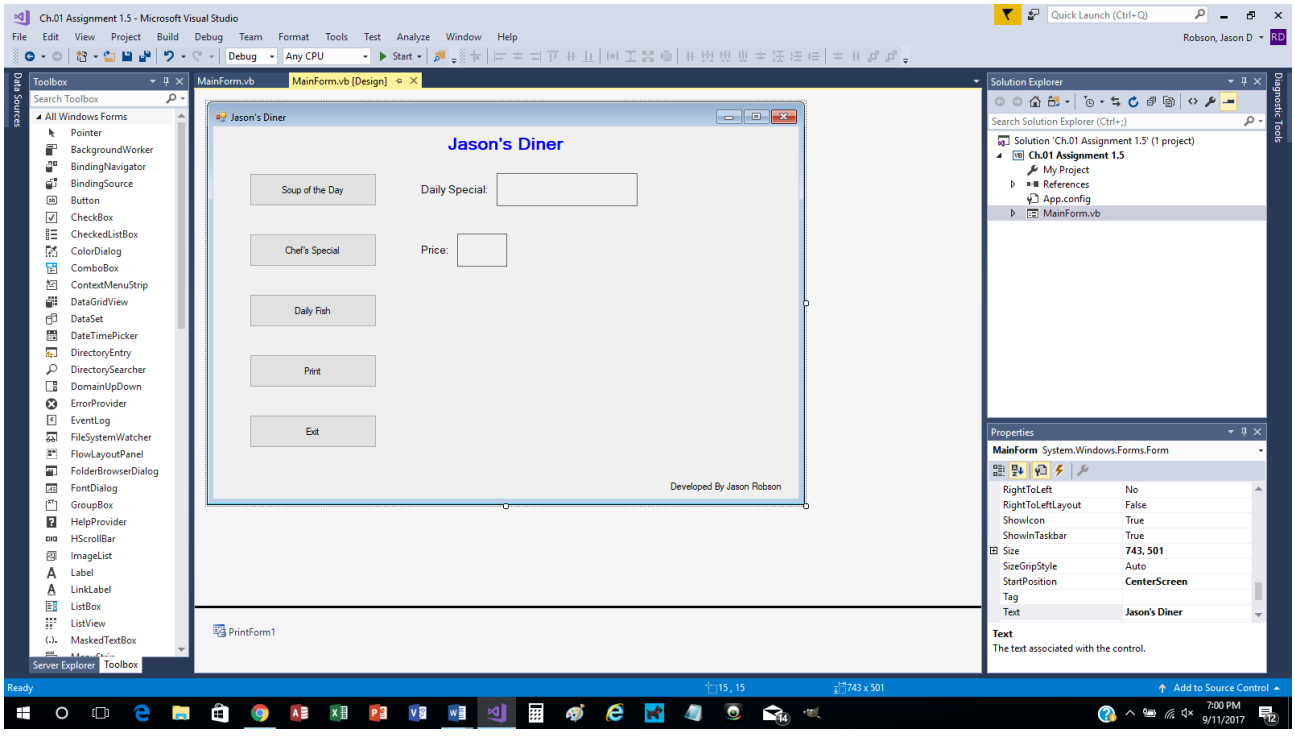
Assignment 1

- Example 1.5 (page 60)

Due Date: 09/13/2017 11:59 pm

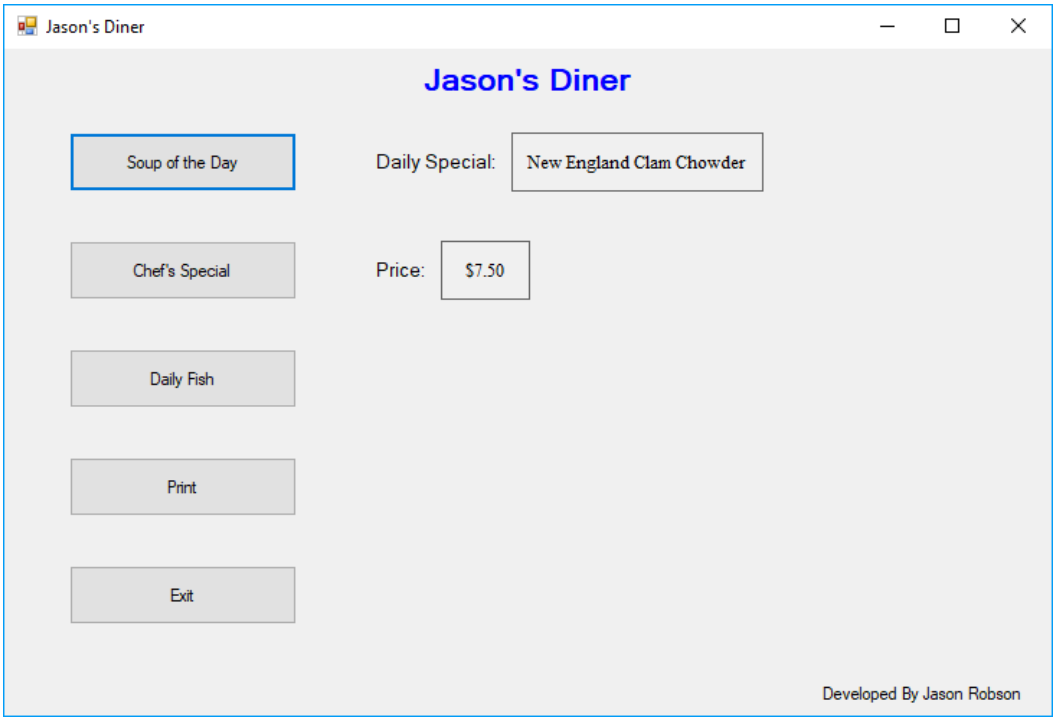
By: Jason Robson

1. Design View



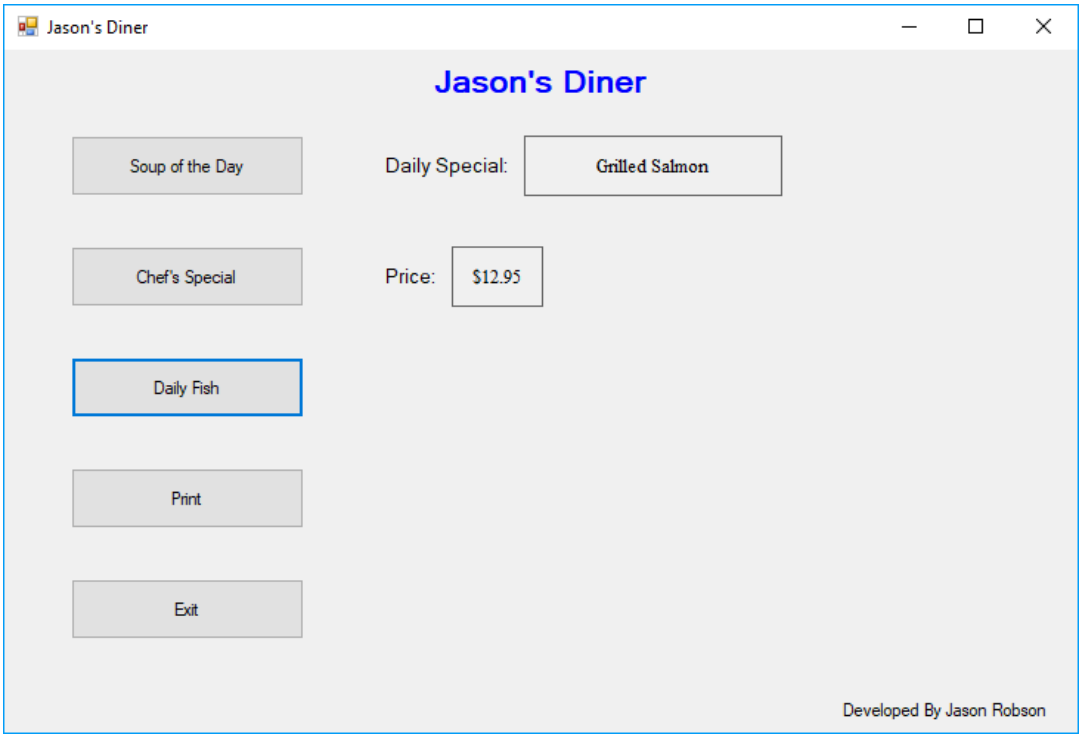
2. Execution Views

2.1 Soup of the Day (“Soup of the Day” button is clicked)

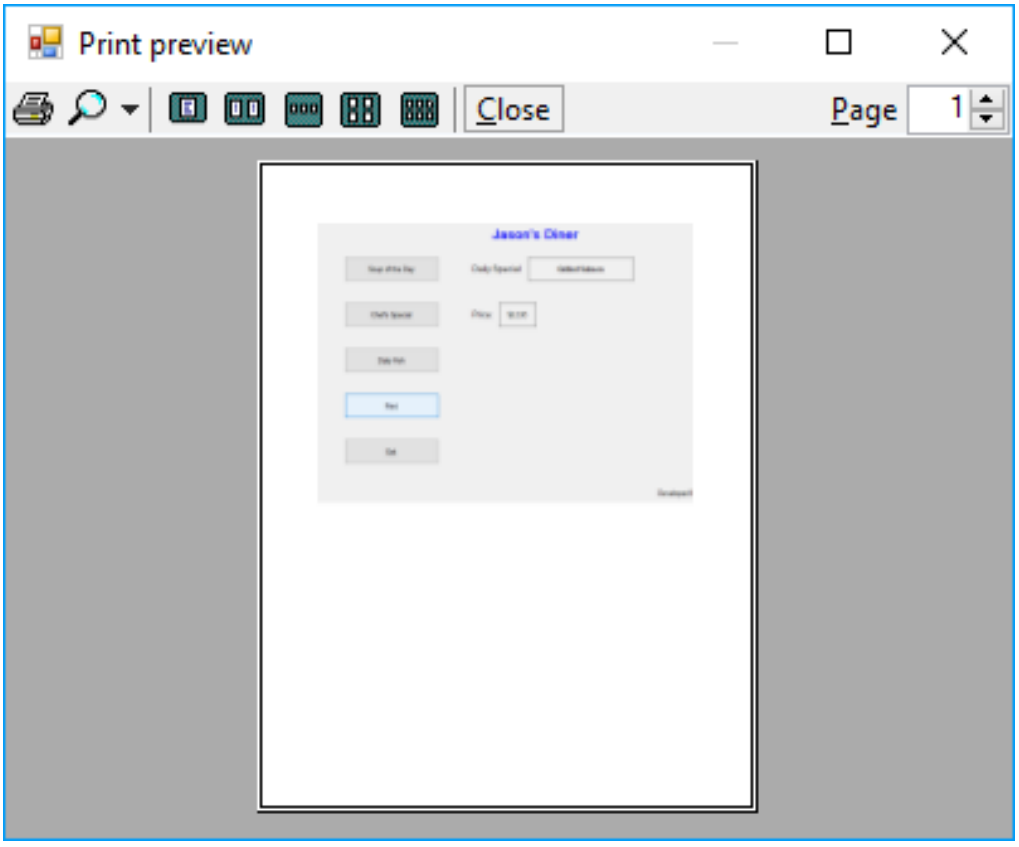


2.2 Chef’s Special (“Chef’s Special” button is clicked)

2.3 Daily Fish (“Daily Fish” button is clicked)



2.4 Print (“Print” button is clicked)



3. Program Code

```
'Project     : Jason’sDiner
'Programmer  : Jason Robson
'Date        : September 11, 2017
'Description : This project displays daily specials and prices for my diner

Public Class MainForm
    Private Sub SoupoftheDayButton_Click(sender As Object, e As EventArgs) Handles
SoupoftheDayButton.Click
        'Display Soup of the Day and price
        DailySpecialLabel.Text = "New England Clam Chowder"
        PriceLabel.Text = "$7.50"
    End Sub

    Private Sub ChefSpecialButton_Click(sender As Object, e As EventArgs) Handles
ChefSpecialButton.Click
        'Display Chef's Special and price
        DailySpecialLabel.Text = "Roasted Duck"
```

```
        PriceLabel.Text = "$14.95"
    End Sub

    Private Sub DailyFishButton_Click(sender As Object, e As EventArgs) Handles
DailyFishButton.Click
        'Display Daily Fish and price
        DailySpecialLabel.Text = "Grilled Salmon"
        PriceLabel.Text = "$12.95"
    End Sub

    Private Sub PrintButton_Click(sender As Object, e As EventArgs) Handles
PrintButton.Click
        'Print the form on the printer
        PrintForm1.PrintAction = Printing.PrintAction.PrintToPreview
        PrintForm1.Print()
    End Sub

    Private Sub ExitButton_Click(sender As Object, e As EventArgs) Handles
ExitButton.Click
        'End the program
        Me.Close()
    End Sub
End Class
```

BIS 628 – Application Development

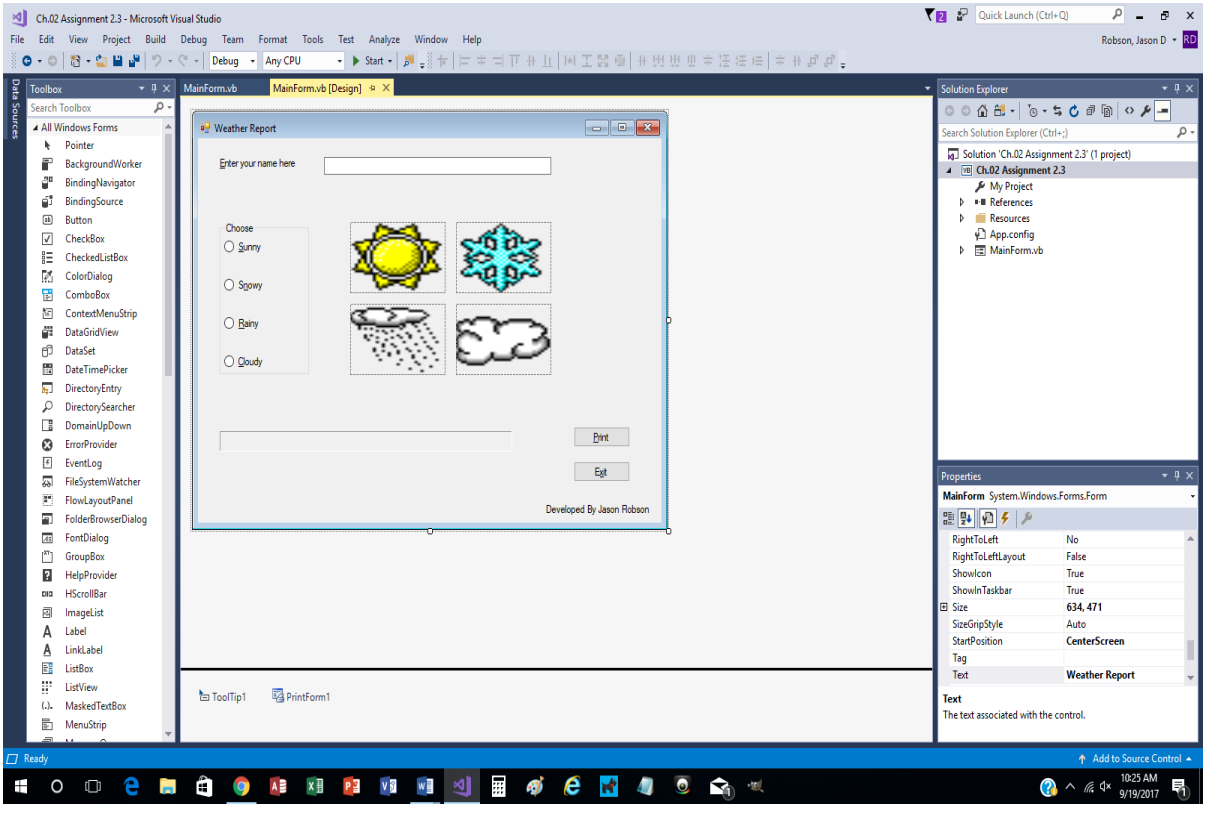
Assignment 2

- Example 2.3 (page 100)

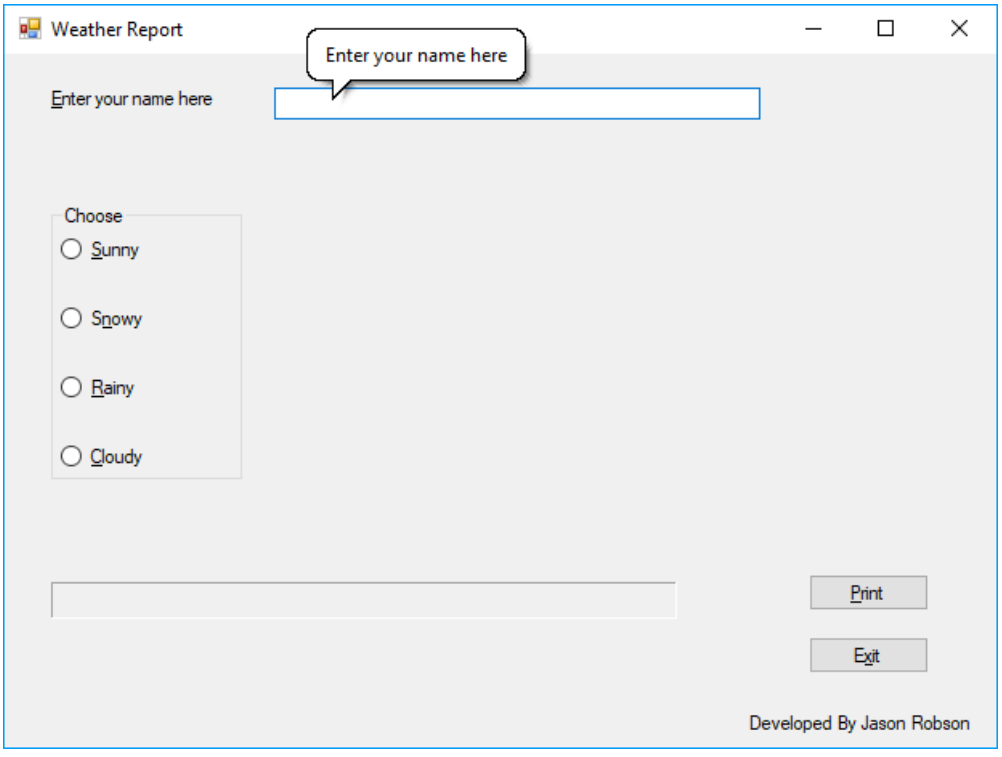
Due Date: 09/20/2017 11:59 pm

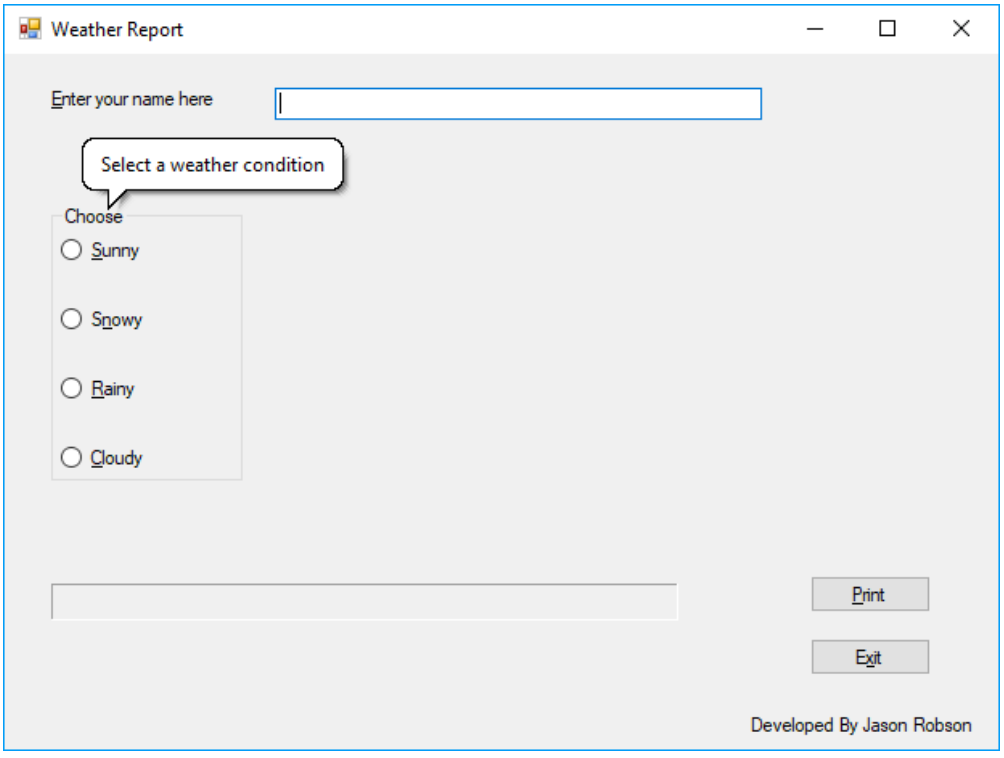
By: Jason Robson

1. Design View



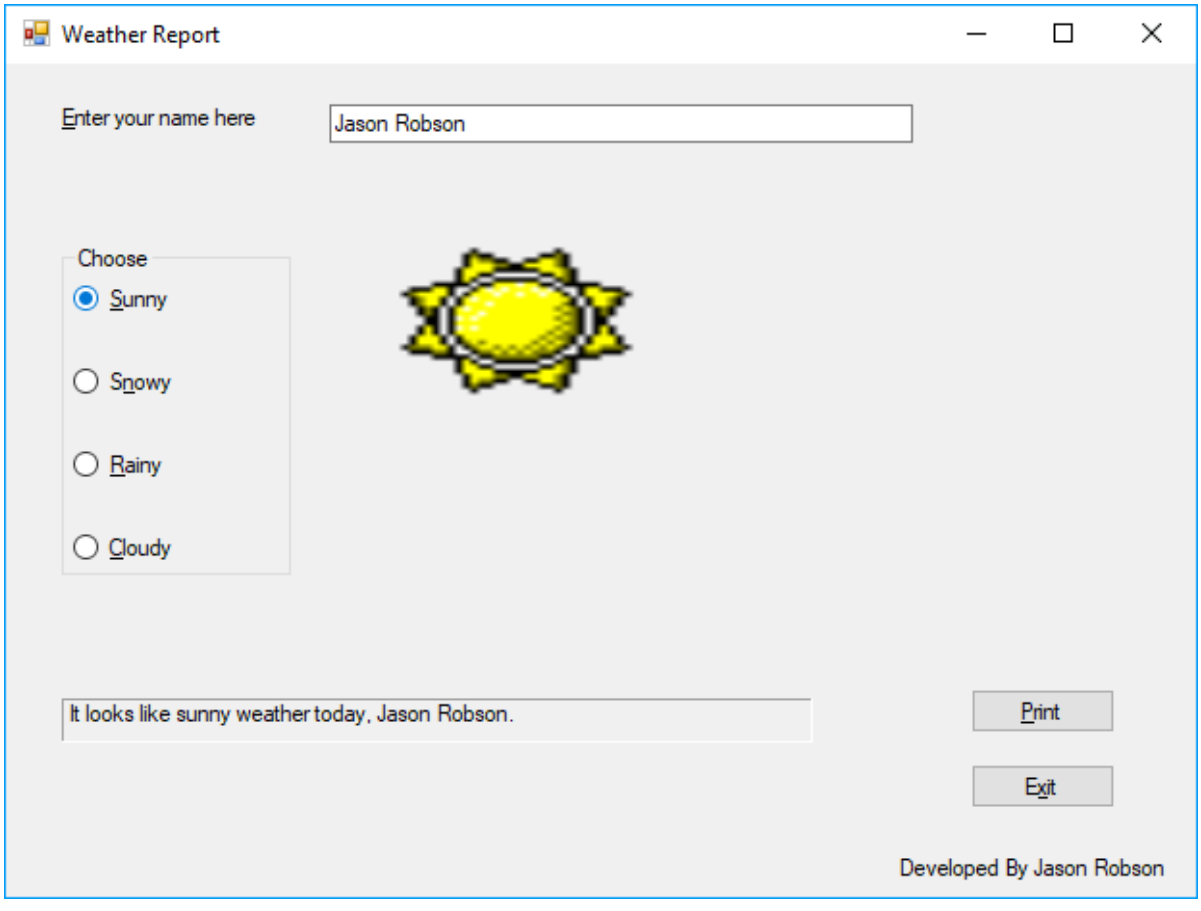
2. ToolTips


Weather Report
Enter your name here
Enter your name here
Choose
Sunny
Snowy
Rainy
Cloudy
Print
Exit
Developed By Jason Robson

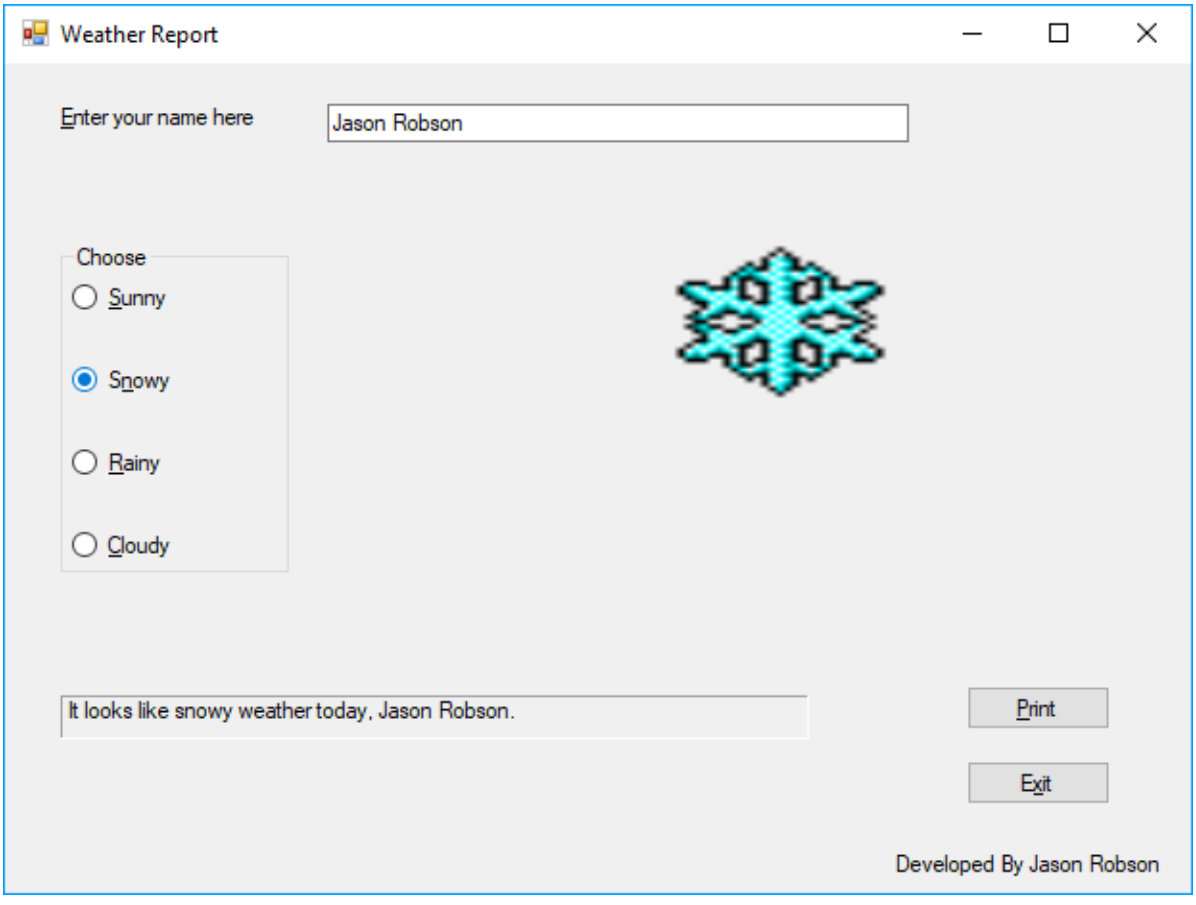

Weather Report
Enter your name here
Select a weather condition
Choose
Sunny
Snowy
Rainy
Cloudy
Print
Exit
Developed By Jason Robson

3. Execution Views

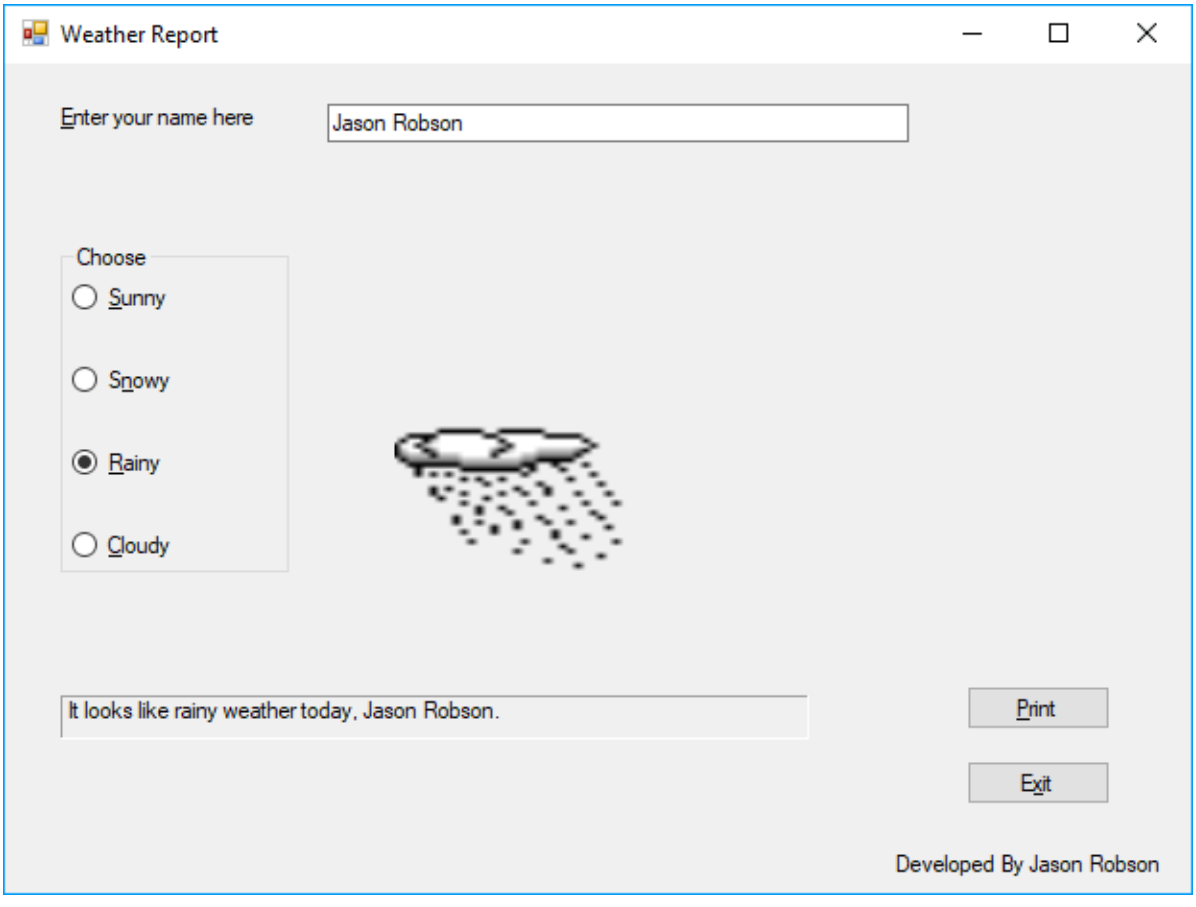
3.1 Sunny (“Sunny Radio Button” is checked)



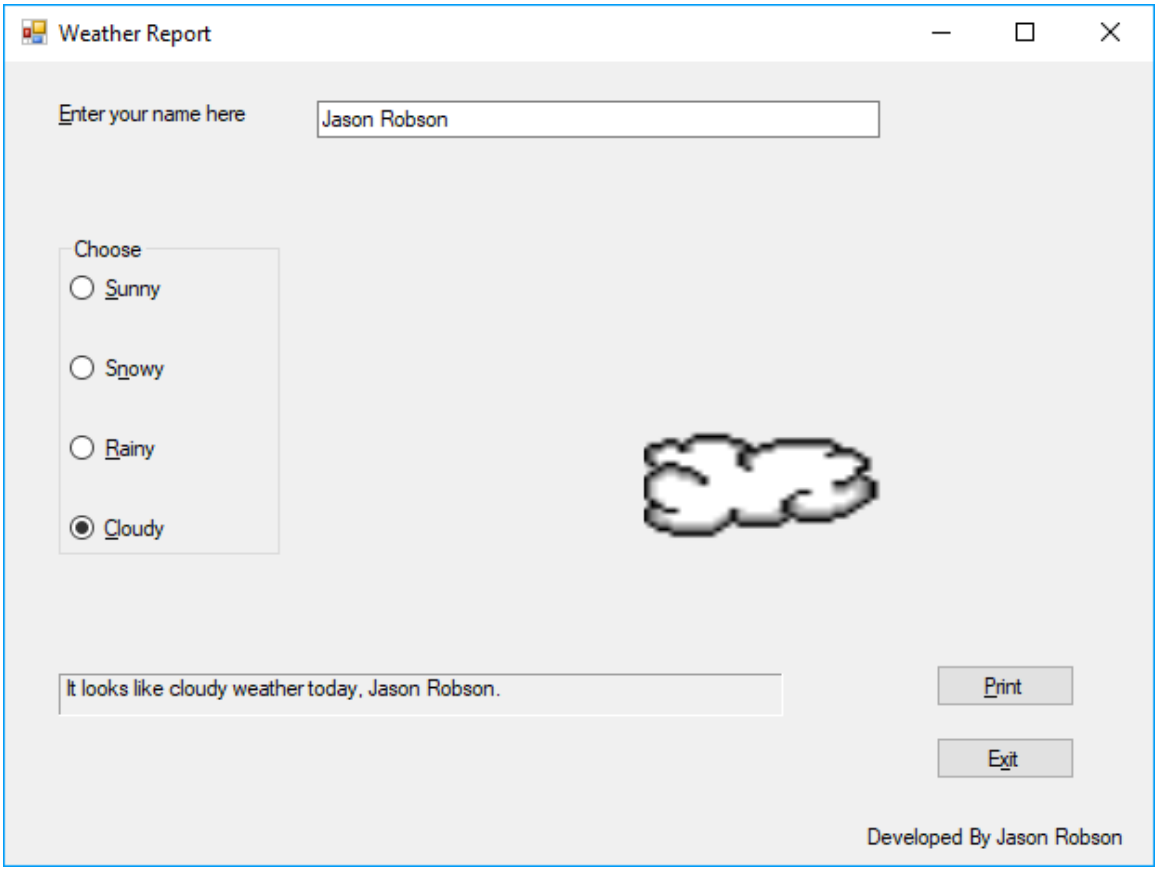
3.2 Snowy (“Snowy Radio Button” is clicked)



3.3 Rainy (“Rainy Radio Button” is clicked)



3.4 Cloudy (“Cloudy Radio Button” is clicked)



4. Program Code

```
'Project     : WeatherReport
'Programmer  : Jason Robson
'Date        : September 17, 2017
'Description : This project displays a weather report message with user name and a
weather image based on radio button selected

Public Class MainForm
    Private Sub SunnyRadioButton_CheckedChanged(sender As Object, e As EventArgs)
Handles SunnyRadioButton.CheckedChanged
        'Display sun picture and message containing user name input
        SunnyPictureBox.Visible = SunnyRadioButton.Checked
        MessageLabel.Text = "It looks like sunny weather today, " &
NameTextBox.Text & "."
    End Sub

    Private Sub SnowyRadioButton_CheckedChanged(sender As Object, e As EventArgs)
Handles SnowyRadioButton.CheckedChanged
        'Display snow picture and message containing user name input
        SnowyPictureBox.Visible = SnowyRadioButton.Checked
        MessageLabel.Text = "It looks like snowy weather today, " &
NameTextBox.Text & "."
    End Sub

    Private Sub RainyRadioButton_CheckedChanged(sender As Object, e As EventArgs)
Handles RainyRadioButton.CheckedChanged
        'Display rain picture and message containing user name input
        RainyPictureBox.Visible = RainyRadioButton.Checked
        MessageLabel.Text = "It looks like rainy weather today, " &
NameTextBox.Text & "."
    End Sub

    Private Sub CloudyRadioButton_CheckedChanged(sender As Object, e As EventArgs)
Handles CloudyRadioButton.CheckedChanged
        'Display cloud picture and message containing user name input
        CloudyPictureBox.Visible = CloudyRadioButton.Checked
        MessageLabel.Text = "It looks like cloudy weather today, " &
NameTextBox.Text & "."
    End Sub

    Private Sub PrintButton_Click(sender As Object, e As EventArgs) Handles
PrintButton.Click
        'Print the form
        PrintForm1.PrintAction = Printing.PrintAction.PrintToPreview
        PrintForm1.Print()
    End Sub

    Private Sub ExitButton_Click(sender As Object, e As EventArgs) Handles
ExitButton.Click
        'Close the program
        Me.Close()
    End Sub
End Class
```

BIS 628 – Application Development

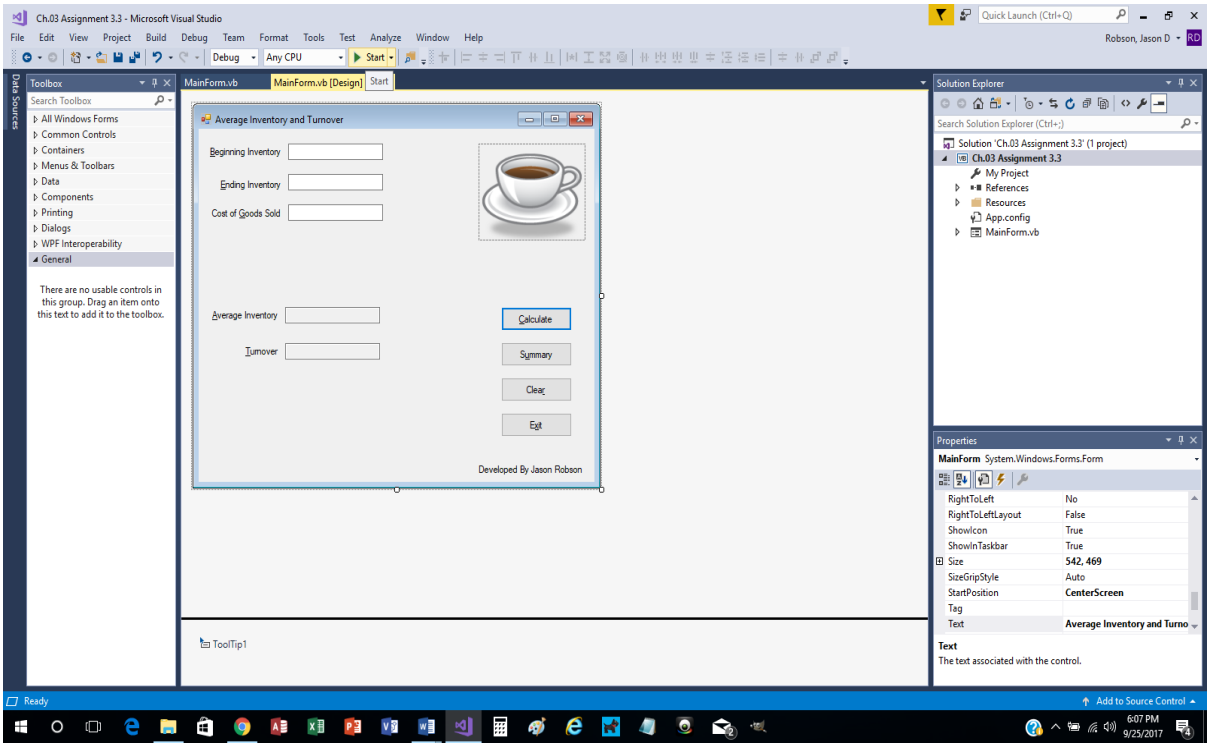
Assignment 3

- Example 3.3 (page 150)

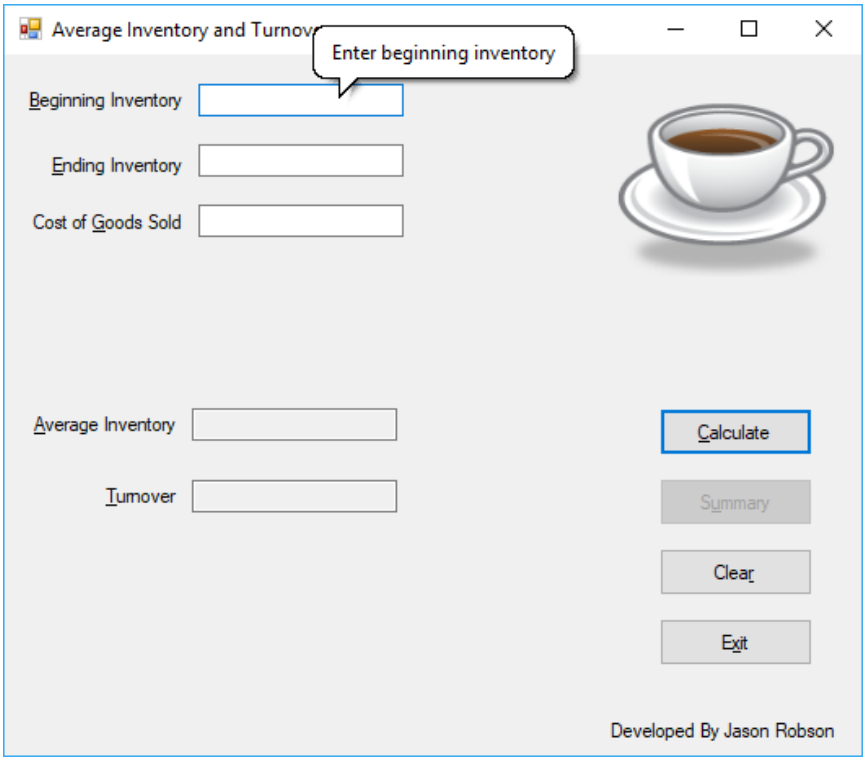
Due Date: 09/27/2017 11:59 pm

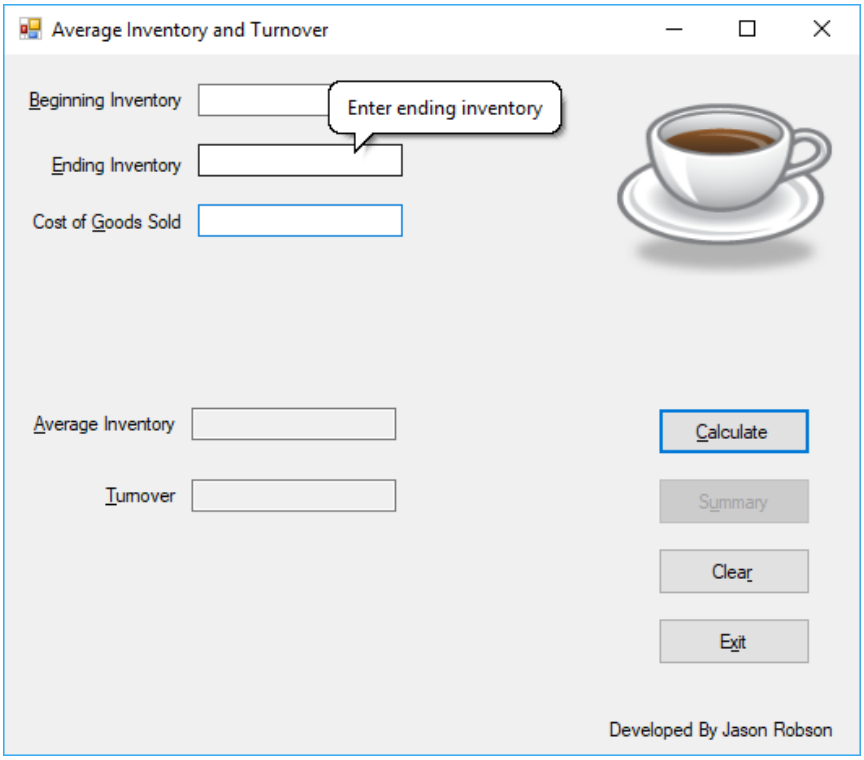
By: Jason Robson

1. Design View

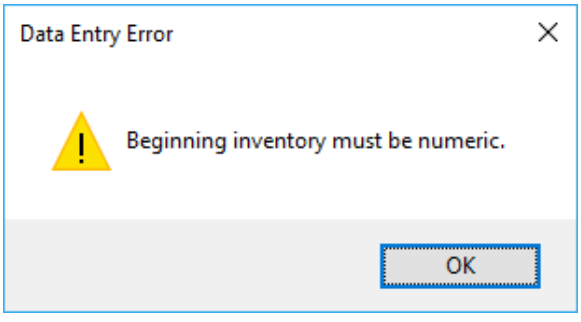


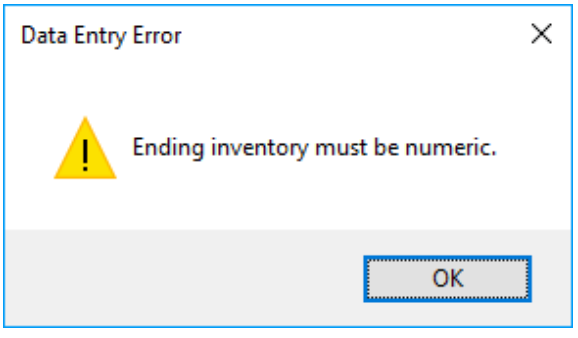
2. ToolTips

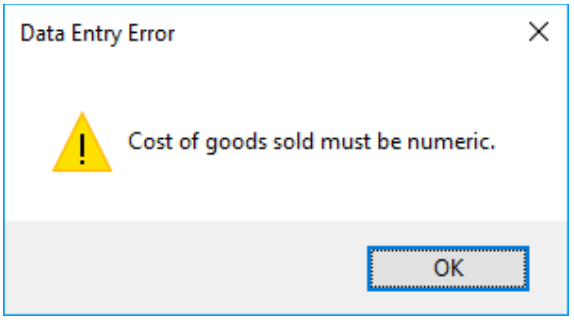

Average Inventory and Turnov
Enter beginning inventory
Beginning Inventory
Ending Inventory
Cost of Goods Sold
Average Inventory
Turnover
Calculate
Summary
Clear
Exit
Developed By Jason Robson

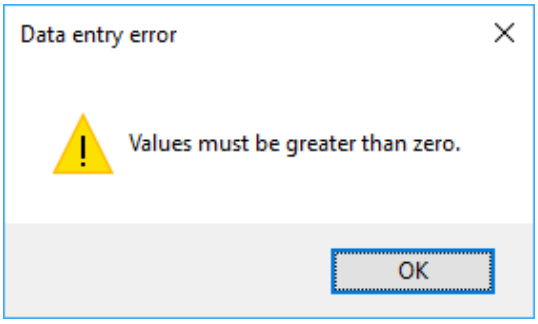

Average Inventory and Turnover
Beginning Inventory
Enter ending inventory
Ending Inventory
Cost of Goods Sold
Average Inventory
Turnover
Calculate
Summary
Clear
Exit
Developed By Jason Robson

3. Error messages (Non- Numeric Data Entered Or Value Is Zero)

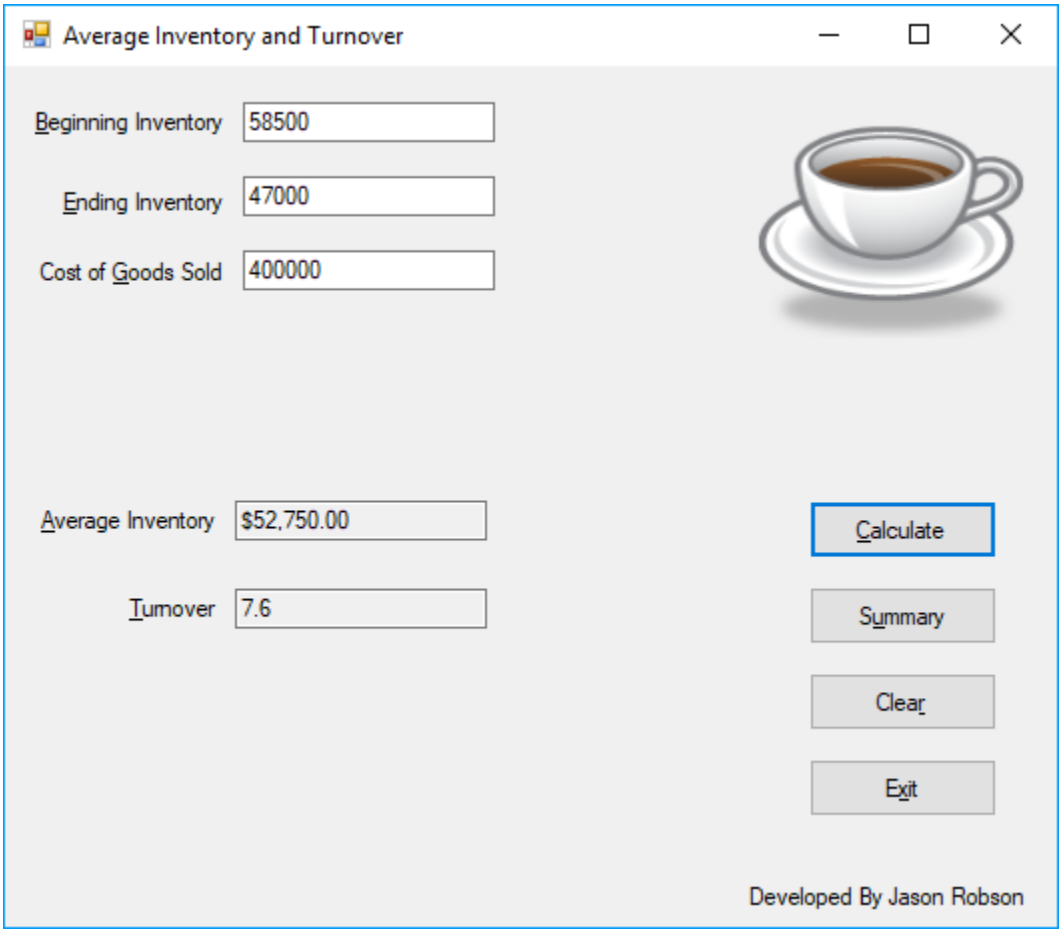




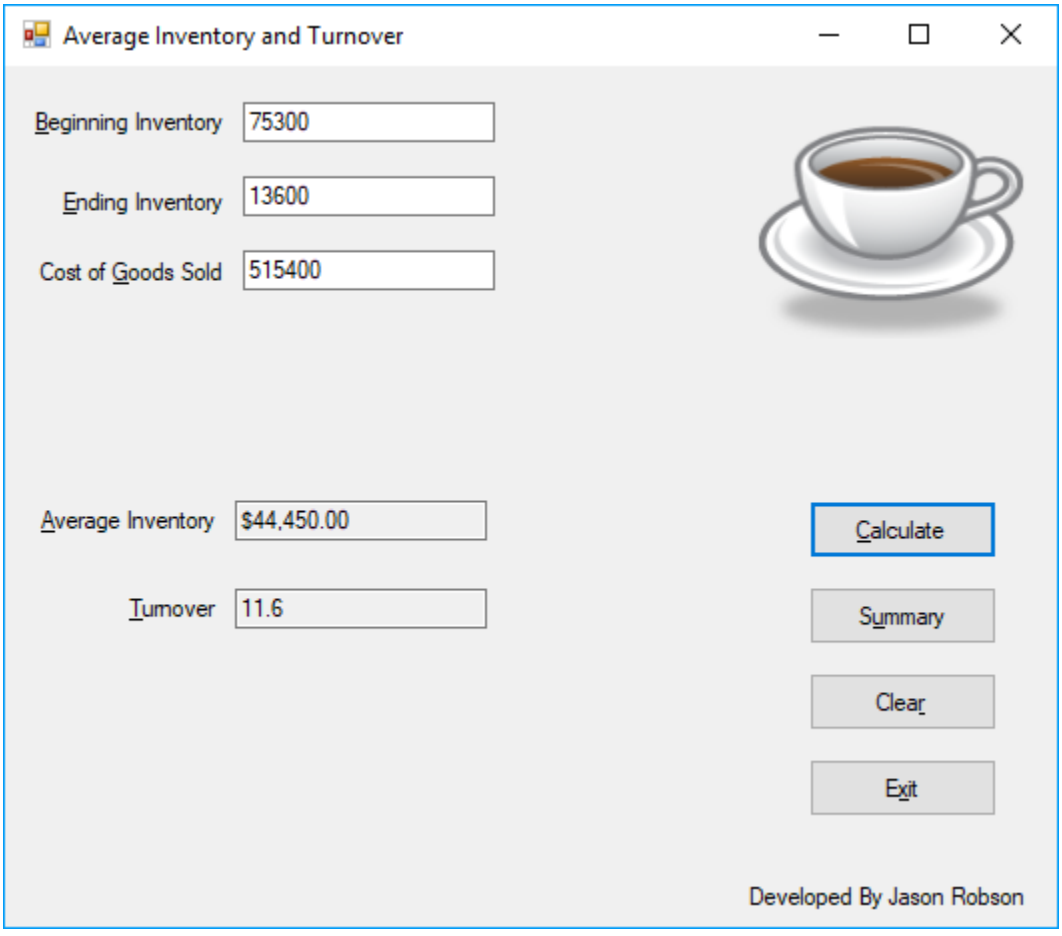




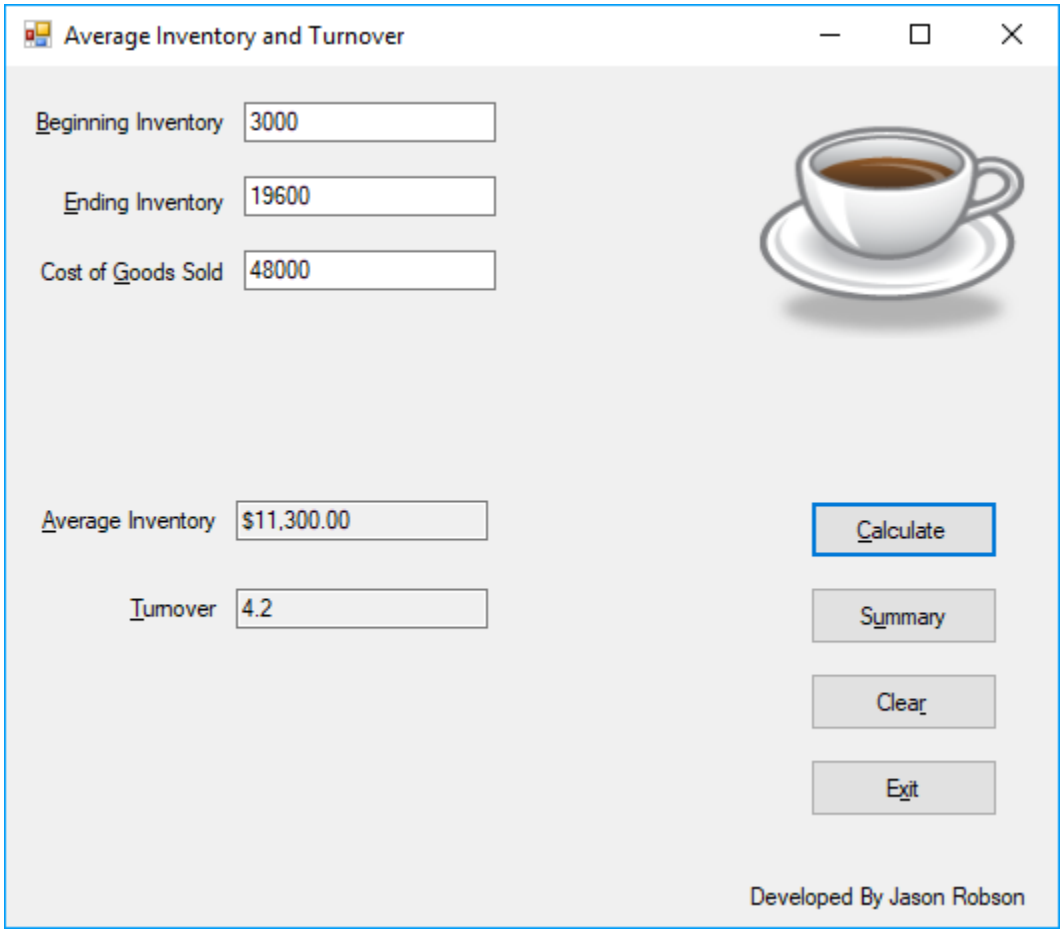
4. Test Data Entered (Calculate Button Clicked)



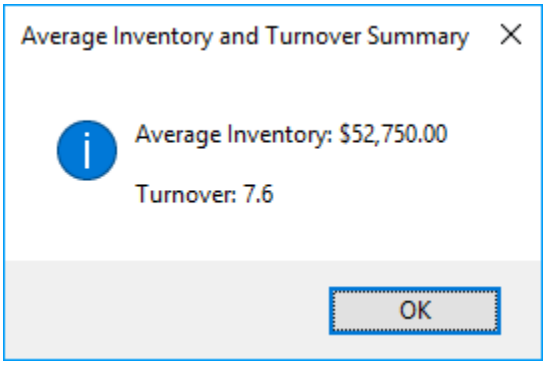
4.1 Test Data Entered (Calculate Button Clicked)

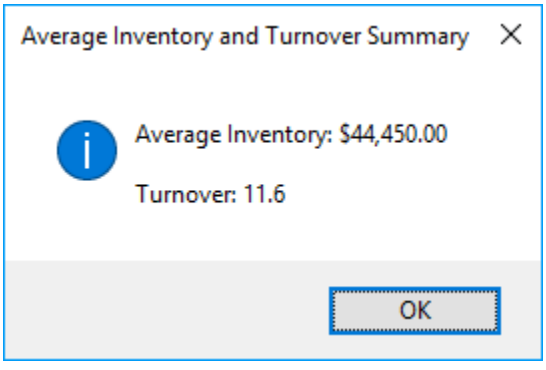


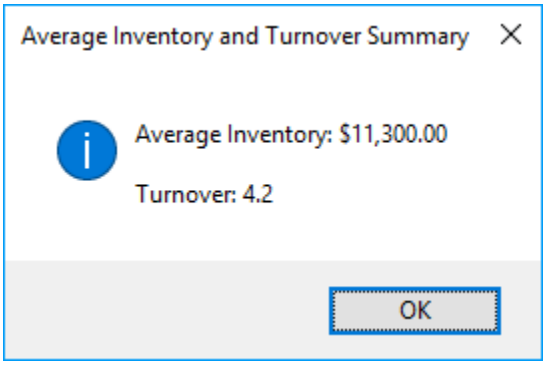
4.2 Test Data Entered (Calculate Button Clicked)



5 Summary (Summary Button Clicked After Execution)







6 Program Code

```
'Project : Average inventory and turnover project
'Programmer : Jason Robson
'Date : September 25, 2017
'Description : This project displays average inventory and turnover based on
values entered in beginning inventory,
'ending inventory, and cost Of goods sold.


Public Class MainForm
    'Declare module-level variables for summary information
    Private AverageInventoryDecimal, TurnoverDecimal As Decimal

    Private Sub CalculateButton_Click(sender As Object, e As EventArgs) Handles
CalculateButton.Click
        'Calculate beginning inventory, ending inventory, cost of goods sold
        Dim BeginningInventoryInteger, EndingInventoryInteger,
CostOfGoodsSoldInteger As Integer

        Try
            'Convert beginning inventory to numeric variable
            BeginningInventoryInteger =
Integer.Parse(BeginningInventoryTextBox.Text)

            Try
                'Convert ending inventory to numeric variable
                EndingInventoryInteger =
Integer.Parse(EndingInventoryTextBox.Text)

                Try
                    'Convert cost of goods sold to numeric variable
                    CostOfGoodsSoldInteger =
Integer.Parse(CostofGoodsSoldTextBox.Text)

                    If BeginningInventoryInteger And EndingInventoryInteger And
CostOfGoodsSoldInteger > 0 Then
                        'If value is greater than zero, summary button activated
                        SummaryButton.Enabled = True

                        'Calculate average inventory and turnover
                        AverageInventoryDecimal =
Convert.ToDecimal(BeginningInventoryInteger + EndingInventoryInteger) / 2
                        TurnoverDecimal = Decimal.Round(CostOfGoodsSoldInteger /
AverageInventoryDecimal, 1)

                        'Format and display answers for average inventory and
turnover
                        AverageInventoryTextBox.Text =
AverageInventoryDecimal.ToString("C")
```

```
                    TurnoverTextBox.Text = TurnoverDecimal.ToString("N1")

                Else
                    'Value is zero or less than
                    MessageBox.Show("Values must be greater than zero.", "Data
entry error",
                        MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
                End If

            Catch CostOfGoodsSold As Exception
                'Handle a cost of goods sold exception
                MessageBox.Show("Cost of goods sold must be numeric.", "Data
Entry Error",
                                        MessageBoxButtons.OK,
MessageBoxIcon.Exclamation)
                With CostofGoodsSoldTextBox
                    .Focus()
                    .SelectAll()
                End With

            End Try

        Catch EndingInventory As Exception
            'Handle an ending inventory exception
            MessageBox.Show("Ending inventory must be numeric.", "Data Entry
Error",
                                    MessageBoxButtons.OK,
MessageBoxIcon.Exclamation)
            With EndingInventoryTextBox
                .Focus()
                .SelectAll()
            End With

        End Try

    Catch BeginningInventory As Exception
        'Handle a beginning inventory exception
        MessageBox.Show("Beginning inventory must be numeric.", "Data Entry
Error",
                                MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
        With BeginningInventoryTextBox
            .Focus()
            .SelectAll()
        End With
    End Try
End Sub

Private Sub SummaryButton_Click(sender As Object, e As EventArgs) Handles
SummaryButton.Click
    'Create messagestring containing average inventory and turnover
    Dim MessageString As String

    'Concatenate message string
    MessageString = "Average Inventory: " &
AverageInventoryDecimal.ToString("C") &
```

```
            Environment.NewLine & Environment.NewLine &
            "Turnover: " & TurnoverDecimal.ToString("N1")

        'Display message string in message box
        MessageBox.Show(MessageString, "Average Inventory and Turnover Summary",
                        MessageBoxButtons.OK, MessageBoxIcon.Information)

    End Sub

    Private Sub ClearButton_Click(sender As Object, e As EventArgs) Handles
ClearButton.Click
        'Clear previous amounts from the form
        AverageInventoryDecimal = 0
        TurnoverDecimal = 0
        EndingInventoryTextBox.Clear()
        CostofGoodsSoldTextBox.Clear()
        AverageInventoryTextBox.Clear()
        TurnoverTextBox.Clear()
        With BeginningInventoryTextBox
            .Clear()
            .Focus()
        End With
        'Disable summary button
        SummaryButton.Enabled = False
    End Sub

    Private Sub ExitButton_Click(sender As Object, e As EventArgs) Handles
ExitButton.Click
        'Close the form
        Me.Close()
    End Sub
End Class
```

BIS 628 – Application Development

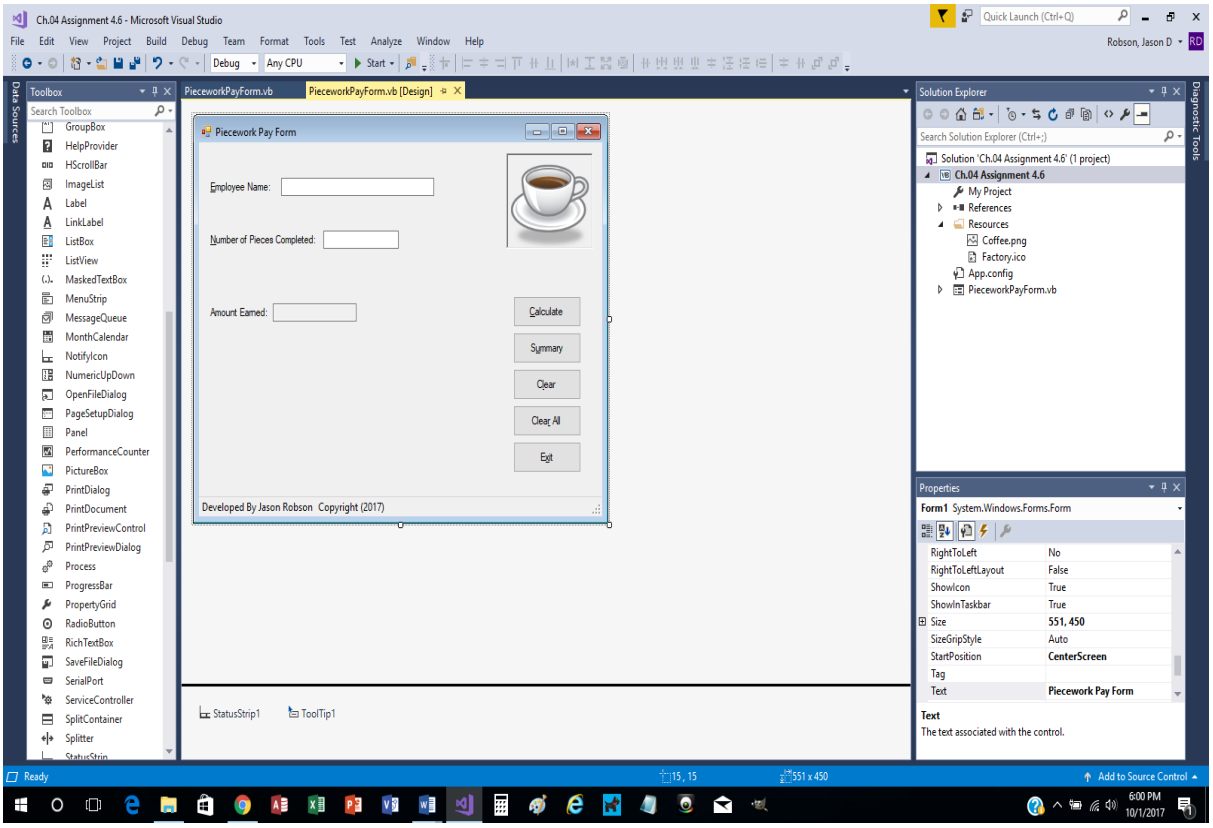
Assignment 4

- Example 4.6 (page 203)

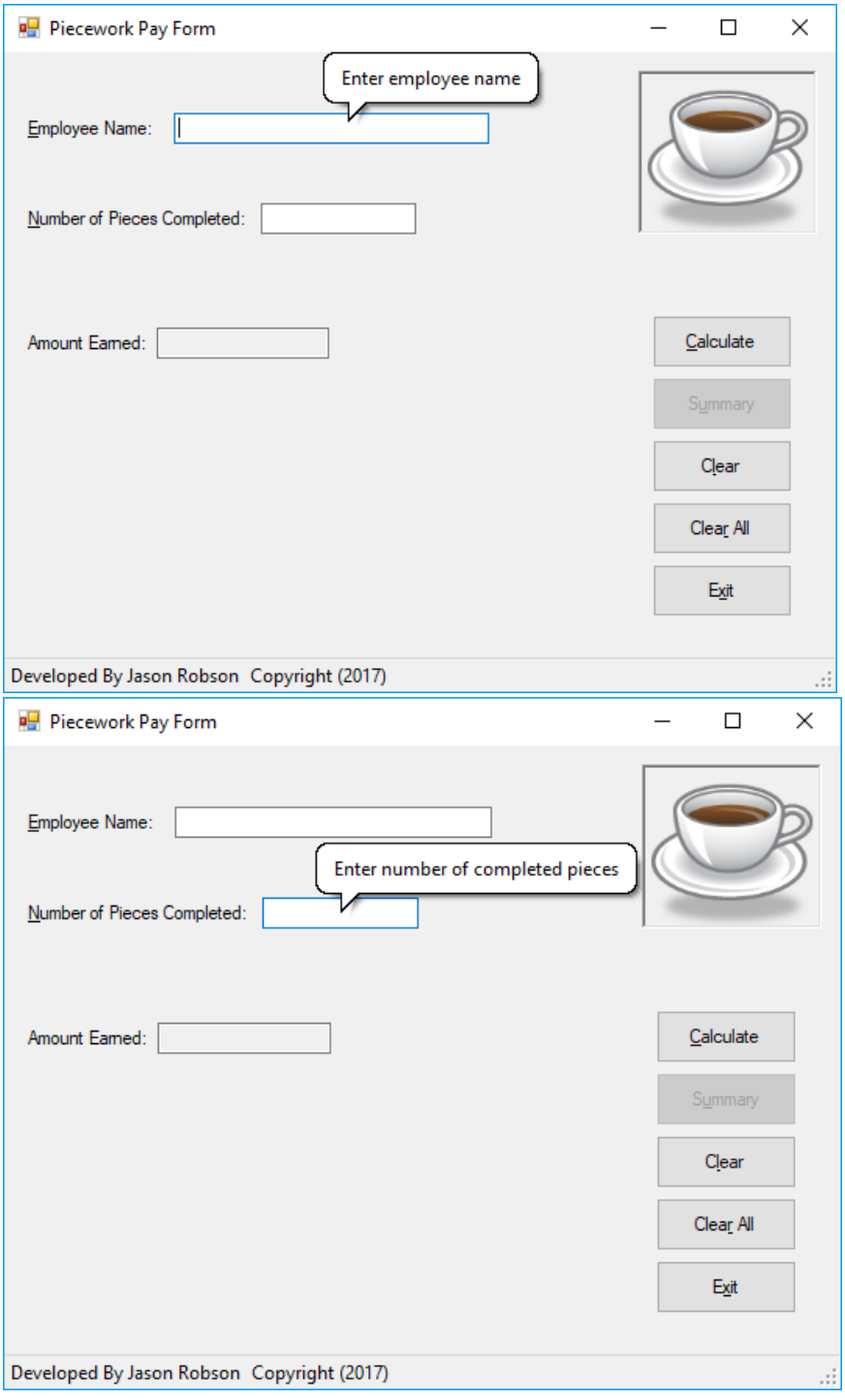
Due Date: 10/04/2017 11:59 pm

By: Jason Robson

1. Design View

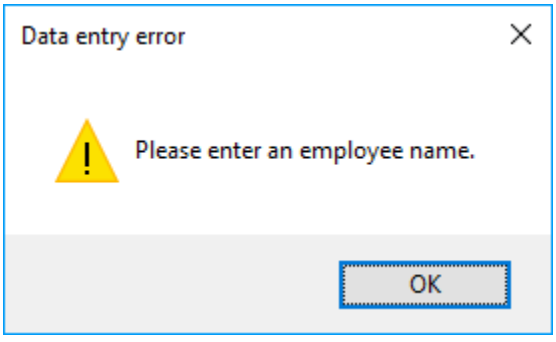


2. ToolTip

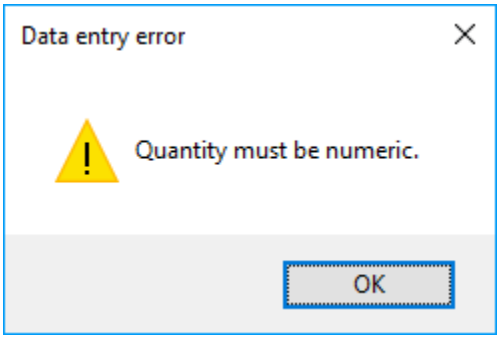

Piecework Pay Form
Enter employee name
Employee Name:
Number of Pieces Completed:
Amount Earned:
Calculate
Summary
Clear
Clear All
Exit
Developed By Jason Robson Copyright (2017)
Piecework Pay Form
Employee Name:
Enter number of completed pieces
Number of Pieces Completed:
Amount Earned:
Calculate
Summary
Clear
Clear All
Exit
Developed By Jason Robson Copyright (2017)

3. Else/Catch Errors

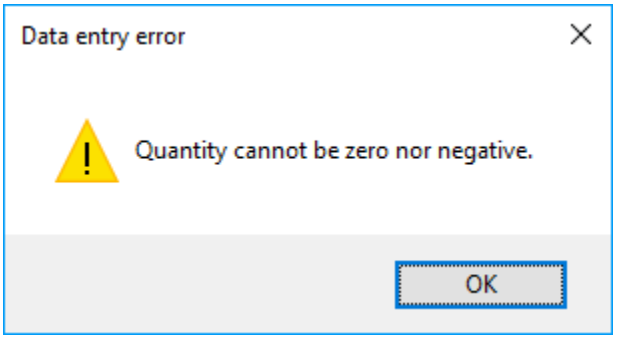
3.1 Employee name not entered



3.2 Non-numeric entered into quantity textbox

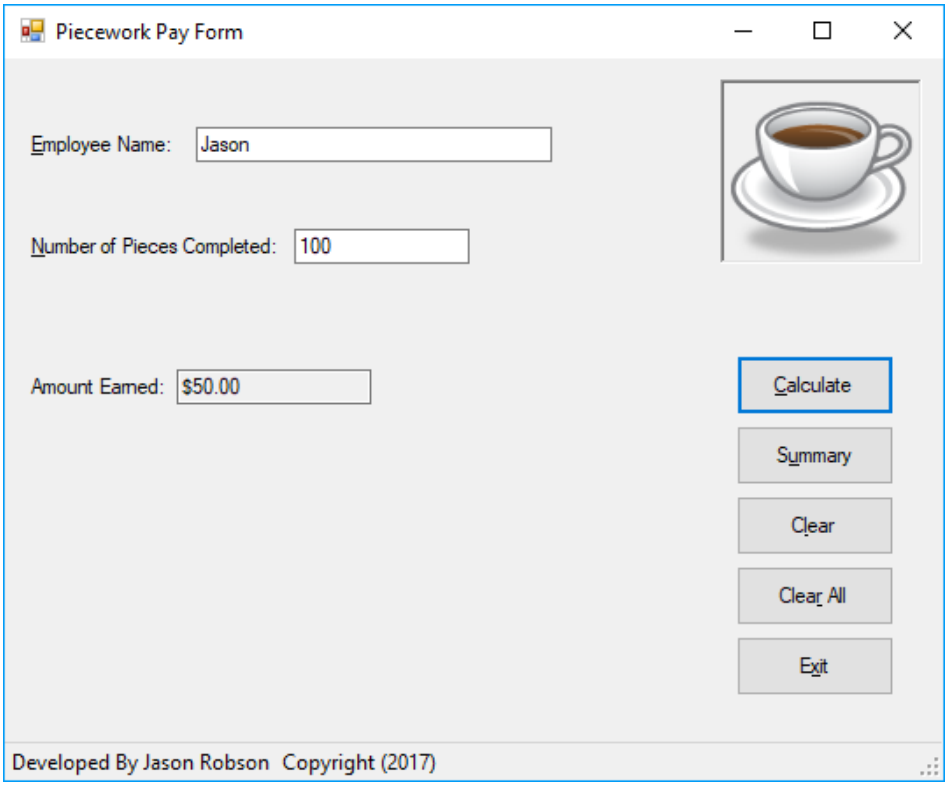


3.3 Numeral is zero or less

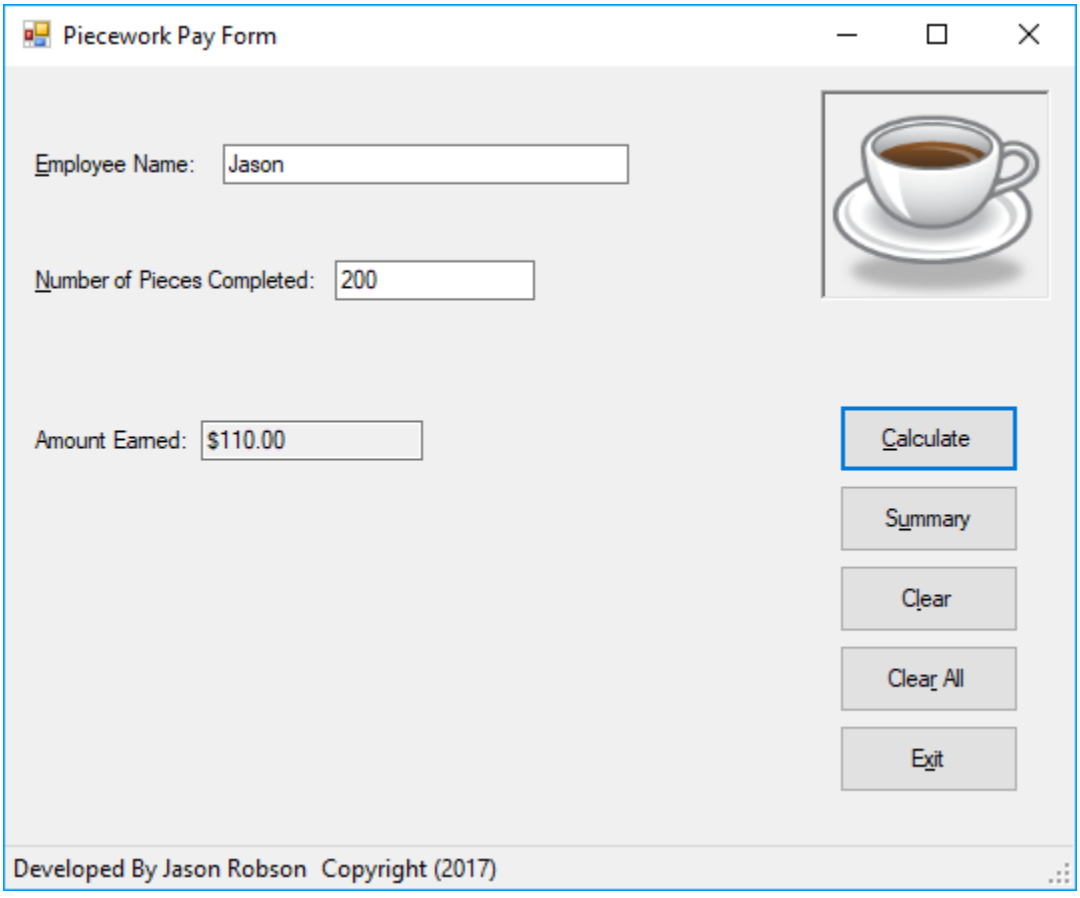


4. Data Calculations (Calculate Button Clicked)

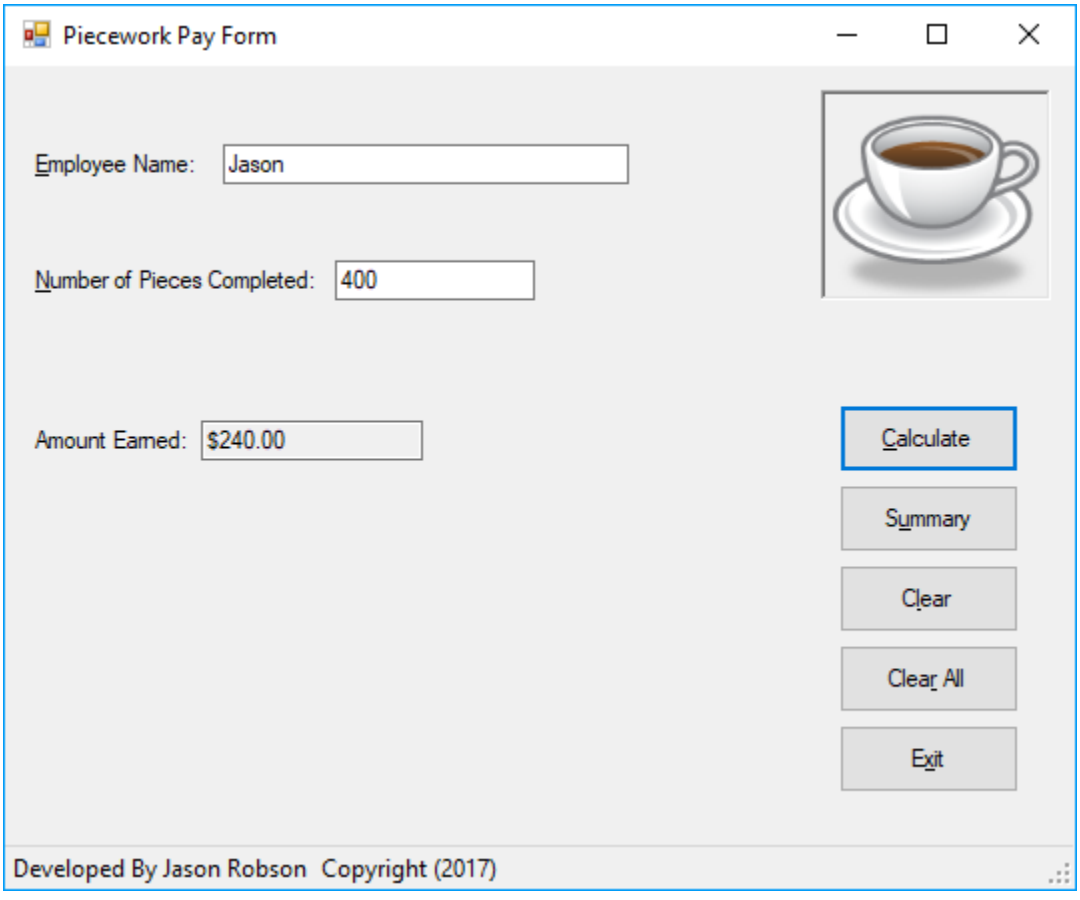
4.1 100 Pieces



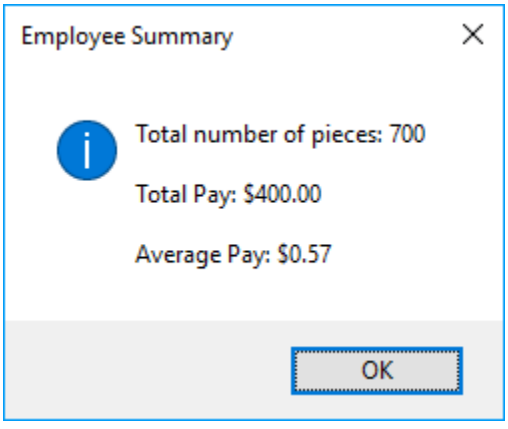
4.2 200 Pieces



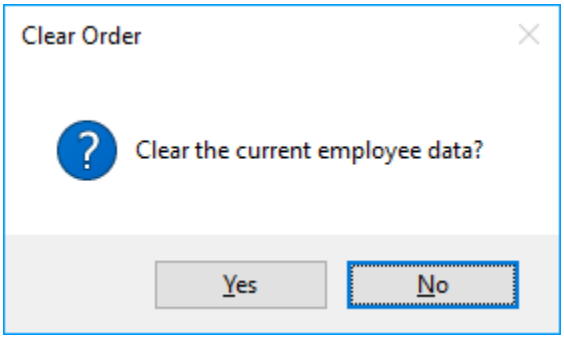
4.3 400 Pieces



5. Summary (Summary Button Clicked)



6. Clear All (Clear All Button Clicked)



7. Program Code

```
'Project     : Piecework Pay Form Project
'Programmer  : Jason Robson
'Date        : October 04, 2017
'Description : This project determines the fixed pay rate based on number of
pieces completed by an employee and determines the pay earned by the employee.

Public Class Form1
    'Declare fixed price paid for pieces completed
    Const PIECES_COMPLETED_199_Decimal As Decimal = 0.5D
    Const PIECES_COMPLETED_399_Decimal As Decimal = 0.55D
    Const PIECES_COMPLETED_599_Decimal As Decimal = 0.6D
    Const PIECES_COMPLETED_600_Decimal As Decimal = 0.65D

    'Declare variables
    Private TotalPiecesInteger As Integer
    Private TotalPayDecimal As Decimal
    Private AveragePayDecimal As Decimal

    Private Function FindPayRate(ByVal QuantityInteger As Integer) As Decimal
        'Get number of pieces completed by person
        QuantityInteger = Integer.Parse(NumberOfPiecesTextBox.Text)
        'Use case to select payrate
        Select Case QuantityInteger
            Case Is >= 600
                Return PIECES_COMPLETED_600_Decimal
            Case 400 To 599
                Return PIECES_COMPLETED_599_Decimal
            Case 200 To 399
                Return PIECES_COMPLETED_399_Decimal
            Case 1 To 199
                Return PIECES_COMPLETED_199_Decimal
        End Select
        'Set return if nothing is entered
        Return ""
    End Function

    Private Sub CalculateButton_Click(sender As Object, e As EventArgs) Handles
CalculateButton.Click
        'Declare local variables
        Dim PayEarnedDecimal As Decimal
        Dim QuantityInteger As Integer

        'Ensure employee name is entered. If not, invoke Else.
        If EmployeeNameTextBox.Text <> "" Then
            'If employee name entered, try to convert number of pieces into
quantity integer
            Try
                QuantityInteger = Integer.Parse(NumberOfPiecesTextBox.Text)
                'If quantity is greater than zero and not negative, begin
calculations.
```

```
                If QuantityInteger > 0 Then
                    'Enable summary button
                    SummaryButton.Enabled = True

                    'Calculate values
                    'Call the function procedure FindPayRate to find payrate and
multiply
                    PayEarnedDecimal =
Convert.ToDecimal(FindPayRate(QuantityInteger) * QuantityInteger)

                    'Calculate remaining values
                    TotalPiecesInteger += QuantityInteger
                    TotalPayDecimal += PayEarnedDecimal
                    AveragePayDecimal = TotalPayDecimal / TotalPiecesInteger

                    'Display employee pay
                    AmountEarnedTextBox.Text = PayEarnedDecimal.ToString("C")

                    'Invoke Else clause if number of pieces entered are zero or
negative
                ElseIf QuantityInteger <= 0 Then
                    'Keep summary button disabled
                    SummaryButton.Enabled = False
                    'Show messagebox informing user quantity can't be zero nor
negative
                    MessageBox.Show("Quantity cannot be zero nor negative.", "Data
entry error",
                                    MessageBoxButtons.OK,
MessageBoxIcon.Exclamation)
                    With NumberOfPiecesTextBox
                        .Focus()
                        .SelectAll()
                    End With
                End If
                'Catch number of pieces entered but not numeric
            Catch QuantityException As FormatException
                MessageBox.Show("Quantity must be numeric.", "Data entry error",
                                MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
                With NumberOfPiecesTextBox
                    .Focus()
                    .SelectAll()
                End With
            End Try
            'Invoke Else clause if no employee name entered.
        Else
            MessageBox.Show("Please enter an employee name.", "Data entry error",
                            MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
            With EmployeeNameTextBox
                .Focus()
            End With

        End If
    End Sub
```

```
    Private Sub ClearButton_Click(sender As Object, e As EventArgs) Handles
ClearButton.Click
        'Clear the form
        NumberOfPiecesTextBox.Clear()
        AmountEarnedTextBox.Clear()

        With EmployeeNameTextBox
            .Clear()
            .Focus()
        End With
    End Sub

    Private Sub ClearAllButton_Click(sender As Object, e As EventArgs) Handles
ClearAllButton.Click
        'Clear the current employee data
        Dim ReturnDialogResult As DialogResult
        Dim MessageString As String

        'Confirm clear of employee data
        MessageString = "Clear the current employee data?"
        ReturnDialogResult = MessageBox.Show(MessageString, "Clear Order",
                                             MessageBoxButtons.YesNo,
MessageBoxIcon.Question,
                                             MessageBoxDefaultButton.Button2)

        If ReturnDialogResult = DialogResult.Yes Then
            'User said Yes, clear the fields and disable summary button
            SummaryButton.Enabled = False
            ClearButton_Click(sender, e)
            TotalPayDecimal = 0
            TotalPiecesInteger = 0
            AveragePayDecimal = 0
        End If
    End Sub

    Private Sub SummaryButton_Click(sender As Object, e As EventArgs) Handles
SummaryButton.Click
        'Declare message string
        Dim MessageString As String

        'Write message string
        MessageString = "Total number of pieces: " &
TotalPiecesInteger.ToString("N0") &
            Environment.NewLine & Environment.NewLine &
            "Total Pay: " & TotalPayDecimal.ToString("C") &
            Environment.NewLine & Environment.NewLine &
            "Average Pay: " & AveragePayDecimal.ToString("C")
        'Show message box
        MessageBox.Show(MessageString, "Employee Summary",
                        MessageBoxButtons.OK, MessageBoxIcon.Information)

    End Sub
    Private Sub ExitButton_Click(sender As Object, e As EventArgs) Handles
ExitButton.Click
        'Exit the form
```

```
        Me.Close()
    End Sub
End Class
```

BIS 628 – Application Development

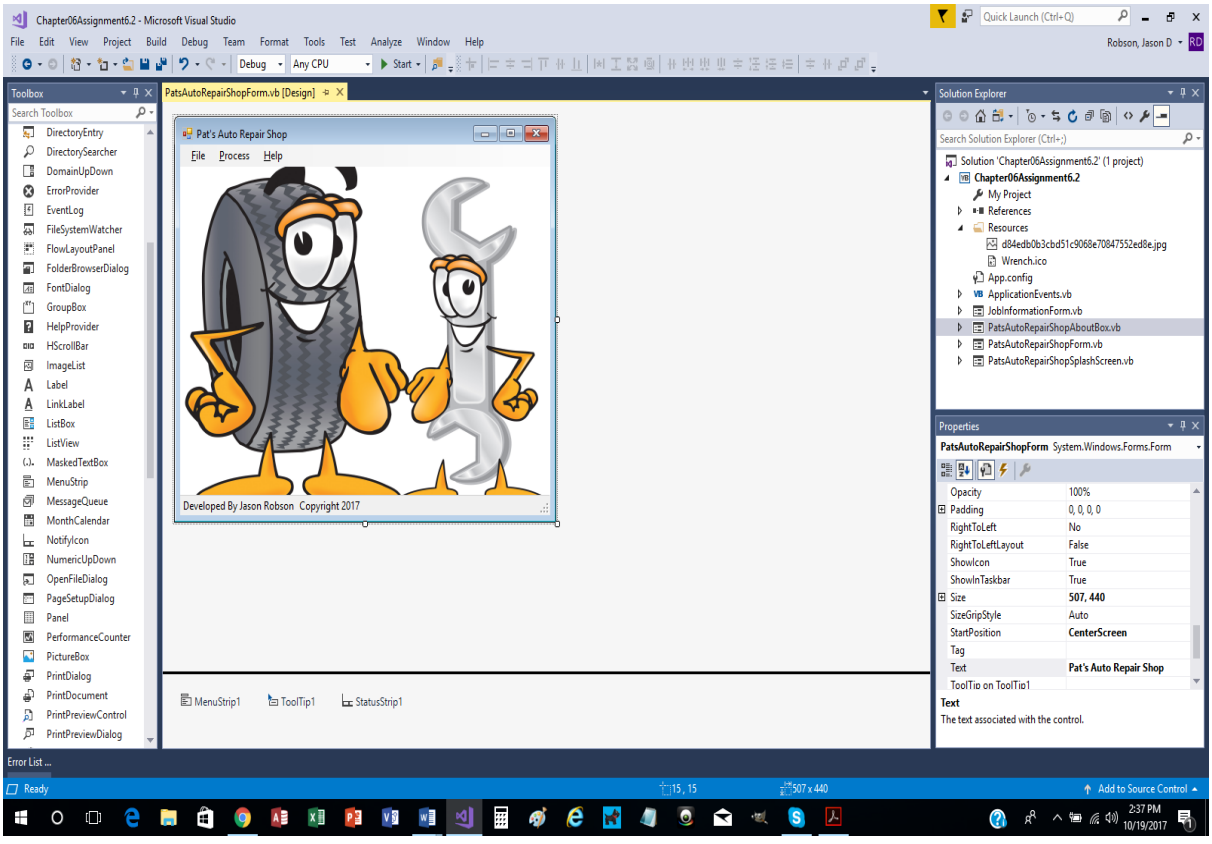
Assignment 6

- Example 6.2 (page 279-280)

Due Date: 10/25/2017 11:59 pm

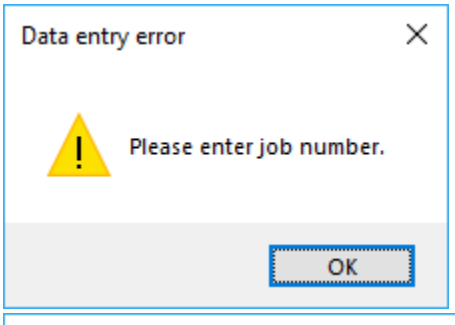
By: Jason Robson

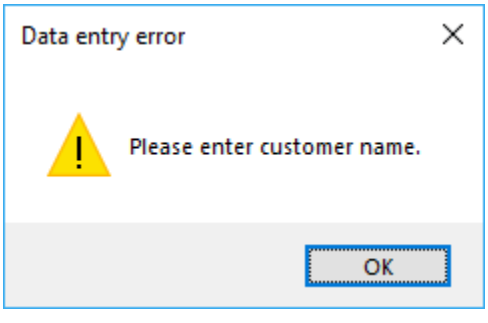
1. Design View

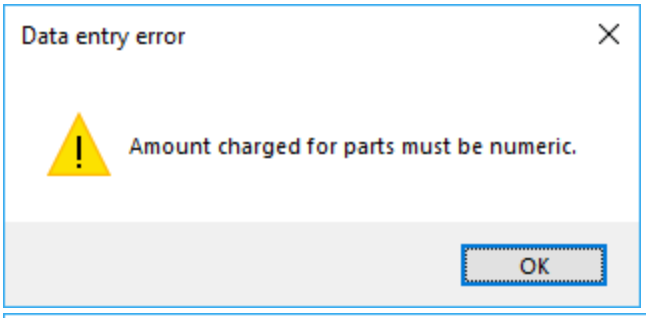


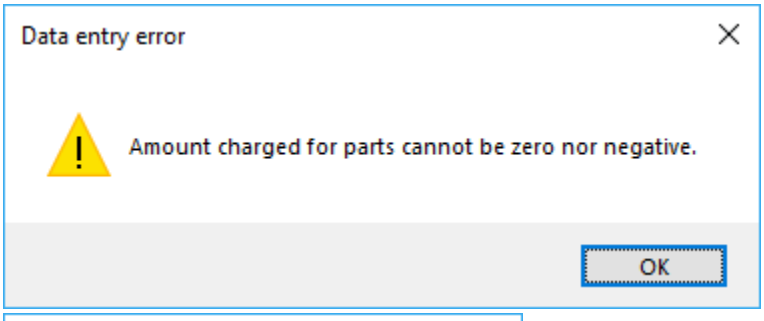
2. Splash Screen

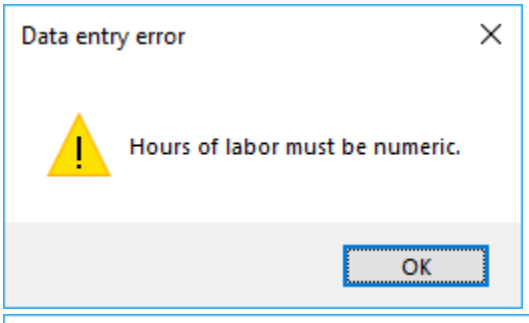
3. Error Messages (Textbox left blank, non-numeric, zero or less than)

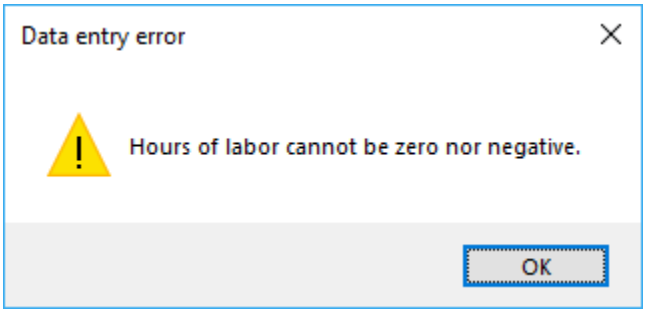




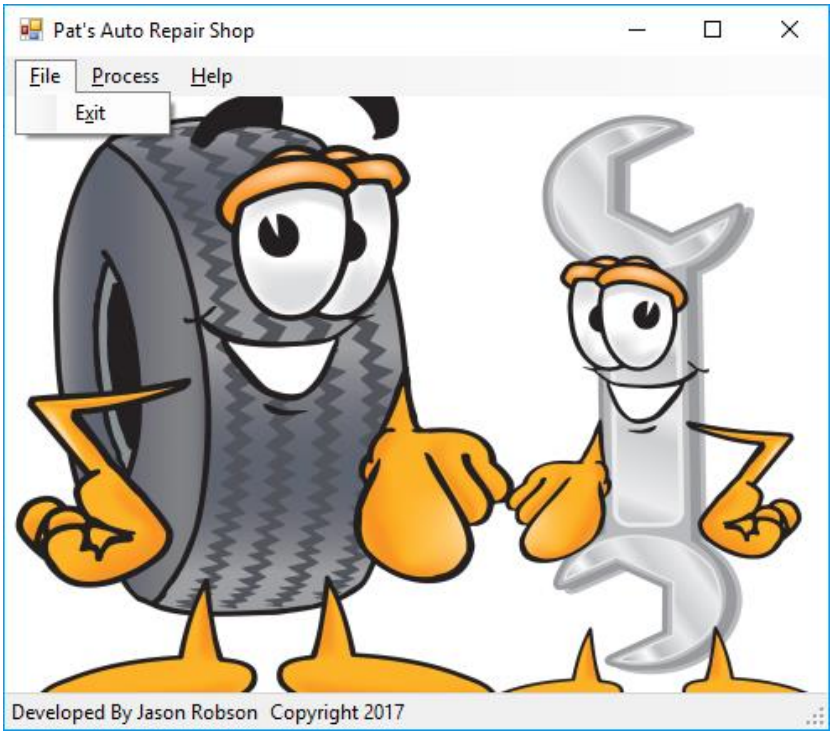

Data entry error
Amount charged for parts must be numeric.
OK

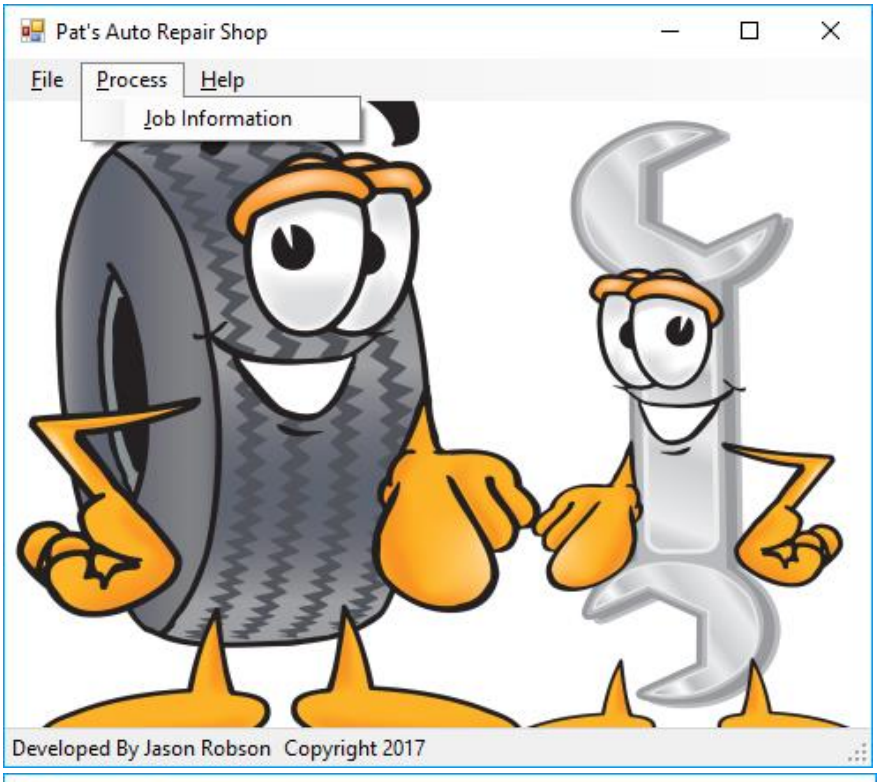

Data entry error
Amount charged for parts cannot be zero nor negative.
OK

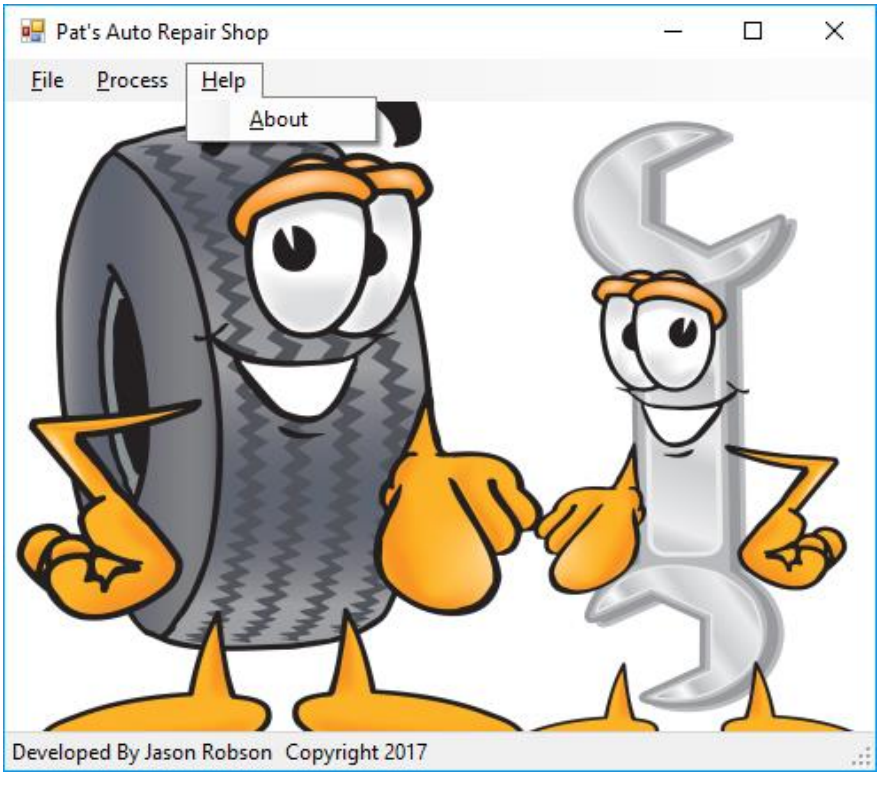

Data entry error
Hours of labor must be numeric.
OK


Data entry error
Hours of labor cannot be zero nor negative.
OK

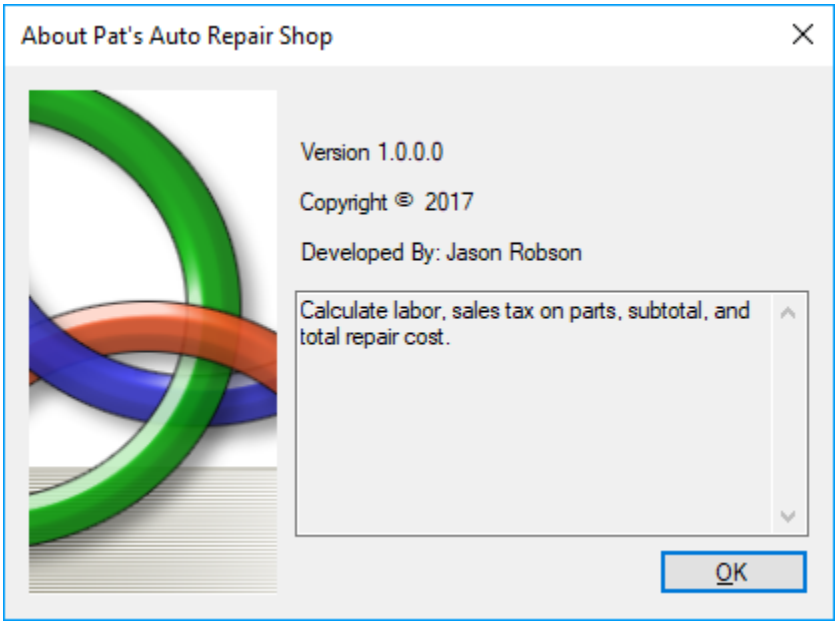
4. Main Form (With menu strip selections)



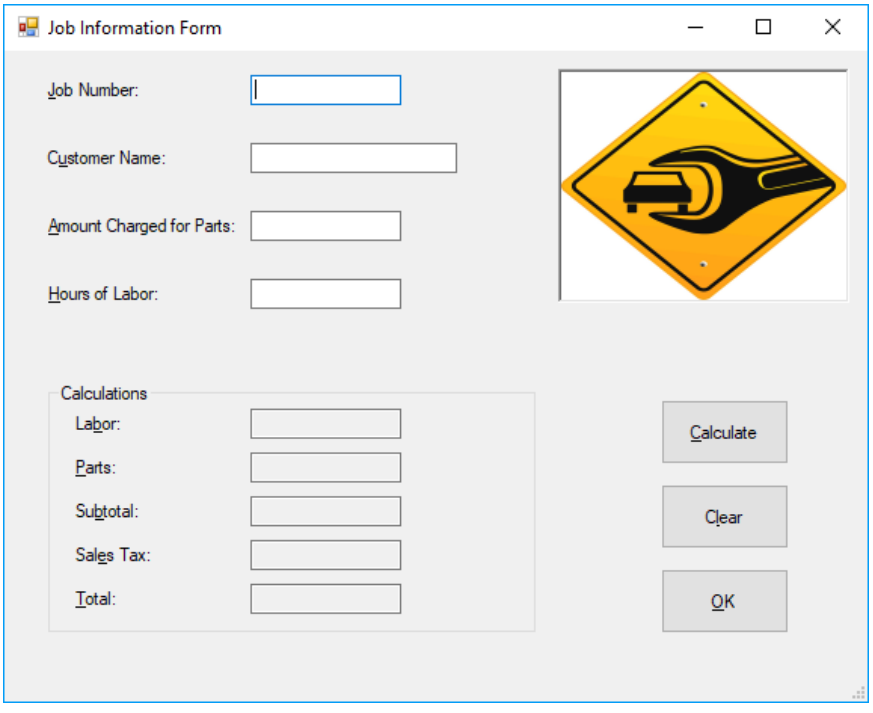

Pat's Auto Repair Shop
File
Process
Help
Job Information
Developed By Jason Robson Copyright 2017


Pat's Auto Repair Shop
File
Process
Help
About
Developed By Jason Robson Copyright 2017

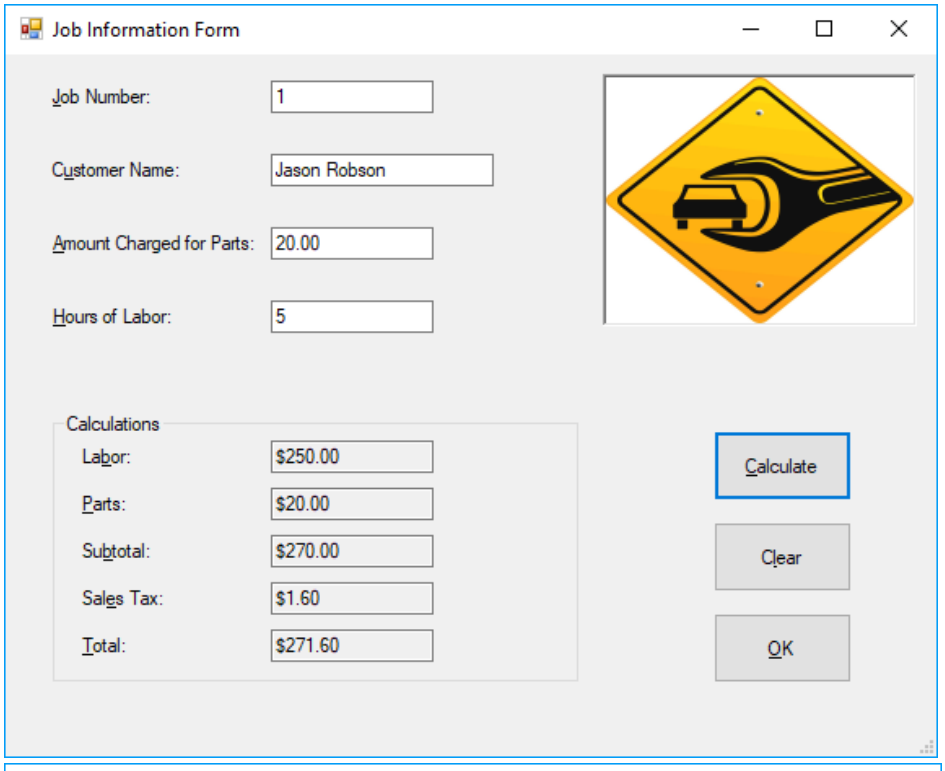
5. About Box

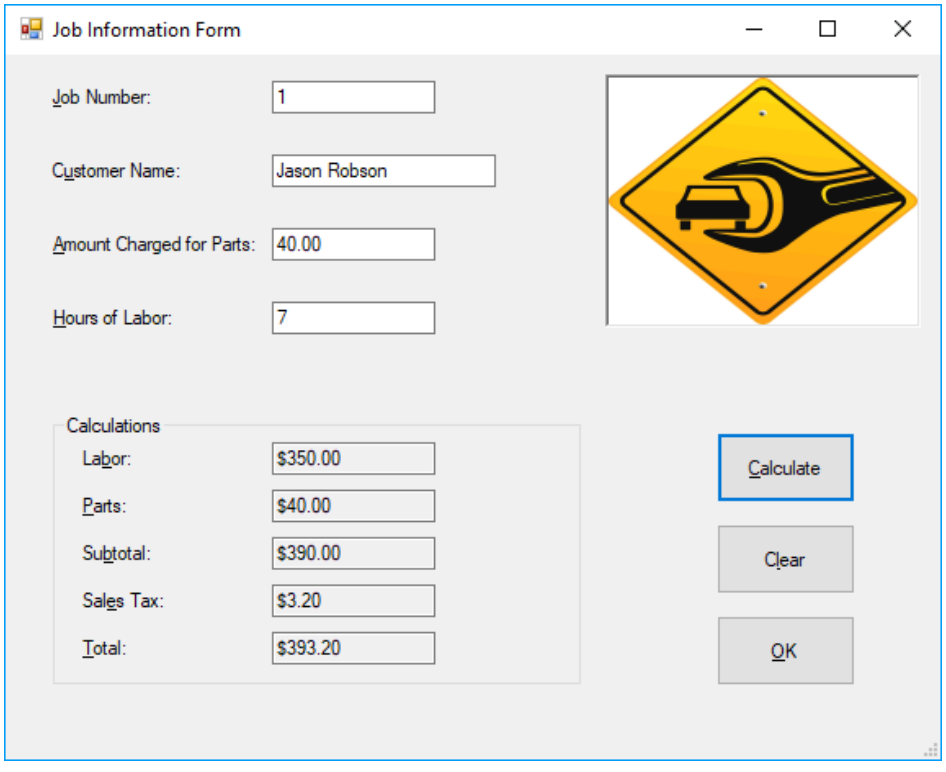


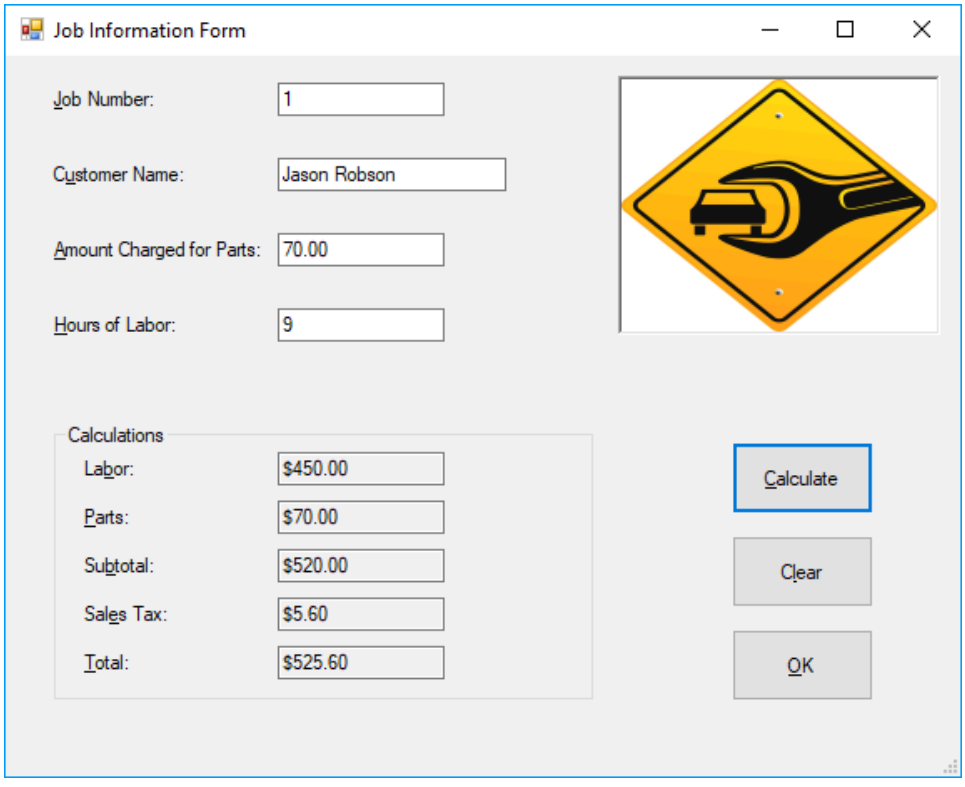
6. Job Information Form



6.1 Calculation







7. Program Code (Job Information Form)

```
'Project     : Pats Auto Repair Shop
'Programmer  : Jason Robson
'Date        : October 25, 2017
'Description : This project displays a splash screen for five seconds and displays
the main form. User can move to the job information form which calculates labor
and parts to determine subtotal, calculates sales tax on parts, and determines the
total cost for the job.

Public Class JobInformationForm
    'Declare constants
    Const TAX_RATE_Decimal As Decimal = 0.08D
    Const HOURLY_LABOR_CHARGE_Integer As Integer = 50

    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles
Button3.Click
        'Close the form
        Me.Close()
    End Sub

    Private Sub ClearButton_Click(sender As Object, e As EventArgs) Handles
ClearButton.Click
        'Clear the form and set focus
        CustomerNameTextBox.Text = ""
        AmountChargedForPartsTextBox.Clear()
```

```
        HoursOfLaborTextBox.Clear()
        SubtotalTextBox.Clear()
        SalesTaxTextBox.Clear()
        TotalTextBox.Clear()
        PartsTextBox.Clear()
        LaborTextBox.Clear()

        With JobNumberTextBox
            .Clear()
            .Focus()
        End With
    End Sub

    Private Sub CalculateButton_Click(sender As Object, e As EventArgs) Handles
CalculateButton.Click
        'Declare local variables
        Dim AmountChargedForPartsDecimal, SalesTaxDecimal, SubtotalDecimal,
TotalDecimal As Decimal
        Dim HoursOfLaborInteger, LaborPayInteger As Integer

        'Ensure job number is not blank
        If JobNumberTextBox.Text <> "" Then
            'Ensure customer name is not blank
            If CustomerNameTextBox.Text <> "" Then
                'If job number and customer name is not blank, try to convert
amount charged for parts from a string to a numeric value
                Try
                    AmountChargedForPartsDecimal =
Decimal.Parse(AmountChargedForPartsTextBox.Text)
                    'If conversion is successful and amount charged for parts is
greater than zero
                    If AmountChargedForPartsTextBox.Text > 0 Then
                        'Try to convert hours of labor from a string to a numeric
value
                        Try
                            HoursOfLaborInteger =
Integer.Parse(HoursOfLaborTextBox.Text)
                            'If conversion is successful and hours of labor is
greater than zero
                            If HoursOfLaborTextBox.Text > 0 Then

                                'Calculate totals
                                LaborPayInteger = HoursOfLaborInteger *
HOURLY_LABOR_CHARGE_Integer
                                SalesTaxDecimal = AmountChargedForPartsDecimal *
TAX_RATE_Decimal
                                SubtotalDecimal = LaborPayInteger +
AmountChargedForPartsDecimal
                                TotalDecimal = SubtotalDecimal + SalesTaxDecimal

                                'Convert to string and display in textboxes
                                LaborTextBox.Text = LaborPayInteger.ToString("C")
                                PartsTextBox.Text =
AmountChargedForPartsDecimal.ToString("C")
```

```
                                SubtotalTextBox.Text =
SubtotalDecimal.ToString("C")
                                SalesTaxTextBox.Text =
SalesTaxDecimal.ToString("C")
                                TotalTextBox.Text = TotalDecimal.ToString("C")

                                'If hours of labor is zero or less than, inform
user
                            ElseIf HoursOfLaborTextBox.Text <= 0 Then

                                MessageBox.Show("Hours of labor cannot be zero nor
negative.", "Data entry error",
                                                MessageBoxButtons.OK,
MessageBoxIcon.Exclamation)
                                With HoursOfLaborTextBox
                                    .SelectAll()
                                    .Focus()
                                End With
                            End If

                            'Catch hours of labor exception is not numeric
                        Catch HoursOfLaborException As FormatException

                            MessageBox.Show("Hours of labor must be numeric.",
"Data entry error",
                                            MessageBoxButtons.OK,
MessageBoxIcon.Exclamation)
                            With HoursOfLaborTextBox
                                .SelectAll()
                                .Focus()
                            End With
                        End Try

                        'If amount charged for parts is zero or less than, inform
user
                    ElseIf AmountChargedForPartsTextBox.Text <= 0 Then

                        MessageBox.Show("Amount charged for parts cannot be zero
nor negative.", "Data entry error",
                                        MessageBoxButtons.OK,
MessageBoxIcon.Exclamation)
                        With AmountChargedForPartsTextBox
                            .SelectAll()
                            .Focus()
                        End With
                    End If

                    'Catch amount charged for parts is not numeric
                Catch AmountChargedForPartsException As FormatException

                    MessageBox.Show("Amount charged for parts must be numeric.",
"Data entry error",
                                    MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
                    With AmountChargedForPartsTextBox
                        .SelectAll()
```

```
                    .Focus()
                End With
            End Try

            'If customer name is blank, inform user
        Else
            MessageBox.Show("Please enter customer name.", "Data entry error",
                        MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
            With CustomerNameTextBox
                .Focus()
            End With

        End If

        'If job number is blank, inform user
    Else
        MessageBox.Show("Please enter job number.", "Data entry error",
                        MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
        With JobNumberTextBox
            .Focus()
        End With

    End If

End Sub
   End Class
```

7.1 Program Code (Main Form)

```
Public Class PatsAutoRepairShopForm
    Private Sub PatsAutoRepairShop_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
        'Delay splash screen for five seconds
        Threading.Thread.Sleep(5000)
    End Sub
    Private Sub AboutToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles AboutToolStripMenuItem.Click
        'Show about form and require user to respond
        PatsAutoRepairShopAboutBox.ShowDialog()
    End Sub

    Private Sub ExitToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles ExitToolStripMenuItem.Click
        'Exit the form
        Me.Close()
    End Sub

    Private Sub JobInformationToolStripMenuItem_Click(sender As Object, e As
EventArgs) Handles JobInformationToolStripMenuItem.Click
        'Show the job information form and require user to respond
        JobInformationForm.ShowDialog()
    End Sub
```

```
End Class
```

BIS 628 – Application Development

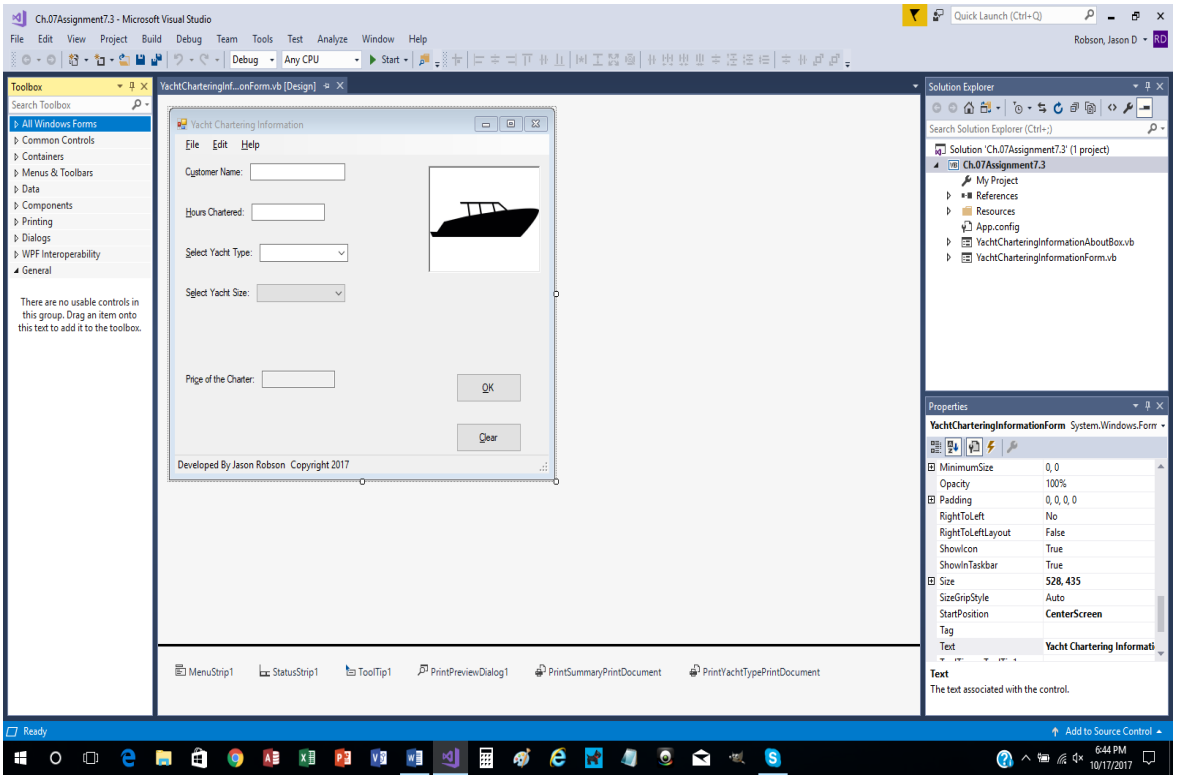
Assignment 7

- Example 7.3 (page 321)

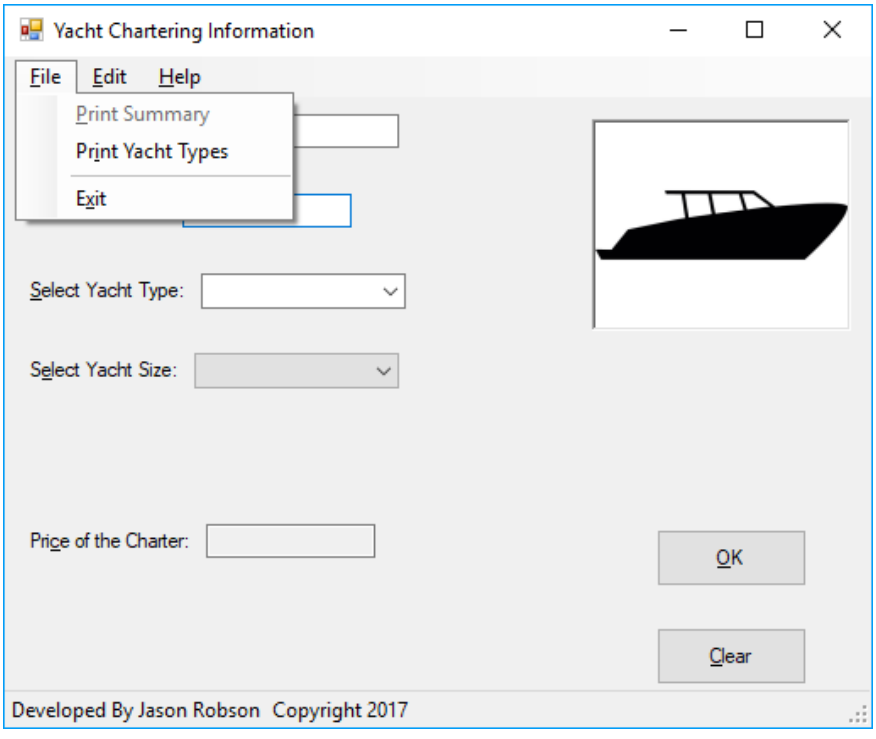
Due Date: 11/01/2017 11:59 pm

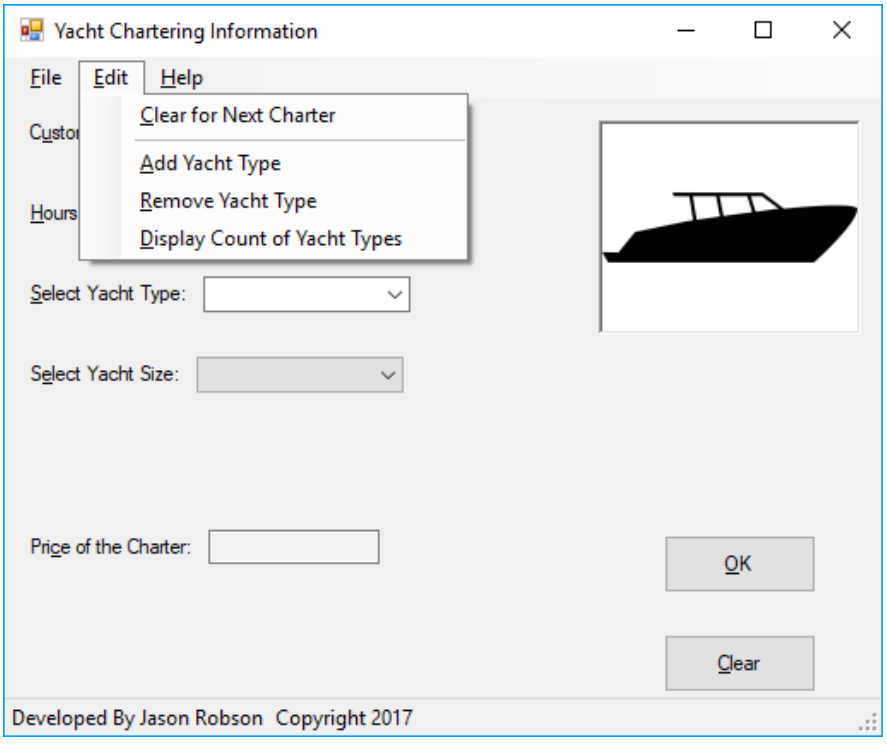
By: Jason Robson

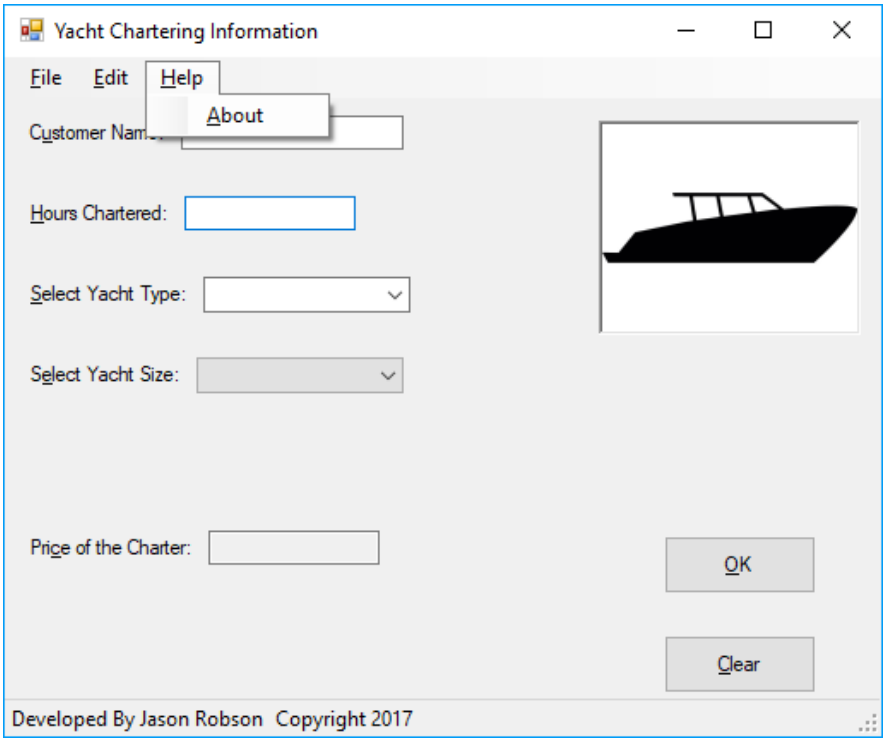
1. Design View



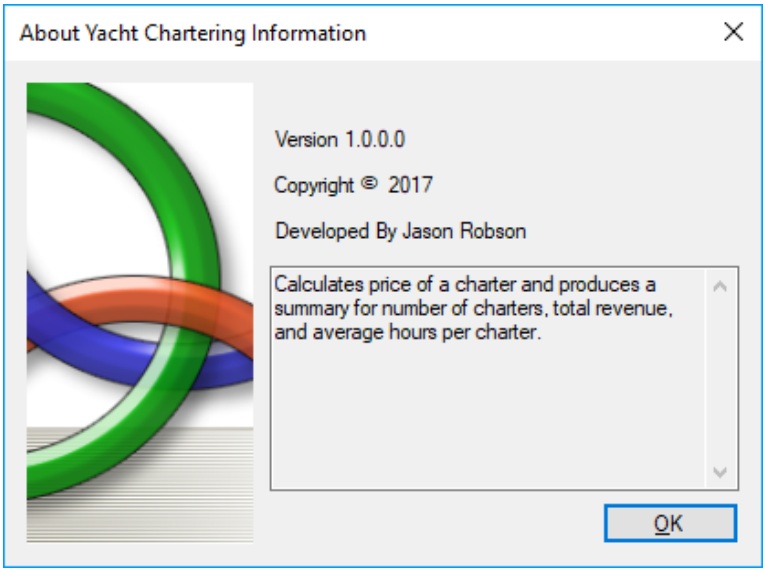
2. MenuStrip


Yacht Chartering Information
File
Edit
Help
Print Summary
Print Yacht Types
Exit
Select Yacht Type:
Select Yacht Size:
Price of the Charter:
OK
Clear
Developed By Jason Robson Copyright 2017

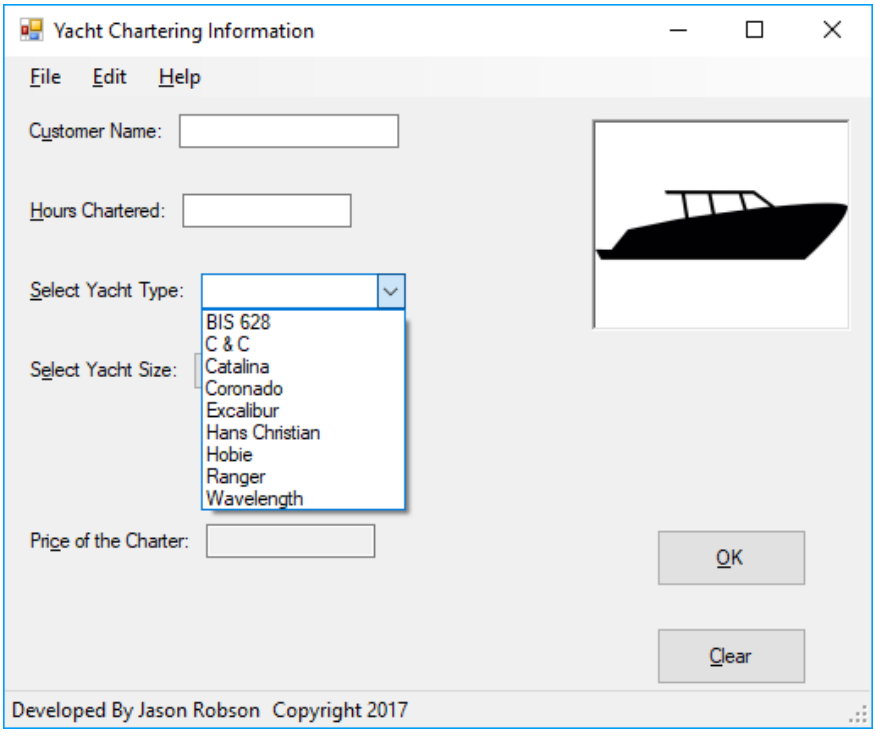

Yacht Chartering Information
File
Edit
Help
Clear for Next Charter
Add Yacht Type
Remove Yacht Type
Display Count of Yacht Types
Select Yacht Type:
Select Yacht Size:
Price of the Charter:
OK
Clear
Developed By Jason Robson Copyright 2017



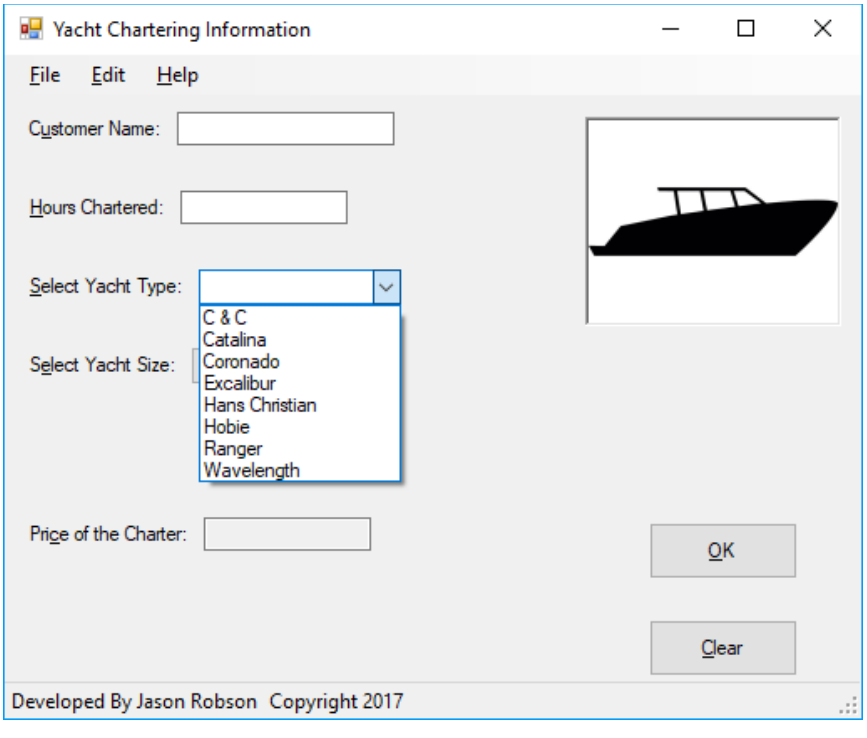
3. About Box (About box menu strip clicked)



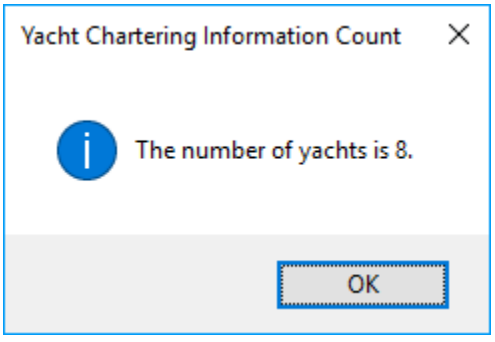
4. Yacht Added (Add yacht type clicked)



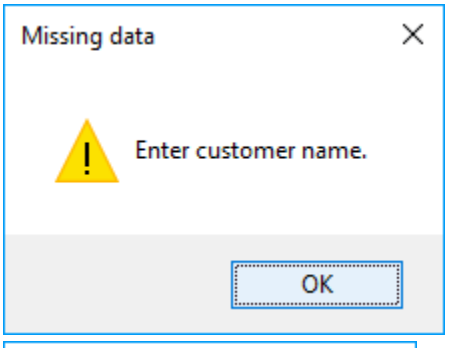
5. Yacht Removed (Remove yacht type clicked)

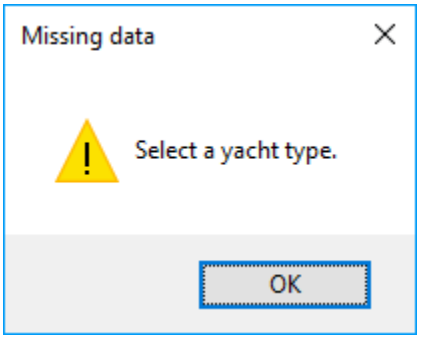


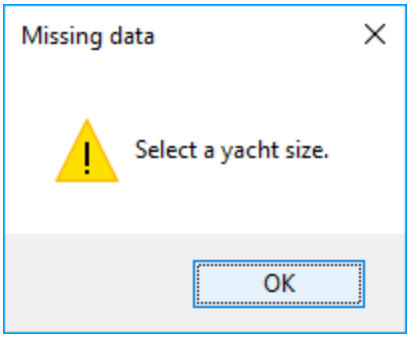
6. Display Yacht Count (Display Count of Yacht Types clicked)

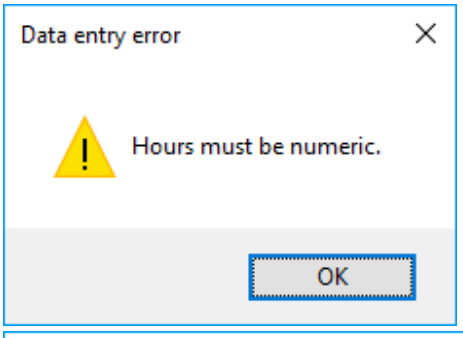


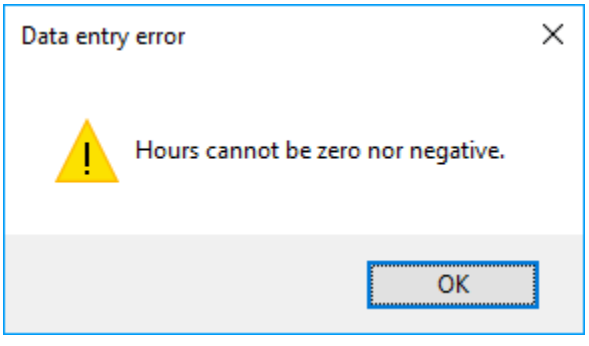
7. Error Messages (Hours chartered less than zero or zero, blank customer name, yacht size and type not selected)



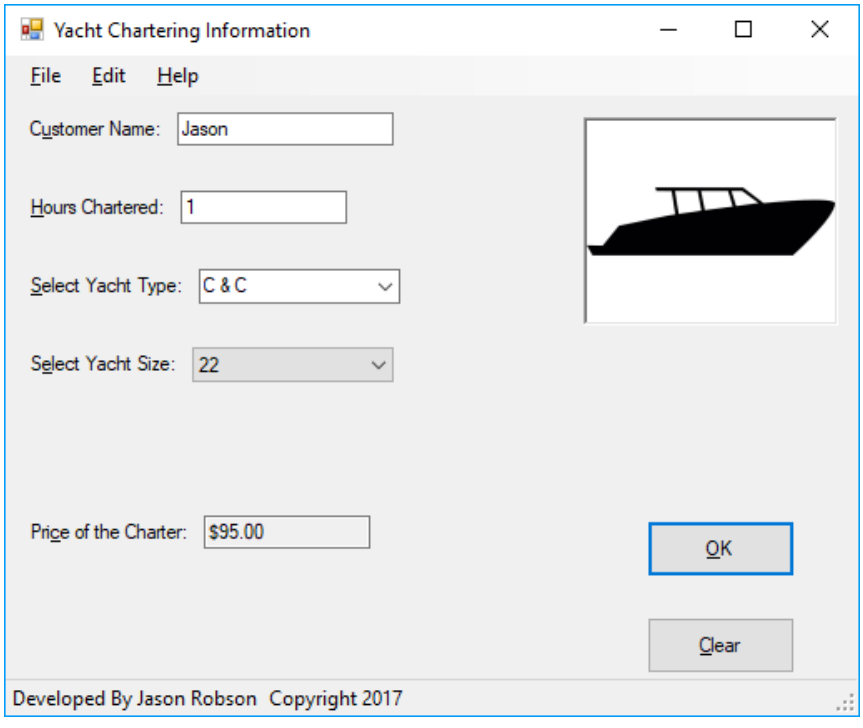



Missing data
Select a yacht size.
OK

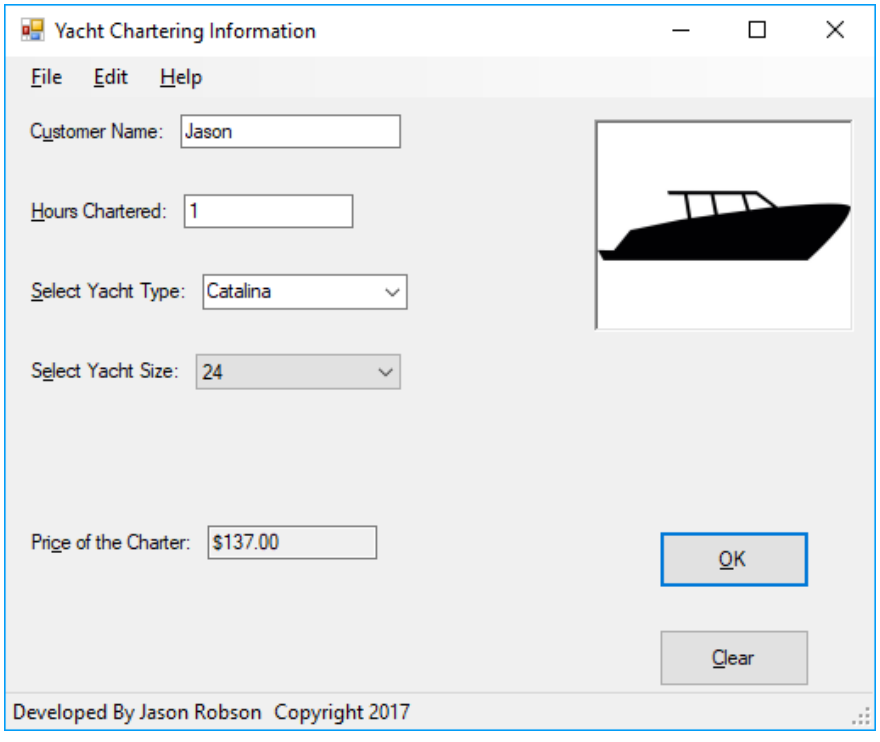

Data entry error
Hours must be numeric.
OK


Data entry error
Hours cannot be zero nor negative.
OK

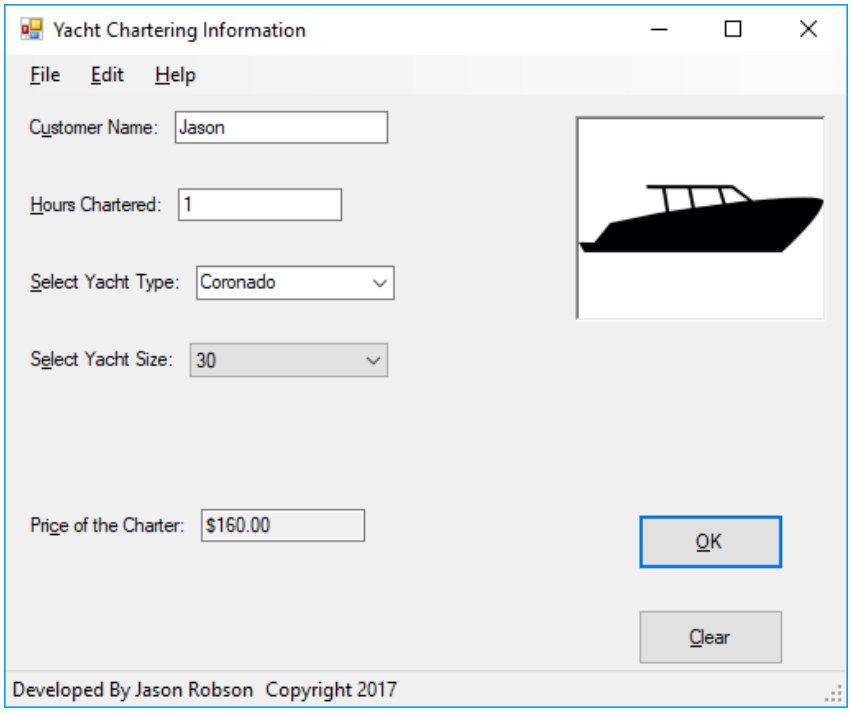
8. Yacht Size 22 Selected (OK button clicked)



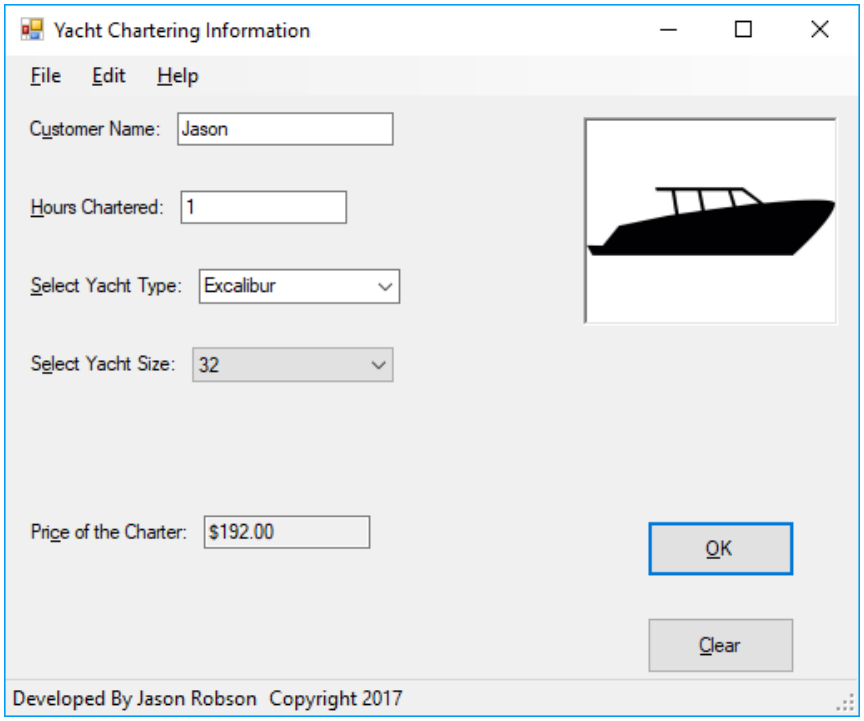
8.1 Yacht Size 24 Selected



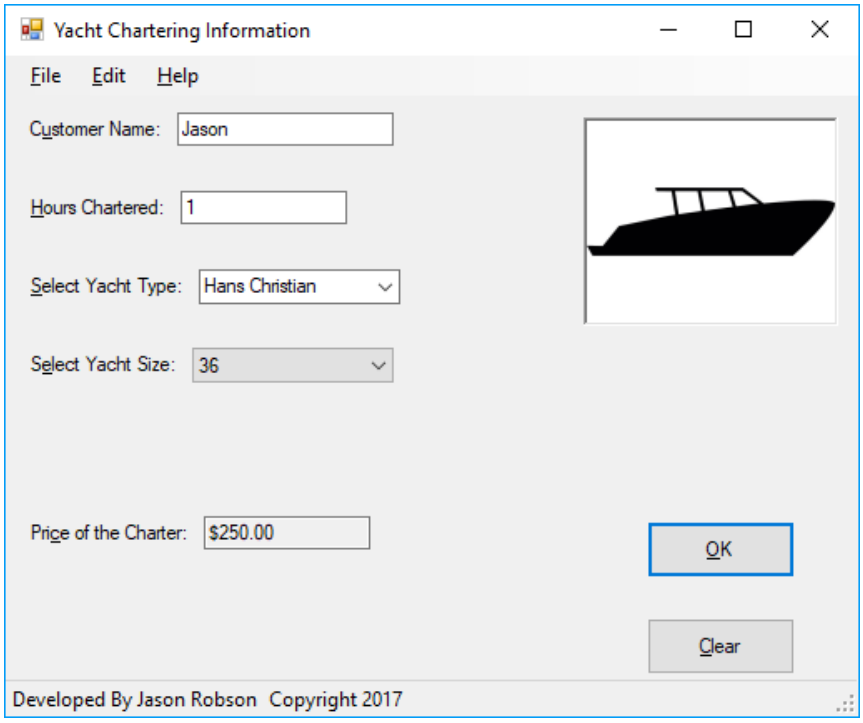
8.2 Yacht Size 30 Selected



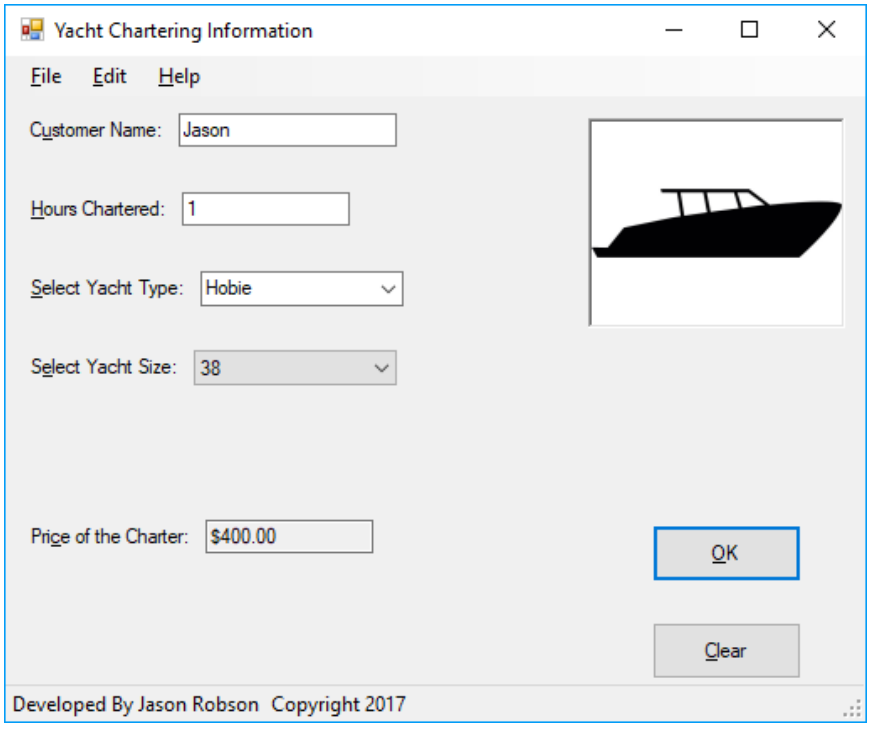
8.3 Yacht Size 32 Selected



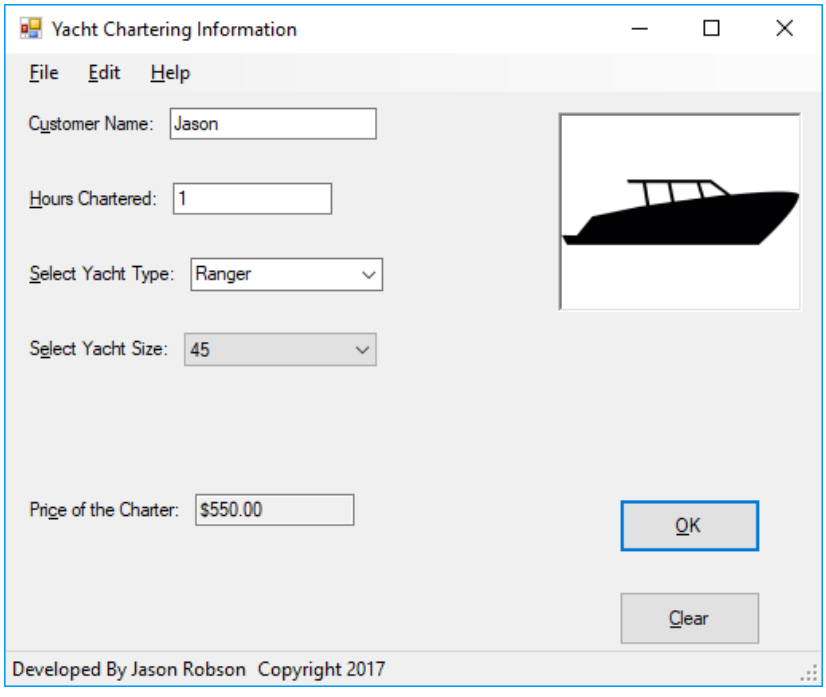
8.4 Yacht Size 36 Selected



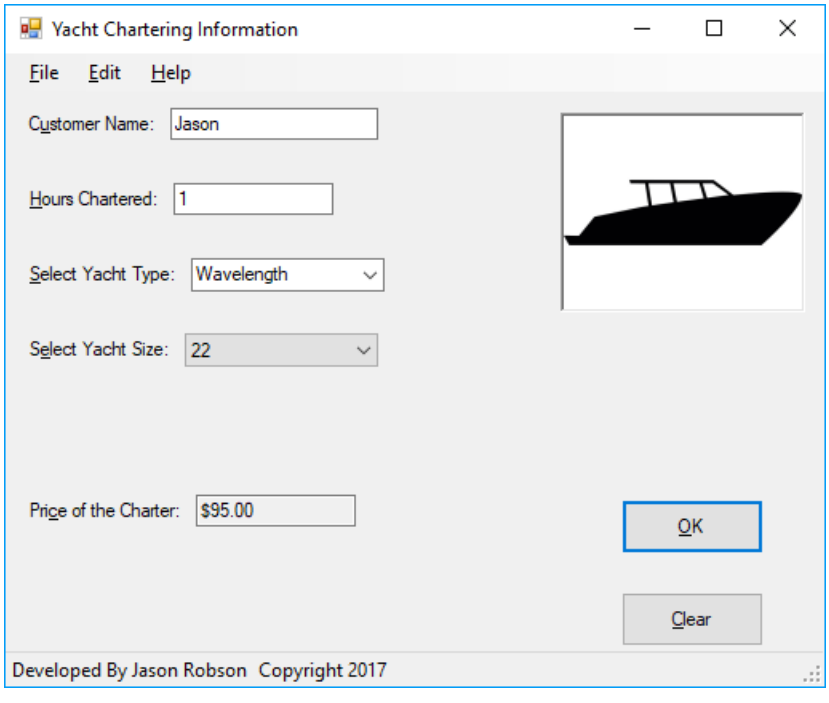
8.5 Yacht Size 38 Selected



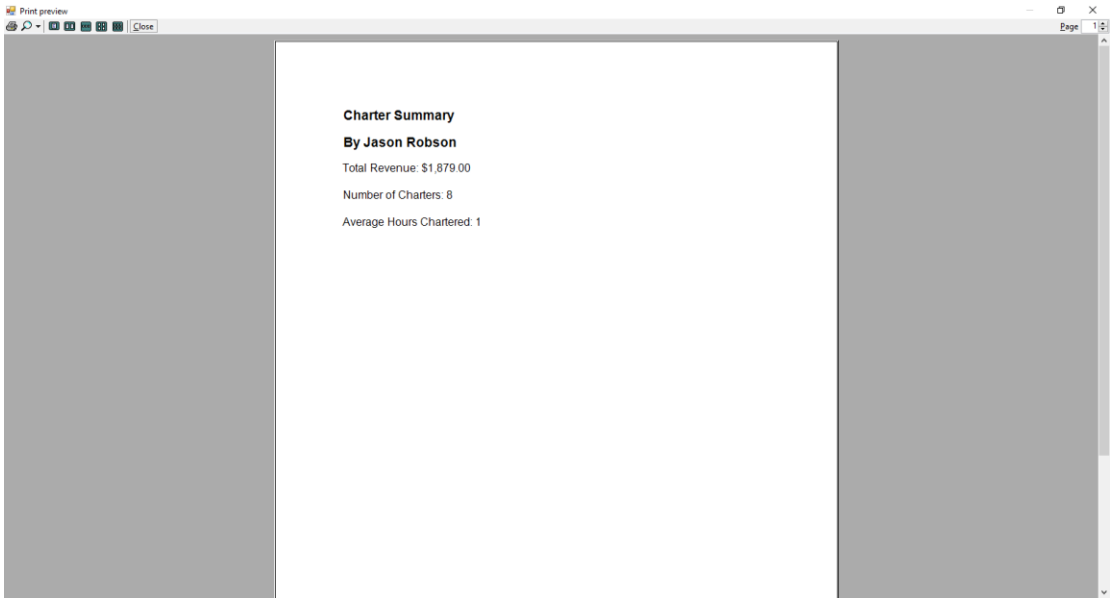
8.6 Yacht Size 45 Selected



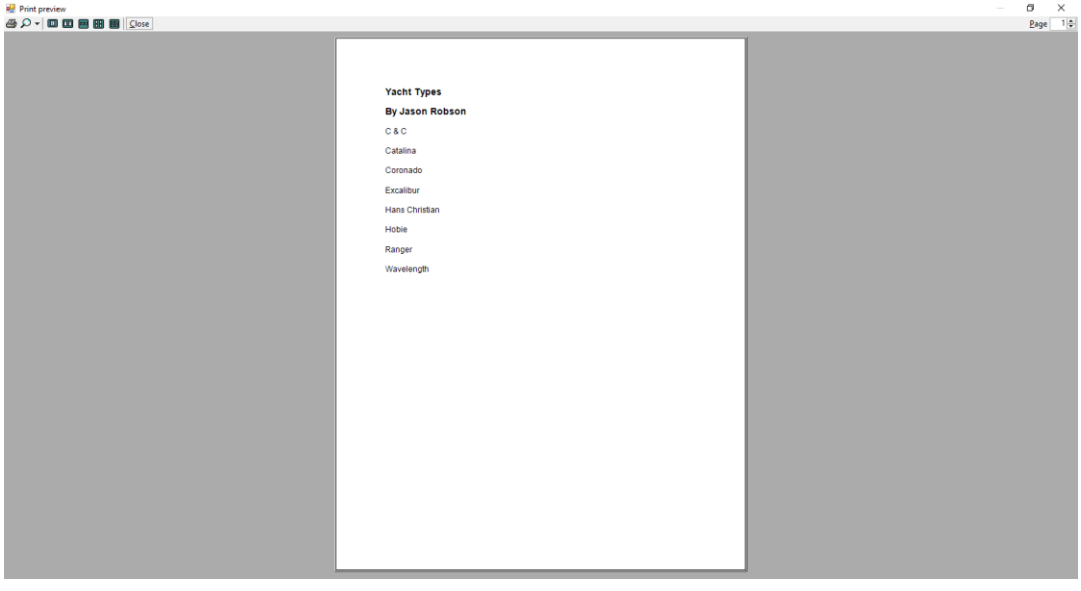
8.7 Yacht Size 22 Selected



9. Print Summary (Print Summary menu strip clicked)



10. Print Yacht Type (Print yacht type menu strip clicked)



11. Program Code

```
'Project : Yacht Chartering Information
'Programmer : Jason Robson
'Date : November 1, 2017
'Description : This project gives the price per charter and summarizes
the total revenue, average hours chartered, and charter count in a
printable summary report.

Public Class YachtCharteringInformationForm
    'Declare module-level constants
    Const YACHT_SIZE_22_HOURLY_RATE_Decimal As Decimal = 95D
    Const YACHT_SIZE_24_HOURLY_RATE_Decimal As Decimal = 137D
    Const YACHT_SIZE_30_HOURLY_RATE_Decimal As Decimal = 160D
    Const YACHT_SIZE_32_HOURLY_RATE_Decimal As Decimal = 192D
    Const YACHT_SIZE_36_HOURLY_RATE_Decimal As Decimal = 250D
    Const YACHT_SIZE_38_HOURLY_RATE_Decimal As Decimal = 400D
    Const YACHT_SIZE_45_HOURLY_RATE_Decimal As Decimal = 550D

    'Declare module-level variables
    Private CharterCountInteger, AverageHoursCharteredInteger As
Integer
    Private TotalRevenueDecimal As Decimal

    Private Function FindHourlyRate(ByVal HoursInteger As Integer) As
Decimal

        'Find the hourly rate
        With YachtSizeComboBox
            Select Case .SelectedIndex
                Case 6
                    Return YACHT_SIZE_45_HOURLY_RATE_Decimal *
HoursInteger
                Case 5
                    Return YACHT_SIZE_38_HOURLY_RATE_Decimal *
HoursInteger
                Case 4
                    Return YACHT_SIZE_36_HOURLY_RATE_Decimal *
HoursInteger
                Case 3
                    Return YACHT_SIZE_32_HOURLY_RATE_Decimal *
HoursInteger
                Case 2
                    Return YACHT_SIZE_30_HOURLY_RATE_Decimal *
HoursInteger
                Case 1
                    Return YACHT_SIZE_24_HOURLY_RATE_Decimal *
HoursInteger
                Case 0
                    Return YACHT_SIZE_22_HOURLY_RATE_Decimal *
HoursInteger
                    'Nothing selected
                Case Else
```

```
                    Return ""
                End Select
            End With
        End Function

        Private Sub OKButton_Click_1(sender As Object, e As EventArgs)
Handles OKButton.Click
            'Declare local-level variables
            Dim HoursInteger As Integer
            Static TotalHoursInteger
            Dim RevenueDecimal As Decimal

            'If customer name is entered, yacht type is selected/added, and
yacht size is selected, move to converting hours string to integer
            If CustomerNameTextBox.Text <> "" Then
                If YachtTypeComboBox.SelectedIndex <> -1 Then
                    If YachtSizeComboBox.SelectedIndex <> -1 Then
                        'Convert hour string into integer
                        Try
                            HoursInteger =
Integer.Parse(HoursCharteredTextBox.Text)
                            'If hours does not equal zero nor negative,
perform calcuations
                            If HoursInteger > 0 Then

                                'Calculations
                                TotalHoursInteger += HoursInteger
                                RevenueDecimal =
FindHourlyRate(HoursInteger)
                                CharterCountInteger += 1
                                TotalRevenueDecimal += RevenueDecimal
                                AverageHoursCharteredInteger =
TotalHoursInteger / CharterCountInteger

                                'Convert to string
                                PriceOfTheCharterTextBox.Text =
RevenueDecimal.ToString("C")

                                'If charter count is greater than one,
enable print summary
                                If CharterCountInteger > 1 Then
                                    PrintSummaryToolStripMenuItem.Enabled =
True
                                    'If charter count is less than or equal
to one, keep print summary disabled
                                ElseIf CharterCountInteger <= 1 Then
                                    PrintSummaryToolStripMenuItem.Enabled =
False
                                End If

                                'If hours is zero or negative, inform user
                            ElseIf HoursInteger <= 0 Then
```

```
                            MessageBox.Show("Hours cannot be zero nor
negative.", "Data entry error",
                                            MessageBoxButtons.OK,
MessageBoxIcon.Exclamation)
                            With HoursCharteredTextBox
                                .SelectAll()
                                .Focus()
                            End With
                        End If

                        'Catch non-numeric hours
                    Catch HoursException As FormatException

                        MessageBox.Show("Hours must be numeric.", "Data
entry error",
                                        MessageBoxButtons.OK,
MessageBoxIcon.Exclamation)
                        With HoursCharteredTextBox
                            .SelectAll()
                            .Focus()
                        End With
                    End Try

                    'If yacht size is not selected, inform user
                ElseIf YachtSizeComboBox.SelectedIndex = -1 Then

                    MessageBox.Show("Select a yacht size.", "Missing
data",
                                                            MessageBoxButtons.OK,
MessageBoxIcon.Exclamation)
                End If

                'If yacht type is not selected, inform user
            ElseIf YachtTypeComboBox.SelectedIndex = -1 Then

                MessageBox.Show("Select a yacht type.", "Missing data",
                                MessageBoxButtons.OK,
MessageBoxIcon.Exclamation)
            End If

            'If customer name is blank, display a message box
        ElseIf CustomerNameTextBox.Text = "" Then

            MessageBox.Show("Enter customer name.", "Missing data",
                            MessageBoxButtons.OK,
MessageBoxIcon.Exclamation)
            With CustomerNameTextBox
                .Focus()
            End With
        End If
    End Sub

    Private Sub ExitToolStripMenuItem_Click(sender As Object, e As
EventArgs) Handles ExitToolStripMenuItem.Click
```

```
        'Close the form
        Me.Close()
    End Sub


    Private Sub ClearButton_Click(sender As Object, e As EventArgs)
Handles ClearButton.Click
        ' Clear the form.
        Dim ResponseDialogResult As DialogResult

        ResponseDialogResult = MessageBox.Show("Clear the form?",
          "Clear all items", MessageBoxButtons.YesNo,
MessageBoxIcon.Question)
        'If yes, clear the form and reset the focus
        If ResponseDialogResult = DialogResult.Yes Then
            YachtTypeComboBox.SelectedIndex = -1
            YachtSizeComboBox.SelectedIndex = -1
            HoursCharteredTextBox.Clear()
            PriceOfTheCharterTextBox.Clear()
            With CustomerNameTextBox
                .Clear()
                .Focus()
            End With
        End If
    End Sub

    Private Sub ClearForNextCharterToolStripMenuItem_Click(sender As
Object, e As EventArgs) Handles
ClearForNextCharterToolStripMenuItem.Click
        ' Clear the form.
        Dim ResponseDialogResult As DialogResult

        ResponseDialogResult = MessageBox.Show("Clear the form?",
          "Clear all items", MessageBoxButtons.YesNo,
MessageBoxIcon.Question)
        'If yes, clear the form and reset the focus
        If ResponseDialogResult = DialogResult.Yes Then
            YachtTypeComboBox.SelectedIndex = -1
            YachtSizeComboBox.SelectedIndex = -1
            HoursCharteredTextBox.Clear()
            PriceOfTheCharterTextBox.Clear()
            With CustomerNameTextBox
                .Clear()
                .Focus()
            End With
        End If
    End Sub

    Private Sub AddYachtTypeToolStripMenuItem_Click(sender As Object, e
As EventArgs) Handles AddYachtTypeToolStripMenuItem.Click
        ' Add a new yacht type to the yacht list.

        With YachtTypeComboBox
            ' Test for blank input.
```

```
            If .Text <> "" Then
                ' Make sure item is not already on the list.
                Dim ItemFoundBoolean As Boolean
                Dim ItemIndexInteger As Integer
                Do Until ItemFoundBoolean Or ItemIndexInteger =
.Items.Count
                    If .Text = .Items(ItemIndexInteger).ToString() Then
                        ItemFoundBoolean = True
                        Exit Do
                    Else
                        ItemIndexInteger += 1
                    End If
                Loop
                'If item is on the list
                If ItemFoundBoolean Then
                    MessageBox.Show("Duplicate item.", "Add Failed",
                     MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
                Else
                    ' If it's not in the list, add it.
                    .Items.Add(.Text)
                    .Text = ""
                End If
                'If input is blank
            Else
                MessageBox.Show("Enter a yacht type to add",
                  "Missing Data", MessageBoxButtons.OK,
                MessageBoxIcon.Exclamation)
            End If
            .Focus()
        End With
    End Sub

    Private Sub RemoveYachtTypeToolStripMenuItem_Click(sender As
Object, e As EventArgs) Handles RemoveYachtTypeToolStripMenuItem.Click
        'If item is selected, remove
        With YachtTypeComboBox
            If .SelectedIndex <> -1 Then
                .Items.RemoveAt(.SelectedIndex)
                'If no item selected, display message
            Else
                MessageBox.Show("Select yacht to remove.", "No
selection made",
                                MessageBoxButtons.OK,
MessageBoxIcon.Exclamation)
            End If
        End With
    End Sub

    Private Sub DisplayCountOfYachtTypesToolStripMenuItem_Click(sender
As Object, e As EventArgs) Handles
DisplayCountOfYachtTypesToolStripMenuItem.Click
        'Display count of yacht list
        Dim MessageString As String
```

```
        MessageString = "The number of yachts is " &
YachtTypeComboBox.Items.Count & "."
        MessageBox.Show(MessageString, "Yacht Chartering Information
Count",
                        MessageBoxButtons.OK,
MessageBoxIcon.Information)
    End Sub

    Private Sub PrintSummaryToolStripMenuItem_Click(sender As Object, e
As EventArgs) Handles PrintSummaryToolStripMenuItem.Click
        'Begin the process for print preview of summary report
        PrintPreviewDialog1.Document = PrintSummaryPrintDocument
        PrintPreviewDialog1.ShowDialog()
    End Sub

    Private Sub PrintSummaryPrintDocument_PrintPage(ByVal sender As
System.Object, ByVal e As System.Drawing.Printing.PrintPageEventArgs)
Handles PrintSummaryPrintDocument.PrintPage
        ' Handle printing and print preview when printing summary
report page.
        Dim PrintFont As New Font("Arial", 12)
        Dim LineHeightSingle As Single = PrintFont.GetHeight + 2
        Dim HorizontalPrintLocationSingle As Single =
e.MarginBounds.Left
        Dim VerticalPrintLocationSingle As Single = e.MarginBounds.Top
        Dim HeadingFont As New Font("Arial", 14, FontStyle.Bold)

        ' Set up and display heading lines.
        e.Graphics.DrawString("Charter Summary", HeadingFont,
                              Brushes.Black,
HorizontalPrintLocationSingle, VerticalPrintLocationSingle)
        VerticalPrintLocationSingle += LineHeightSingle * 2
        e.Graphics.DrawString("By Jason Robson", HeadingFont,
                              Brushes.Black,
HorizontalPrintLocationSingle, VerticalPrintLocationSingle)
        'Leave a blank line between heading and detail line
        VerticalPrintLocationSingle += LineHeightSingle * 2
        e.Graphics.DrawString("Total Revenue: " &
TotalRevenueDecimal.ToString("C"), PrintFont,
                              Brushes.Black,
HorizontalPrintLocationSingle, VerticalPrintLocationSingle)

        VerticalPrintLocationSingle += LineHeightSingle * 2
        e.Graphics.DrawString("Number of Charters: " &
CharterCountInteger.ToString(), PrintFont,
                              Brushes.Black,
HorizontalPrintLocationSingle, VerticalPrintLocationSingle)

        VerticalPrintLocationSingle += LineHeightSingle * 2
        e.Graphics.DrawString("Average Hours Chartered: " &
AverageHoursCharteredInteger.ToString(), PrintFont,
                              Brushes.Black,
HorizontalPrintLocationSingle, VerticalPrintLocationSingle)
    End Sub
```

```
    Private Sub PrintYachtTypesToolStripMenuItem_Click(sender As
Object, e As EventArgs) Handles PrintYachtTypesToolStripMenuItem.Click
        'Begin the process for print preview of yacht types
        PrintPreviewDialog1.Document = PrintYachtTypePrintDocument
        PrintPreviewDialog1.ShowDialog()
    End Sub


    Private Sub PrintYachtTypePrintDocument_PrintPage(ByVal sender As
System.Object, ByVal e As System.Drawing.Printing.PrintPageEventArgs)
Handles PrintYachtTypePrintDocument.PrintPage
        'Handles printing and print preview of all yacht types

        Dim PrintFont As New Font("Arial", 12)
        Dim LineHeightSingle As Single = PrintFont.GetHeight + 2
        Dim HorizontalPrintLocationSingle As Single =
e.MarginBounds.Left
        Dim VerticalPrintLocationSingle As Single = e.MarginBounds.Top
        Dim HeadingFont As New Font("Arial", 14, FontStyle.Bold)
        Dim PrintLineString As String

        ' Set up and display heading lines.
        e.Graphics.DrawString("Yacht Types", HeadingFont,
                              Brushes.Black,
HorizontalPrintLocationSingle, VerticalPrintLocationSingle)
        VerticalPrintLocationSingle += LineHeightSingle * 2
        e.Graphics.DrawString("By Jason Robson", HeadingFont,
                              Brushes.Black,
HorizontalPrintLocationSingle, VerticalPrintLocationSingle)
        'Loop through the entire list
        For ListIndexInteger As Integer = 0 To
YachtTypeComboBox.Items.Count - 1
            'Increment the Y position for the next line
            VerticalPrintLocationSingle += LineHeightSingle * 2
            'Set up a line
            PrintLineString =
YachtTypeComboBox.Items(ListIndexInteger).ToString()
            'Send the line to the graphics page object
            e.Graphics.DrawString(PrintLineString, PrintFont,
Brushes.Black, HorizontalPrintLocationSingle,
VerticalPrintLocationSingle)
        Next ListIndexInteger
    End Sub

    Private Sub AboutToolStripMenuItem_Click(sender As Object, e As
EventArgs) Handles AboutToolStripMenuItem.Click
        'Show about box
        YachtCharteringInformationAboutBox.ShowDialog()
    End Sub
End Class
```

BIS 628 – Application Development

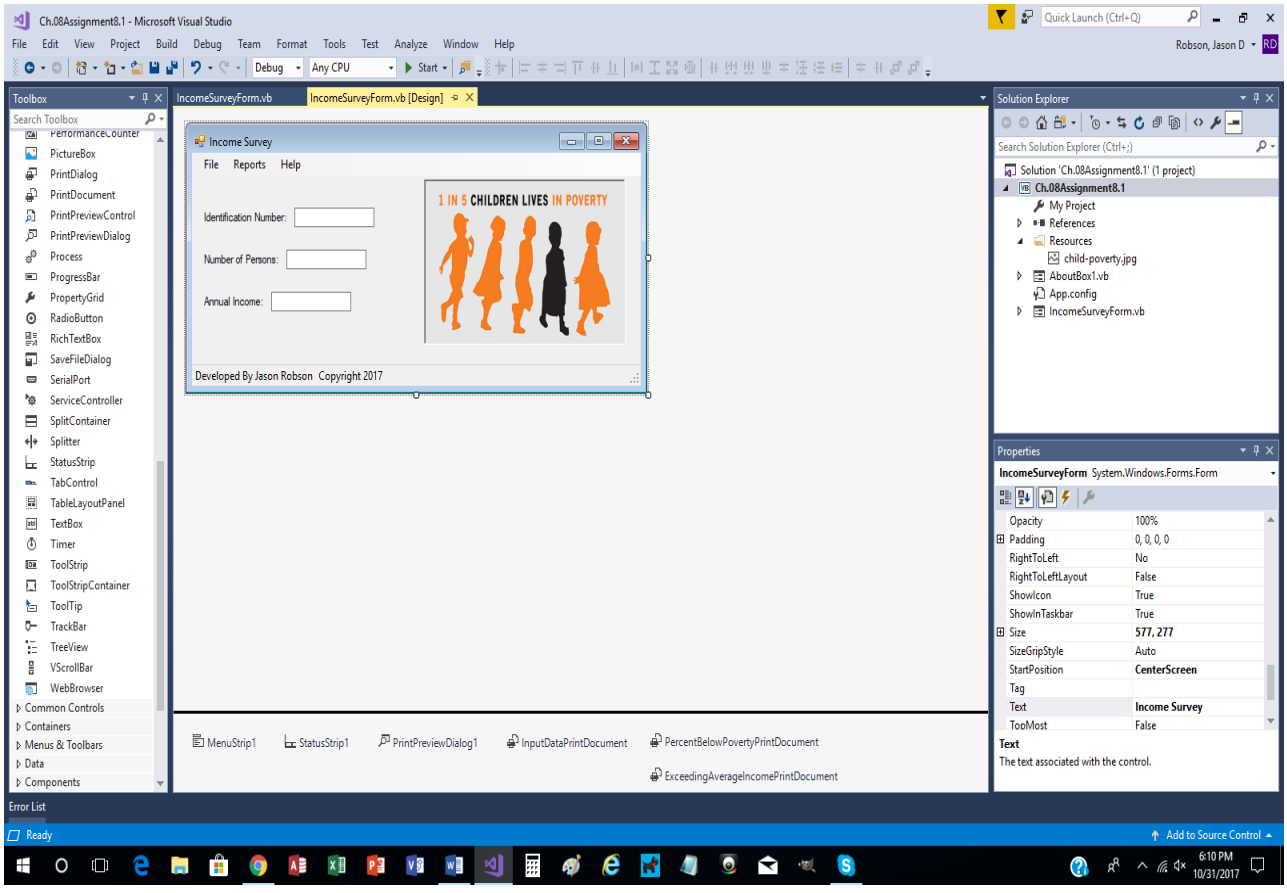
Assignment 8

- Example 8.1 (page 355)

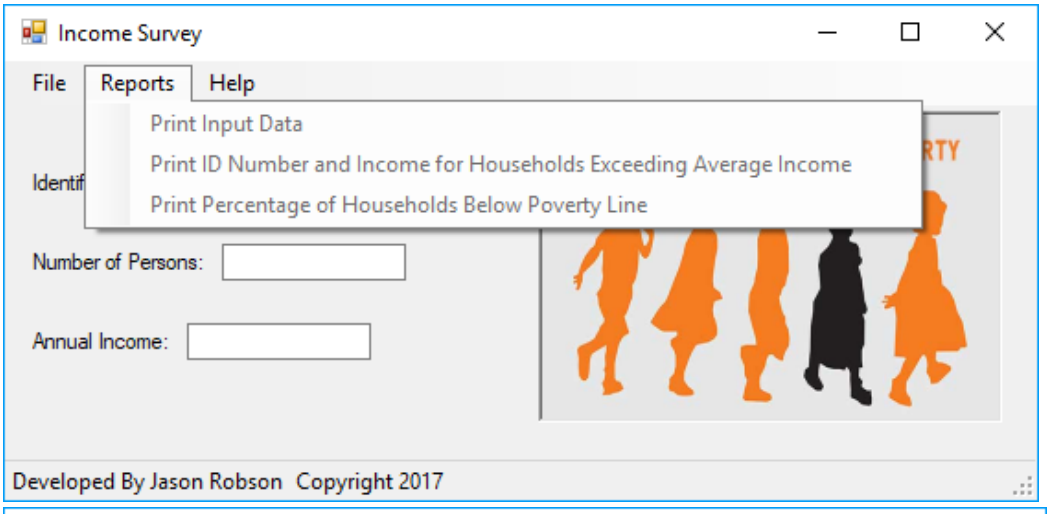
Due Date: 11/13/2017 11:59 pm

By: Jason Robson

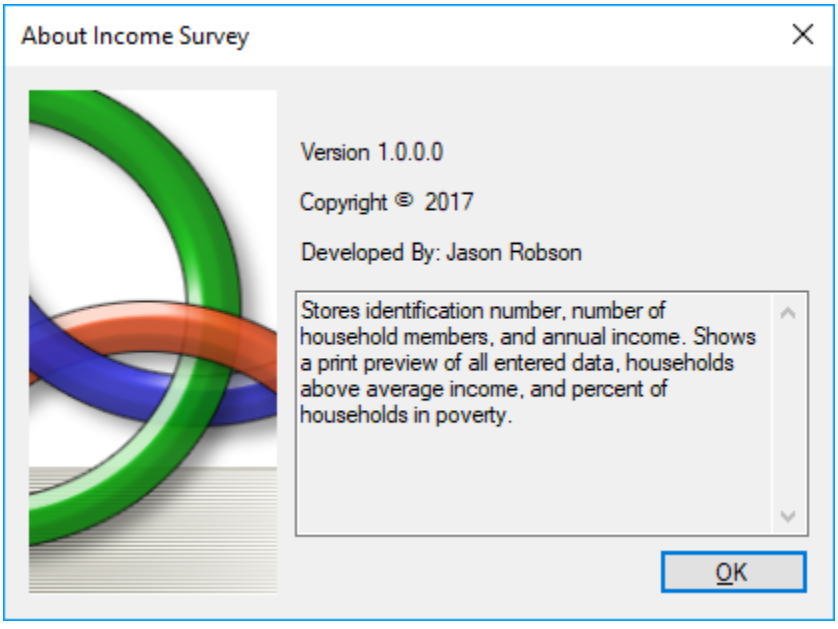
1. Design View



2. Menu Strip (Menu strip clicked)



3. About Box (About box clicked in menu strip)

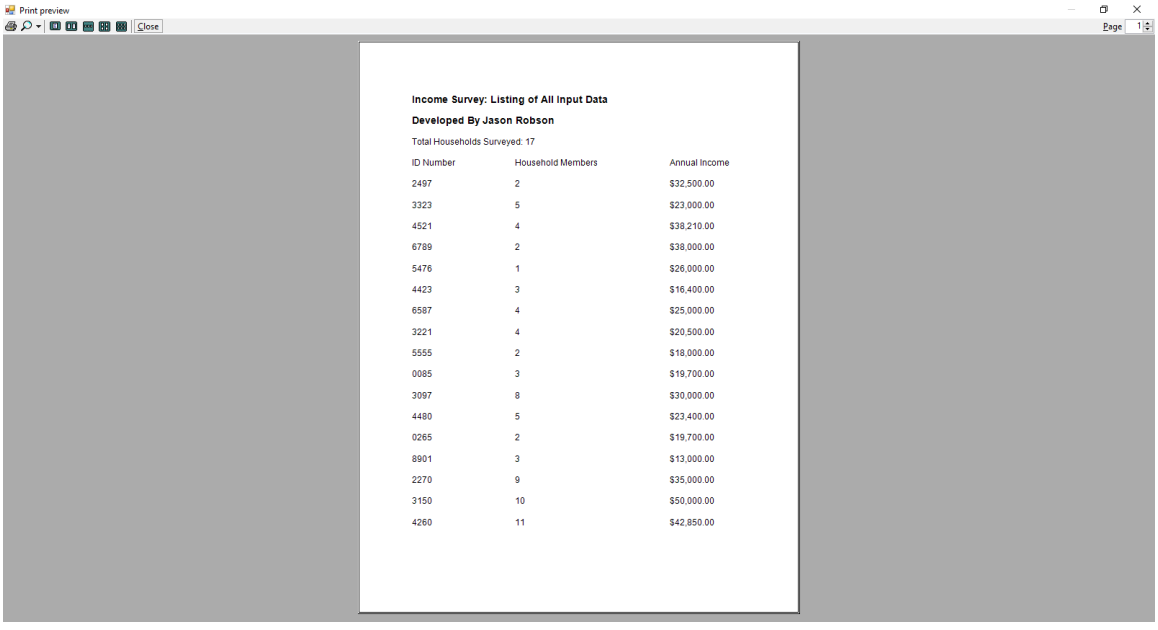


4. Test Data (Enter data clicked)


Income Survey
File
Reports
Help
Identification Number:
3323
Number of Persons:
5
Annual Income:
23000
1 IN 5 CHILDREN LIVES IN POVERTY
Developed By Jason Robson Copyright 2017


Income Survey
File
Reports
Help
Identification Number:
4521
Number of Persons:
4
Annual Income:
38210
1 IN 5 CHILDREN LIVES IN POVERTY
Developed By Jason Robson Copyright 2017

5. Print Preview All Input Data



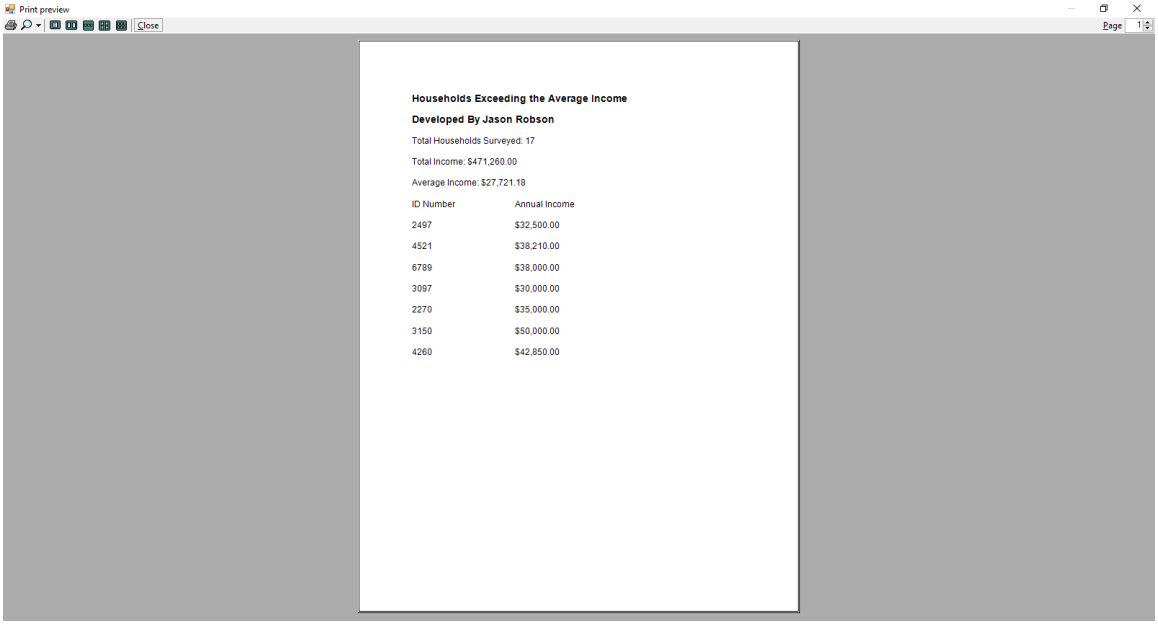
Income Survey: Listing of All Input Data

Developed By Jason Robson

Total Households Surveyed: 17

ID Number	Household Members	Annual Income
2497	2	$32,500.00
3323	5	$23,000.00
4521	4	$38,210.00
6789	2	$38,000.00
5476	1	$26,000.00
4423	3	$16,400.00
6587	4	$25,000.00
3221	4	$20,500.00
5555	2	$18,000.00
0085	3	$19,700.00
3097	8	$30,000.00
4480	5	$23,400.00
0265	2	$19,700.00
8901	3	$13,000.00
2270	9	$35,000.00
3150	10	$50,000.00
4260	11	$42,850.00

6. Print Preview Households Above Average Household Income



Households Exceeding the Average Income

Developed By Jason Robson

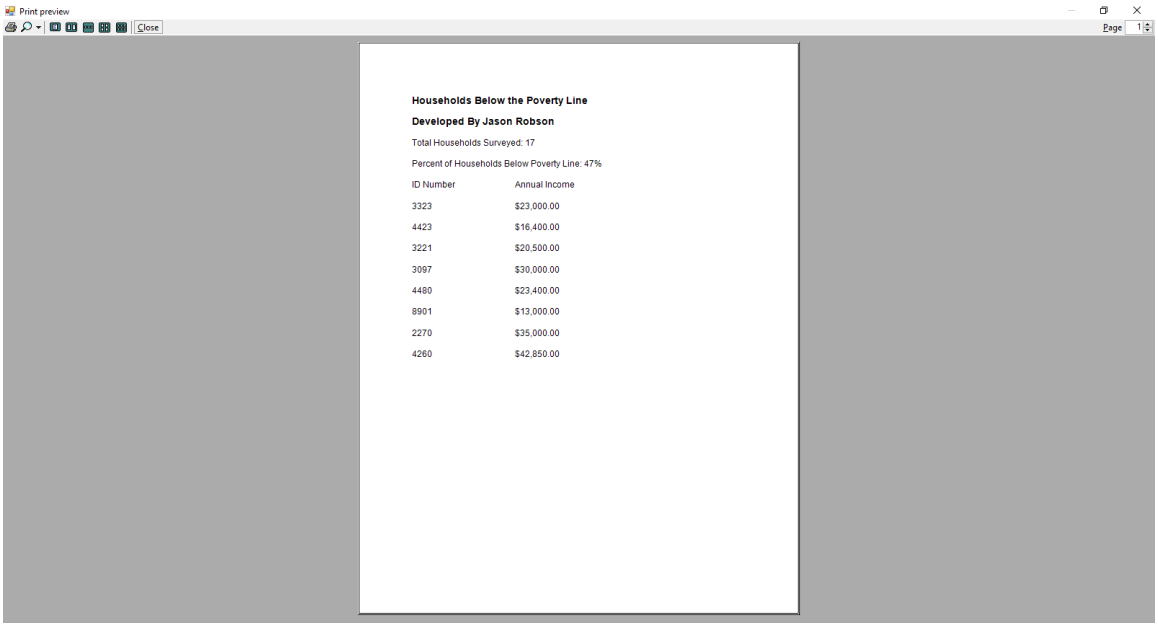
Total Households Surveyed: 17

Total Income: $471,260.00

Average Income: $27,721.18

ID Number	Annual Income
2497	$32,500.00
4521	$38,210.00
6789	$38,000.00
3097	$30,000.00
2270	$35,000.00
3150	$50,000.00
4260	$42,850.00

7. Print Preview of Percentage of Households in Poverty



8. Program Code

```
'Project     : Income Survey
'Programmer  : Jason Robson
'Date        : November 8, 2017
'Description : This program stores identification number, number of
household members, and annual income. It prints a report containing all
input data, households above the average household income, and percent
of households in poverty.

Public Class IncomeSurveyForm
    'Declare constants
    Const POVERTY_INCOME_1_Decimal As Decimal = 10210D
    Const POVERTY_INCOME_2_Decimal As Decimal = 13690D
    Const POVERTY_INCOME_3_Decimal As Decimal = 17170D
    Const POVERTY_INCOME_4_Decimal As Decimal = 20650D
    Const POVERTY_INCOME_5_Decimal As Decimal = 24130D
    Const POVERTY_INCOME_6_Decimal As Decimal = 27610D
    Const POVERTY_INCOME_7_Decimal As Decimal = 31090D
    Const POVERTY_INCOME_8_Decimal As Decimal = 34570D

    'Declare structure and module-level variables
    Structure FamilyData
        Dim IdentificationNumberString As String
        Dim NumberOfPersonsInteger As Integer
        Dim AnnualIncomeDecimal As Decimal
```

```
    End Structure

    Private NewFamily(16) As FamilyData
    Private FamilyCountInteger As Integer
    Private TotalIncomeDecimal As Decimal
    Private AverageIncomeDecimal As Decimal
    Private PovertyPercentDecimal As Decimal
    Private SurveyCountInteger As Integer

    Private Function FindPovertyIncome(ByVal NumberofPersonsInteger As
Integer) As Decimal
        'Select poverty income by household number
        Select Case NumberofPersonsInteger
            Case 1
                Return POVERTY_INCOME_1_Decimal
            Case 2
                Return POVERTY_INCOME_2_Decimal
            Case 3
                Return POVERTY_INCOME_3_Decimal
            Case 4
                Return POVERTY_INCOME_4_Decimal
            Case 5
                Return POVERTY_INCOME_5_Decimal
            Case 6
                Return POVERTY_INCOME_6_Decimal
            Case 7
                Return POVERTY_INCOME_7_Decimal
            Case 8
                Return POVERTY_INCOME_8_Decimal
            Case Else
                Return (34570 + ((NumberofPersonsInteger - 8) * 3480))
        End Select

    End Function

    Private Sub ExitToolStripMenuItem_Click(sender As Object, e As
EventArgs) Handles ExitToolStripMenuItem.Click
        'Close the form
        Me.Close()
    End Sub

    Private Sub AboutToolStripMenuItem_Click(sender As Object, e As
EventArgs) Handles AboutToolStripMenuItem.Click
        'Show about box
        IncomeSurveyAboutBox.ShowDialog()
    End Sub

    Private Sub EnterDataToolStripMenuItem_Click(sender As Object, e As
EventArgs) Handles EnterDataToolStripMenuItem.Click
        'Declare local-level variables
        Static NormalHouseholdCountInteger,
PovertyHouseholdCountInteger, TotalHouseholdCountInteger As Integer
        Dim AnnualIncomeDecimal, PovertyIncomeDecimal As Decimal
        Dim NumberOfPersonsInteger As Integer
```

```
        Dim IdentificationNumberString As String

        'If family count integer is less than 14, proceed
        If FamilyCountInteger <= 17 Then

            'If identification number textbox is not blank, proceed
            If IdentificationNumberTextBox.Text <> "" Then

                'If identification number is greater than or equal to
zero, proceed
                If IdentificationNumberTextBox.Text >= 0 Then
                    IdentificationNumberString =
IdentificationNumberTextBox.Text

                    'If number of people in household is not blank,
proceed
                    If NumberOfPersonsTextBox.Text <> "" Then
                        'Convert to integer from string
                        Try
                            NumberOfPersonsInteger =
Integer.Parse(NumberOfPersonsTextBox.Text)

                            'If number of people in household is
greater than zero, proceed
                            If NumberOfPersonsTextBox.Text > 0 Then

                                'If annual income is not blank, proceed
                                If AnnualIncomeTextBox.Text <> "" Then
                                    'Convert to decimal from string
                                    Try
                                        AnnualIncomeDecimal =
Decimal.Parse(AnnualIncomeTextBox.Text)

                                        'If annual income is greater
than zero, proceed to calculations
                                        If AnnualIncomeTextBox.Text > 0
Then

                                            'Calculations
                                            PovertyIncomeDecimal =
FindPovertyIncome(NumberOfPersonsInteger)
                                            TotalIncomeDecimal +=
AnnualIncomeDecimal
                                            SurveyCountInteger += 1
                                            AverageIncomeDecimal =
TotalIncomeDecimal / SurveyCountInteger

                                            'Save data

NewFamily(FamilyCountInteger).IdentificationNumberString =
IdentificationNumberString

NewFamily(FamilyCountInteger).NumberOfPersonsInteger =
NumberOfPersonsInteger
```

```
NewFamily(FamilyCountInteger).AnnualIncomeDecimal = AnnualIncomeDecimal
                                                FamilyCountInteger += 1

                                                'Determine if family is
within poverty or normal income
                                                If AnnualIncomeDecimal <=
PovertyIncomeDecimal Then
                                                    'Poverty count goes up

PovertyHouseholdCountInteger += 1
                                                ElseIf AnnualIncomeDecimal
> PovertyIncomeDecimal Then
                                                    'Normal income goes up

NormalHouseholdCountInteger += 1
                                                End If

                                                'Continue calculations
                                                TotalHouseholdCountInteger
= NormalHouseholdCountInteger + PovertyHouseholdCountInteger
                                                PovertyPercentDecimal =
(TotalHouseholdCountInteger - NormalHouseholdCountInteger) /
TotalHouseholdCountInteger

                                                'Enable print summaries
                                                If SurveyCountInteger > 1
Then

PrintIDNumberAndToolStripMenuItem.Enabled = True

PrintInputDataToolStripMenuItem.Enabled = True

PrintPercentageOfHouseholdsBelowPovertyLineToolStripMenuItem.Enabled =
True
                                                End If

                                                'Clear the form
                                                AnnualIncomeTextBox.Clear()

NumberOfPersonsTextBox.Clear()
                                                With
IdentificationNumberTextBox
                                                    .Clear()
                                                    .Focus()
                                                End With


                                                'If annual income is zero
or negative, inform user
                                            ElseIf AnnualIncomeTextBox.Text
<= 0 Then
                                                MessageBox.Show("Annual
income cannot be zero nor negative.", "Data entry error",
```

```
MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
                                            With AnnualIncomeTextBox
                                                .Focus()
                                                .SelectAll()
                                            End With
                                        End If

                                        'If annual income is not
numeric, inform user
                                    Catch AnnualIncomeException As
FormatException
                                        MessageBox.Show("Annual income
must be numeric.", "Data entry error",

MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
                                        With AnnualIncomeTextBox
                                            .Focus()
                                            .SelectAll()
                                        End With
                                    End Try

                                    'If annual income is blank, inform
user
                                Else
                                    MessageBox.Show("Please enter
annual income.", "Missing data",

MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
                                    With AnnualIncomeTextBox
                                        .Focus()
                                    End With
                                End If

                                'If number of household members is
equal to or less than zero, inform user
                            ElseIf NumberOfPersonsTextBox.Text <= 0
Then
                                MessageBox.Show("Number of persons
cannot be zero nor negative.", "Data entry error",

MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
                                With NumberOfPersonsTextBox
                                    .Focus()
                                    .SelectAll()
                                End With
                            End If

                            'If number of household members is not
numeric, inform user
                        Catch NumberOfPersonsException As
FormatException
                            MessageBox.Show("Number of household
members must be numeric.", "Data entry error",
```

```
                                                    MessageBoxButtons.OK,
MessageBoxIcon.Exclamation)
                            With NumberOfPersonsTextBox
                                .Focus()
                                .SelectAll()
                            End With
                        End Try

                        'If number of household members is blank,
inform user
                    Else
                        MessageBox.Show("Please enter number of
household members.", "Missing data",
                                                    MessageBoxButtons.OK,
MessageBoxIcon.Exclamation)
                        With NumberOfPersonsTextBox
                            .Focus()
                        End With
                    End If

                    'If identification number is negative, inform user
                Else
                    MessageBox.Show("Identification number cannot be
negative.", "Data entry error",
                                                    MessageBoxButtons.OK,
MessageBoxIcon.Exclamation)
                    With IdentificationNumberTextBox
                        .Focus()
                        .SelectAll()
                    End With
                End If

                'If identification number is blank, inform user
            Else
                MessageBox.Show("Please enter identification number.",
"Missing data",
                                        MessageBoxButtons.OK,
MessageBoxIcon.Exclamation)
                With IdentificationNumberTextBox
                    .Focus()
                End With
            End If

            'If family count integer is full, display message
        ElseIf FamilyCountInteger > 17 Then

            MessageBox.Show("Data is full.", "Excess Data",
                        MessageBoxButtons.OK,
MessageBoxIcon.Exclamation)
        End If
    End Sub

    Private Sub PrintInputDataToolStripMenuItem_Click(sender As Object,
e As EventArgs) Handles PrintInputDataToolStripMenuItem.Click
```

```
        'Begin the print process to print all input data
        PrintPreviewDialog1.Document = InputDataPrintDocument
        PrintPreviewDialog1.ShowDialog()
    End Sub

    Private Sub InputDataPrintDocument_PrintPage(ByVal sender As
Object, ByVal e As System.Drawing.Printing.PrintPageEventArgs) Handles
InputDataPrintDocument.PrintPage
        'Handle printing and print preview for all input data

        Dim PrintFont As New Font("Arial", 12)
        Dim LineHeightSingle As Single = PrintFont.GetHeight + 2
        Dim HeadingFont As New Font("Arial", 14, FontStyle.Bold)
        Dim Column1HorizontalLocationSingle As Single =
e.MarginBounds.Left
        Dim Column2HorizontalLocationSingle As Single = 300
        Dim Column3HorizontalLocationSingle As Single = 600
        Dim VerticalPrintLocationSingle As Single = e.MarginBounds.Top
        Dim PrintLineString As String

        'Report title and developer
        PrintLineString = "Income Survey: Listing of All Input Data"
        e.Graphics.DrawString(PrintLineString, HeadingFont,
                              Brushes.Black,
Column1HorizontalLocationSingle, VerticalPrintLocationSingle)
        VerticalPrintLocationSingle += LineHeightSingle * 2
        PrintLineString = "Developed By Jason Robson"
        e.Graphics.DrawString(PrintLineString, HeadingFont,
                              Brushes.Black,
Column1HorizontalLocationSingle, VerticalPrintLocationSingle)
        VerticalPrintLocationSingle += LineHeightSingle * 2

        'Total households surveyed
        e.Graphics.DrawString("Total Households Surveyed: " &
SurveyCountInteger.ToString(), PrintFont,
                              Brushes.Black,
Column1HorizontalLocationSingle, VerticalPrintLocationSingle)
        VerticalPrintLocationSingle += LineHeightSingle * 2

        'Column headings
        e.Graphics.DrawString("ID Number", PrintFont,
                              Brushes.Black,
Column1HorizontalLocationSingle, VerticalPrintLocationSingle)

        e.Graphics.DrawString("Household Members", PrintFont,
                              Brushes.Black,
Column2HorizontalLocationSingle, VerticalPrintLocationSingle)

        e.Graphics.DrawString("Annual Income", PrintFont,
                              Brushes.Black,
Column3HorizontalLocationSingle, VerticalPrintLocationSingle)
        VerticalPrintLocationSingle += LineHeightSingle * 2

        'Loop through input data
```

```
        For Each IndexInteger As FamilyData In NewFamily
            'Don't print if blank
            If IndexInteger.IdentificationNumberString <> "" Then

e.Graphics.DrawString(IndexInteger.IdentificationNumberString,
PrintFont,
                                      Brushes.Black,
Column1HorizontalLocationSingle, VerticalPrintLocationSingle)

e.Graphics.DrawString(IndexInteger.NumberOfPersonsInteger, PrintFont,
                                      Brushes.Black,
Column2HorizontalLocationSingle, VerticalPrintLocationSingle)

e.Graphics.DrawString(IndexInteger.AnnualIncomeDecimal.ToString("C"),
PrintFont,
                                      Brushes.Black,
Column3HorizontalLocationSingle, VerticalPrintLocationSingle)
                VerticalPrintLocationSingle += LineHeightSingle * 2
            End If
        Next IndexInteger
    End Sub

    Private Sub PrintIDNumberAndToolStripMenuItem_Click(sender As
Object, e As EventArgs) Handles PrintIDNumberAndToolStripMenuItem.Click
        'Begin process to print income exceeding average income
        PrintPreviewDialog1.Document =
ExceedingAverageIncomePrintDocument
        PrintPreviewDialog1.ShowDialog()
    End Sub

    Private Sub ExceedingAverageIncomePrintDocument_PrintPage(ByVal
sender As Object, ByVal e As
System.Drawing.Printing.PrintPageEventArgs) Handles
ExceedingAverageIncomePrintDocument.PrintPage
        'Handle printing and print preview for income exceeding average
income

        Dim PrintFont As New Font("Arial", 12)
        Dim LineHeightSingle As Single = PrintFont.GetHeight + 2
        Dim HeadingFont As New Font("Arial", 14, FontStyle.Bold)
        Dim Column1HorizontalLocationSingle As Single =
e.MarginBounds.Left
        Dim Column2HorizontalLocationSingle As Single = 300
        Dim VerticalPrintLocationSingle As Single = e.MarginBounds.Top
        Dim PrintLineString As String

        'Report title and developer
        PrintLineString = "Households Exceeding the Average Income"
        e.Graphics.DrawString(PrintLineString, HeadingFont,
                              Brushes.Black,
Column1HorizontalLocationSingle, VerticalPrintLocationSingle)
        VerticalPrintLocationSingle += LineHeightSingle * 2
        PrintLineString = "Developed By Jason Robson"
        e.Graphics.DrawString(PrintLineString, HeadingFont,
```

```
                                Brushes.Black,
Column1HorizontalLocationSingle, VerticalPrintLocationSingle)
        VerticalPrintLocationSingle += LineHeightSingle * 2

        'Total households surveyed
        e.Graphics.DrawString("Total Households Surveyed: " &
SurveyCountInteger.ToString(), PrintFont,
                                Brushes.Black,
Column1HorizontalLocationSingle, VerticalPrintLocationSingle)
        VerticalPrintLocationSingle += LineHeightSingle * 2

        'Total Income
        e.Graphics.DrawString("Total Income: " &
TotalIncomeDecimal.ToString("C"), PrintFont,
                                Brushes.Black,
Column1HorizontalLocationSingle, VerticalPrintLocationSingle)
        VerticalPrintLocationSingle += LineHeightSingle * 2

        'Average Income
        e.Graphics.DrawString("Average Income: " &
AverageIncomeDecimal.ToString("C"), PrintFont,
                                Brushes.Black,
Column1HorizontalLocationSingle, VerticalPrintLocationSingle)
        VerticalPrintLocationSingle += LineHeightSingle * 2

        'Column headings
        e.Graphics.DrawString("ID Number", PrintFont,
                                Brushes.Black,
Column1HorizontalLocationSingle, VerticalPrintLocationSingle)
        e.Graphics.DrawString("Annual Income", PrintFont,
                                Brushes.Black,
Column2HorizontalLocationSingle, VerticalPrintLocationSingle)
        VerticalPrintLocationSingle += LineHeightSingle * 2

        'Loop to find annual income exceeding average income
        For Each IndexInteger As FamilyData In NewFamily
            'If greater than average income decimal, loop
            If IndexInteger.AnnualIncomeDecimal > AverageIncomeDecimal
Then
                'Don't print blank
                If IndexInteger.IdentificationNumberString <> "" Then

e.Graphics.DrawString(IndexInteger.IdentificationNumberString,
PrintFont,
                                                    Brushes.Black,
Column1HorizontalLocationSingle, VerticalPrintLocationSingle)

e.Graphics.DrawString(IndexInteger.AnnualIncomeDecimal.ToString("C"),
PrintFont,
                                                Brushes.Black,
Column2HorizontalLocationSingle, VerticalPrintLocationSingle)
                    VerticalPrintLocationSingle += LineHeightSingle * 2
                End If
            End If
```

```
        Next IndexInteger
    End Sub

    Private Sub
PrintPercentageOfHouseholdsBelowPovertyLineToolStripMenuItem_Click(send
er As Object, e As EventArgs) Handles
PrintPercentageOfHouseholdsBelowPovertyLineToolStripMenuItem.Click
        'Begin process to print percent below poverty
        PrintPreviewDialog1.Document = PercentBelowPovertyPrintDocument
        PrintPreviewDialog1.ShowDialog()
    End Sub

    Private Sub PercentBelowPovertyPrintDocument_PrintPage(ByVal sender
As Object, ByVal e As System.Drawing.Printing.PrintPageEventArgs)
Handles PercentBelowPovertyPrintDocument.PrintPage
        'Handle printing and print preview for income poverty percent

        Dim PrintFont As New Font("Arial", 12)
        Dim LineHeightSingle As Single = PrintFont.GetHeight + 2
        Dim HeadingFont As New Font("Arial", 14, FontStyle.Bold)
        Dim Column1HorizontalLocationSingle As Single =
e.MarginBounds.Left
        Dim Column2HoriztonalLocationSingle As Single = 300
        Dim VerticalPrintLocationSingle As Single = e.MarginBounds.Top
        Dim PrintLineString As String

        'Report title and developer
        PrintLineString = "Households Below the Poverty Line"
        e.Graphics.DrawString(PrintLineString, HeadingFont,
                              Brushes.Black,
Column1HorizontalLocationSingle, VerticalPrintLocationSingle)
        VerticalPrintLocationSingle += LineHeightSingle * 2
        PrintLineString = "Developed By Jason Robson"
        e.Graphics.DrawString(PrintLineString, HeadingFont,
                              Brushes.Black,
Column1HorizontalLocationSingle, VerticalPrintLocationSingle)
        VerticalPrintLocationSingle += LineHeightSingle * 2

        'Total households surveyed
        e.Graphics.DrawString("Total Households Surveyed: " &
SurveyCountInteger.ToString(), PrintFont,
                              Brushes.Black,
Column1HorizontalLocationSingle, VerticalPrintLocationSingle)
        VerticalPrintLocationSingle += LineHeightSingle * 2

        'Percent of households in poverty
        e.Graphics.DrawString("Percent of Households Below Poverty
Line: " & PovertyPercentDecimal.ToString("P0"), PrintFont,
                              Brushes.Black,
Column1HorizontalLocationSingle, VerticalPrintLocationSingle)
        VerticalPrintLocationSingle += LineHeightSingle * 2

        'Column headings
```

```
        e.Graphics.DrawString("ID Number", PrintFont,
                              Brushes.Black,
Column1HorizontalLocationSingle, VerticalPrintLocationSingle)
        e.Graphics.DrawString("Annual Income", PrintFont,
                              Brushes.Black,
Column2HoriztonalLocationSingle, VerticalPrintLocationSingle)
        VerticalPrintLocationSingle += LineHeightSingle * 2

        'Loop to find households below poverty line
        For Each IndexInteger As FamilyData In NewFamily
            'If household is less than poverty line, loop
            If IndexInteger.AnnualIncomeDecimal <
FindPovertyIncome(IndexInteger.NumberOfPersonsInteger) Then
                'Don't print blank
                If IndexInteger.IdentificationNumberString <> "" Then

e.Graphics.DrawString(IndexInteger.IdentificationNumberString,
PrintFont,
                                                  Brushes.Black,
Column1HorizontalLocationSingle, VerticalPrintLocationSingle)

e.Graphics.DrawString(IndexInteger.AnnualIncomeDecimal.ToString("C"),
PrintFont,
                                        Brushes.Black,
Column2HoriztonalLocationSingle, VerticalPrintLocationSingle)
                    VerticalPrintLocationSingle += LineHeightSingle * 2
                End If
            End If
        Next IndexInteger
    End Sub
End Class
```

BIS 628 – Application Development

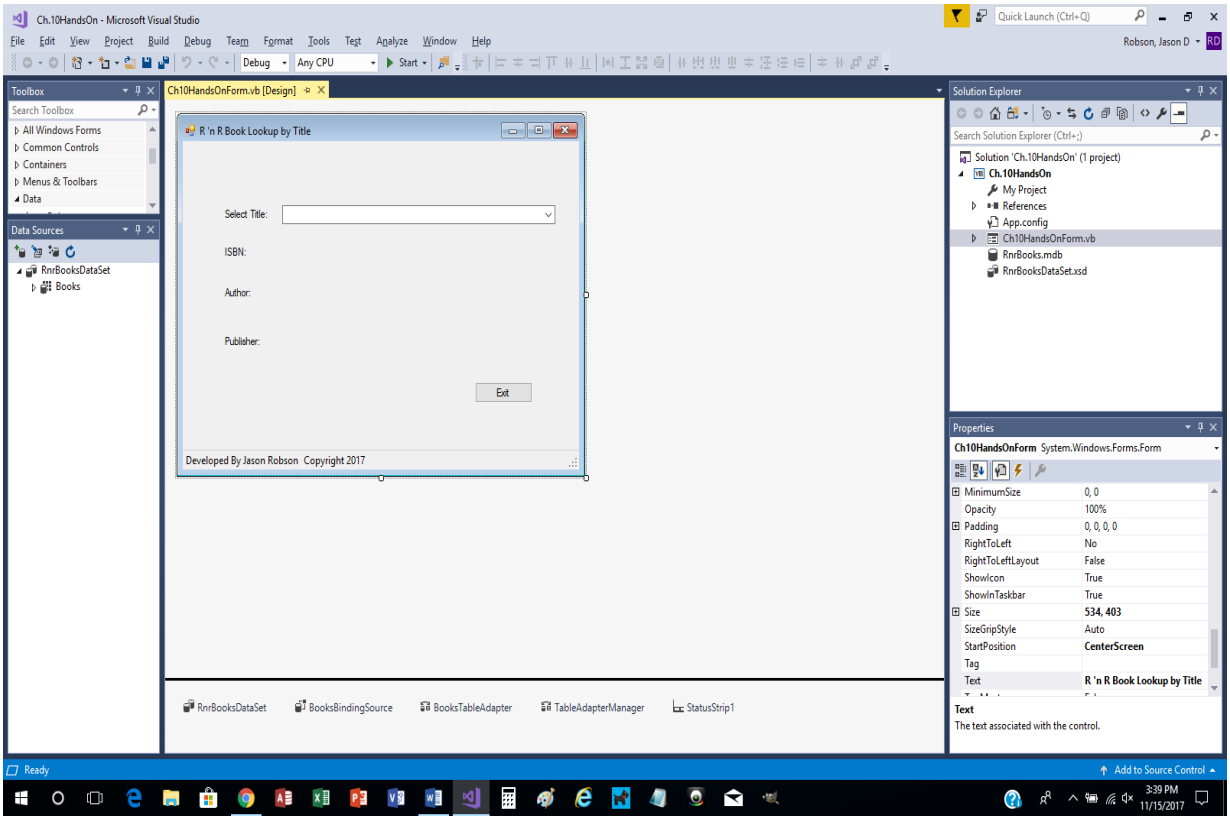
Assignment 10

- Chapter 10 Hands-On
- Example 10.1 (page 433)

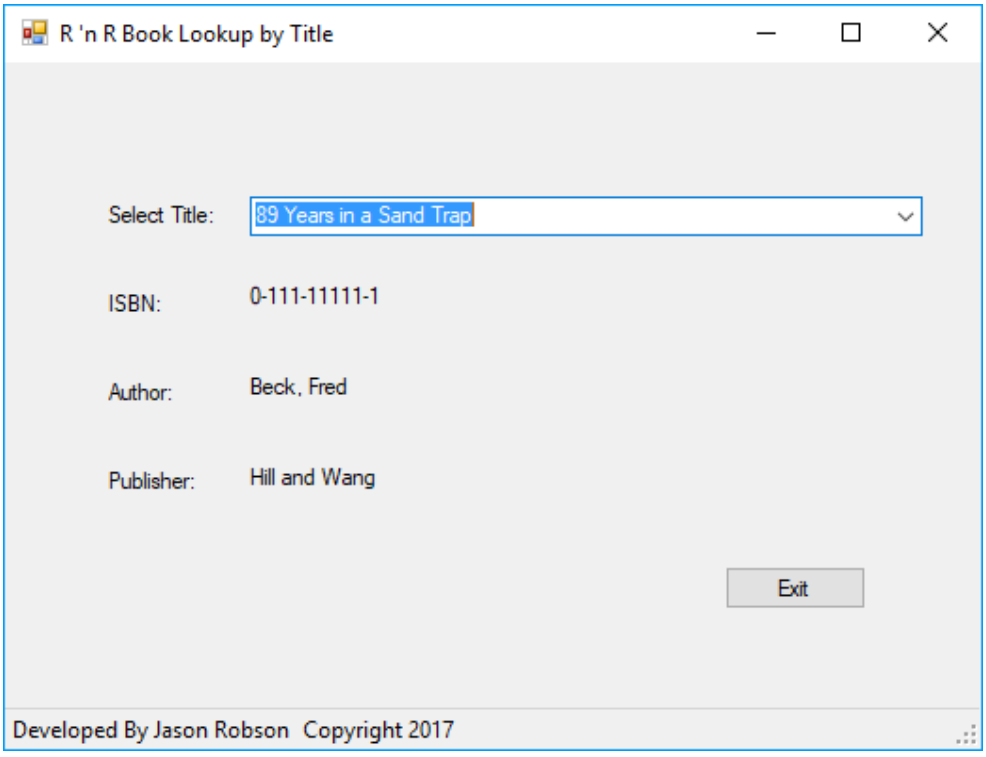
Due Date: 11/20/2017 11:59 pm

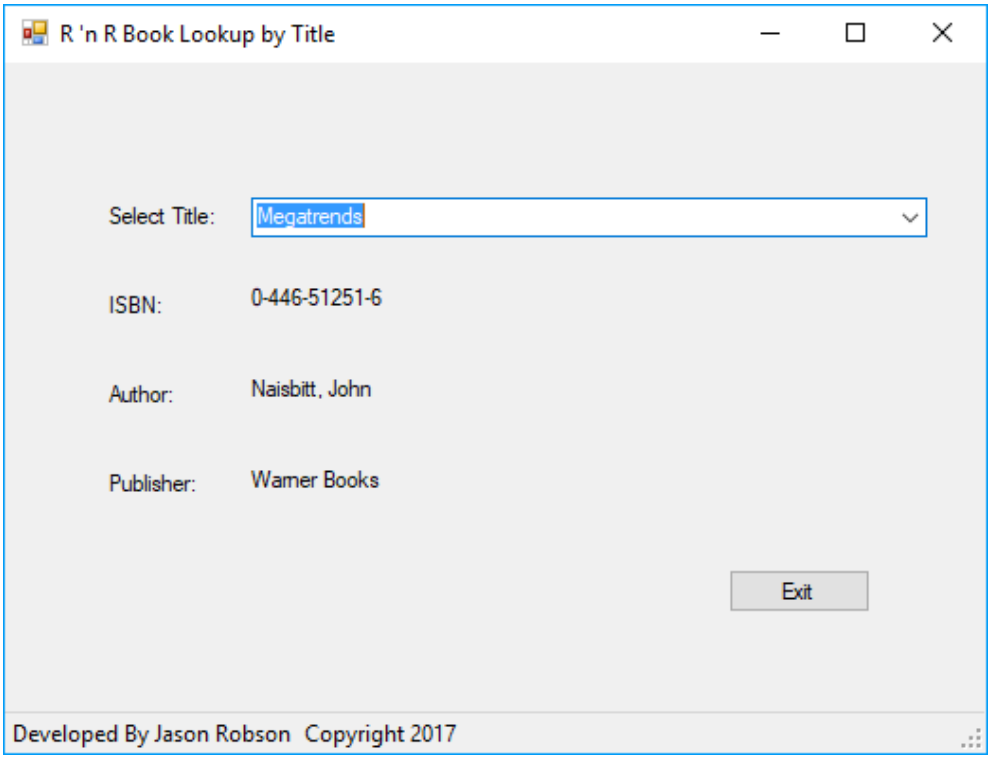
By: Jason Robson

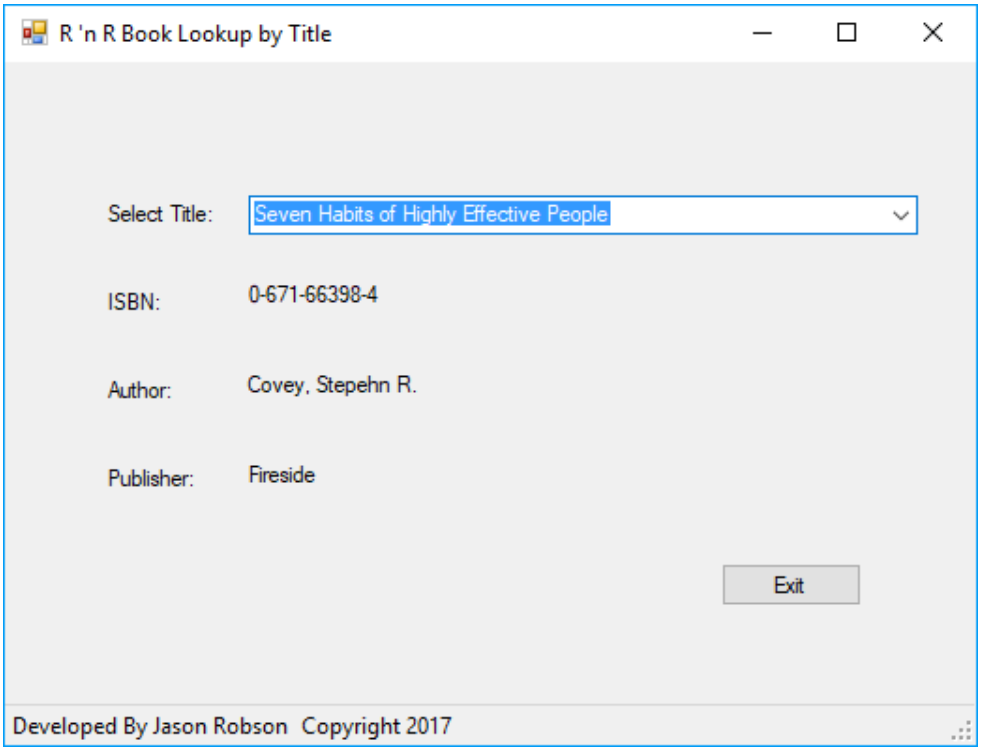
1. Design View (Book Lookup)



2. Looking Up Books By Title





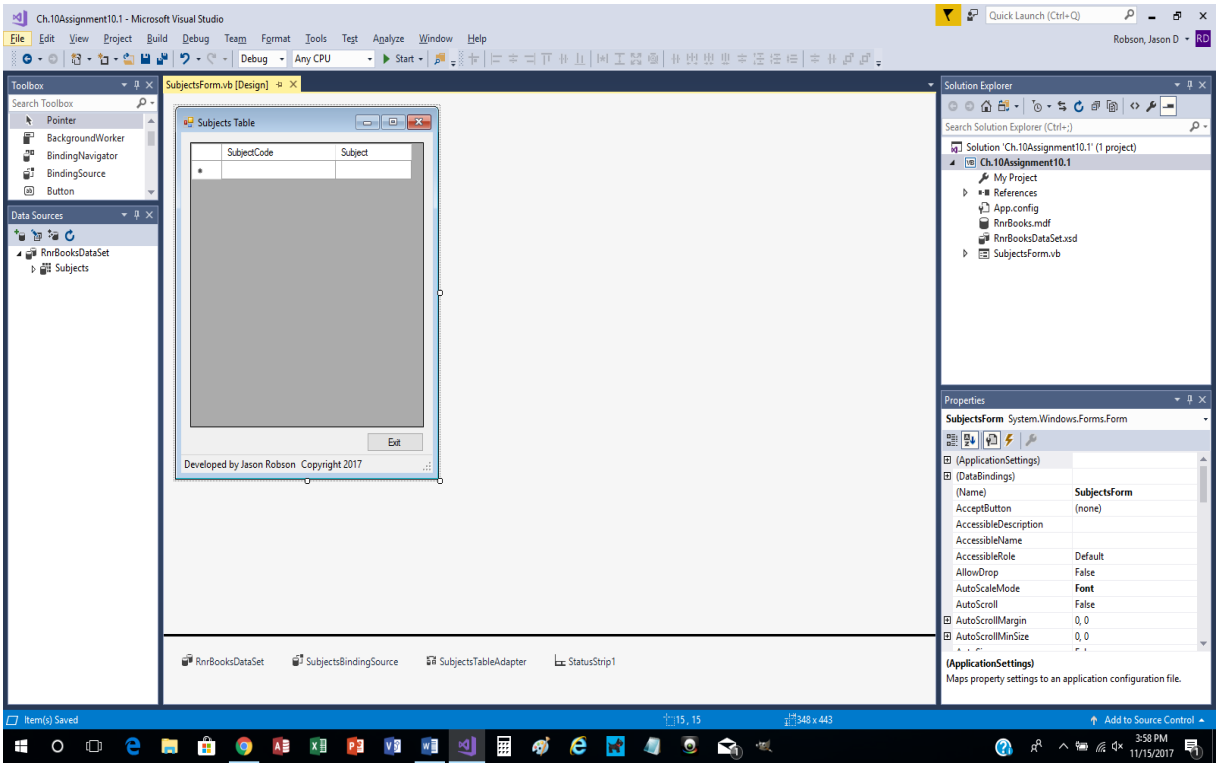


3. Program Code (Book lookup)

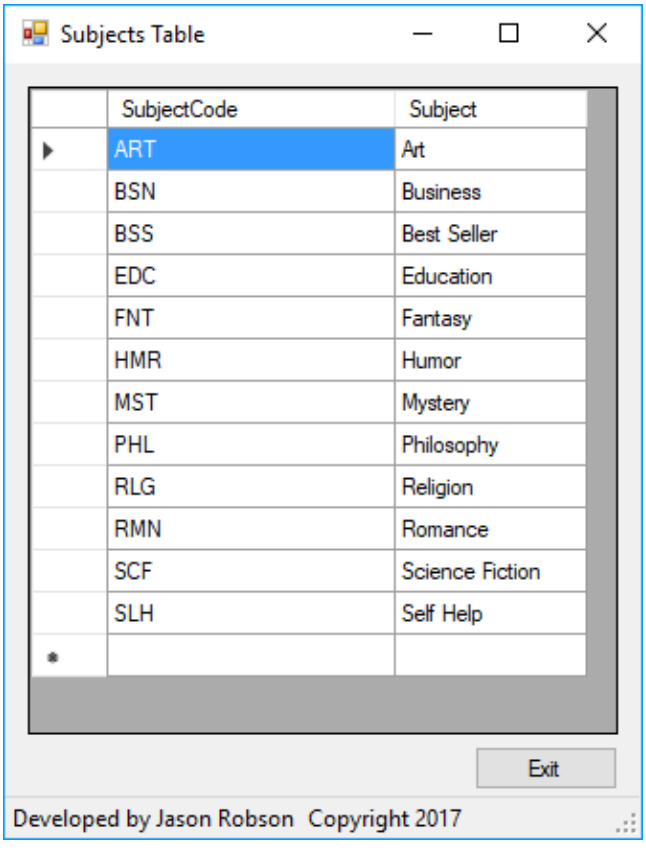
```
'Project     : Book Lookup Table
'Programmer  : Jason Robson
'Date        : November 15, 2017
'Description : Project looks up books by title and displays ISBN, author, and
publisher.

Public Class Ch10HandsOnForm
    Private Sub BooksBindingNavigatorSaveItem_Click(sender As Object, e As
EventArgs)
        Me.Validate()
        Me.BooksBindingSource.EndEdit()
        Me.TableAdapterManager.UpdateAll(Me.RnrBooksDataSet)

    End Sub

    Private Sub Ch10HandsOnForm_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        'TODO: This line of code loads data into the 'RnrBooksDataSet.Books'
table. You can move, or remove it, as needed.
        Me.BooksTableAdapter.Fill(Me.RnrBooksDataSet.Books)

    End Sub

    Private Sub ExitButton_Click(sender As Object, e As EventArgs) Handles
ExitButton.Click
```

```
    'Close the form
    Me.Close()
End Sub
   End Class
```

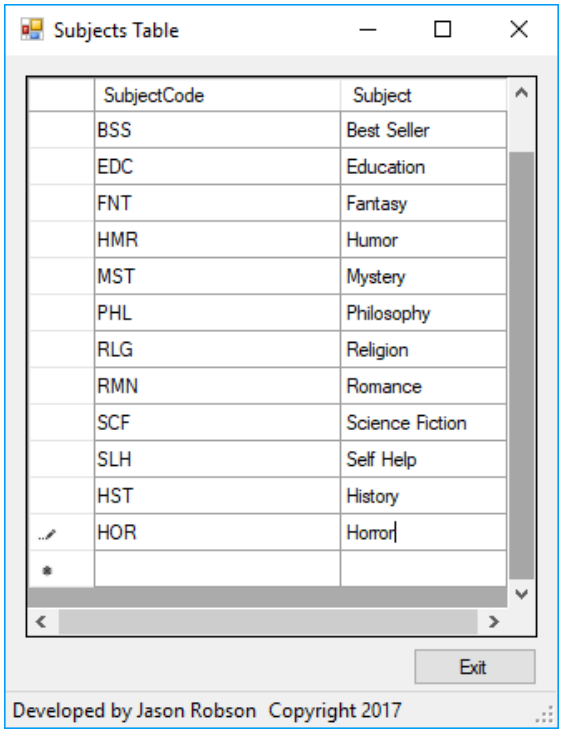
4. Design View (Subject Grid)



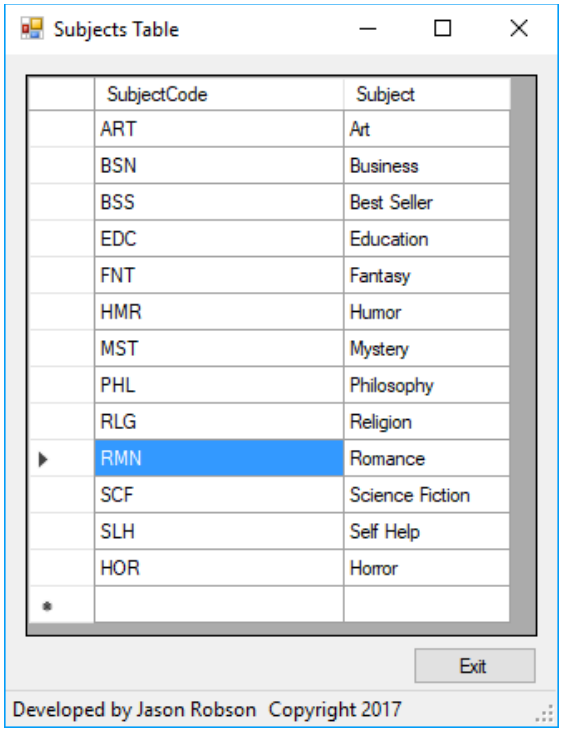
5. Program Running (Subject grid)



5.1 Adding to Grid



5.2 Deleting from Gird



6. Program Code (Subject Data Grid)

```
'Project    : Subjects Data Grid
'Programmer : Jason Robson
'Date       : November 12, 2017
'Description: Transfer subjects.mdf database file to table grid.

Public Class SubjectsForm
    Private Sub SubjectsForm_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        'TODO: This line of code loads data into the 'RnrBooksDataSet.Subjects'
table. You can move, or remove it, as needed.
        Me.SubjectsTableAdapter.Fill(Me.RnrBooksDataSet.Subjects)

    End Sub

    Private Sub ExitButton_Click(sender As Object, e As EventArgs) Handles
ExitButton.Click
        'Close the form
        Me.Close()
    End Sub
       End Class
```