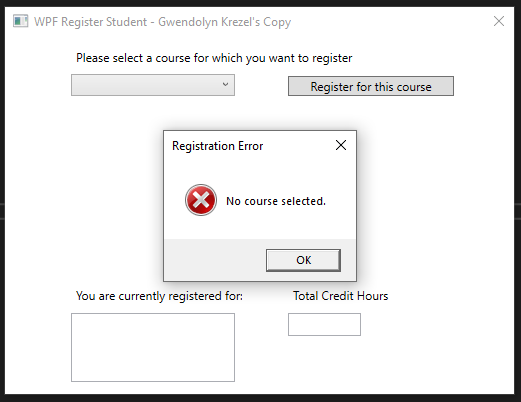
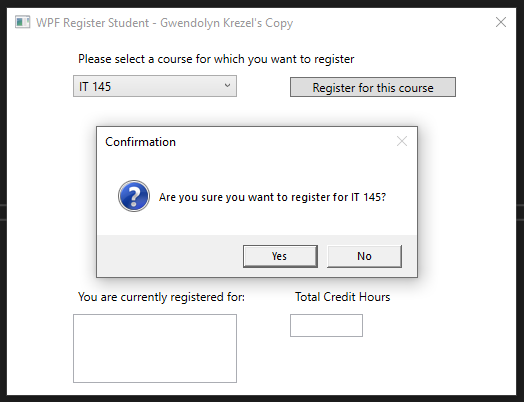


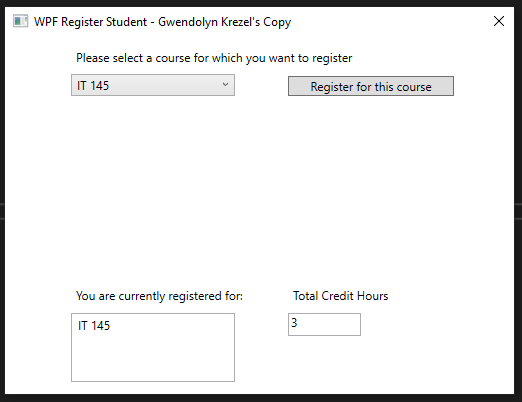
Upon opening the program, the selection box is blank. This will lead to an error, so there is a check for if the selection box is null.



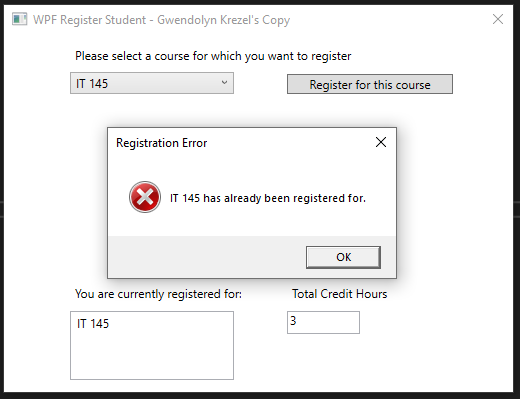
When the selection box is null, it gives the user a warning.



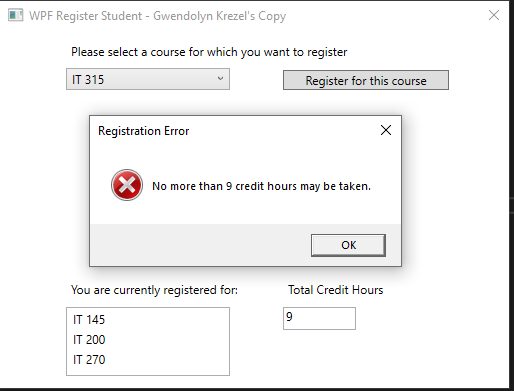
Attempting to register for a course shows a confirmation message, prompting the user for input.



Upon hitting “Yes”, the listBox is updated as well as the total credit hours.



Attempting to register for the same course twice shows the user an error.



Attempting to register for more than 9 credit hours also shows an error.

**Source Code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using System.Xml.Serialization;

namespace WPFRegisterStudent

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

Course choice;

int creditHours = 0;

public MainWindow()

{

InitializeComponent();

}

private void Window\_Loaded(object sender, RoutedEventArgs e)

{

Course course1 = new Course("IT 145");

Course course2 = new Course("IT 200");

Course course3 = new Course("IT 201");

Course course4 = new Course("IT 270");

Course course5 = new Course("IT 315");

Course course6 = new Course("IT 328");

Course course7 = new Course("IT 330");

this.comboBox.Items.Add(course1);

this.comboBox.Items.Add(course2);

this.comboBox.Items.Add(course3);

this.comboBox.Items.Add(course4);

this.comboBox.Items.Add(course5);

this.comboBox.Items.Add(course6);

this.comboBox.Items.Add(course7);

this.creditHourBox.Text = "";

}

private void button\_Click(object sender, RoutedEventArgs e)

{

choice = (Course)(this.comboBox.SelectedItem);

// TO DO - Create code to validate user selection (the choice object)

// and to display an error or a registation confirmation message accordinlgy

// Also update the total credit hours textbox if registration is confirmed for a selected course

// Check if no choice was selected

if (choice == null)

{

MessageBox.Show("No course selected.", "Registration Error", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

// Check if the choice has already been chosen

if (checkListBox(choice)) return;

// Check if the user already has 9 credit hours

if (creditHours >= 9)

{

MessageBox.Show("No more than 9 credit hours may be taken.", "Registration Error", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

// Confirm with user if the right course was selected

if (MessageBox.Show($"Are you sure you want to register for {choice.ToString()}?", "Confirmation", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

// Update UI

this.listBox.Items.Add(choice);

creditHours += 3;

this.creditHourBox.Text = creditHours.ToString();

}

}

// Checks if the listbox contains the desired course;

private bool checkListBox(Course course)

{

foreach (object listItem in this.listBox.Items)

{

if (listItem.ToString() == course.ToString())

{

MessageBox.Show($"{choice.ToString()} has already been registered for.", "Registration Error", MessageBoxButton.OK, MessageBoxImage.Error);

return true;

}

}

return false;

}

}

}