Tyler Baynes

4/9/2025

IT-230

Mod 7 WPFRegisterStudent

Source Code:

Course.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Controls;

namespace WPFRegisterStudent

{

class Course

{

private string name = "";

private bool isRegisteredAlready = false;

public Course(string name)

{

this.name = name;

}

public void setName(string name)

{

this.name = name;

}

public string getName()

{

return name;

}

public bool IsRegisteredAlready()

{

return isRegisteredAlready;

}

public void SetToRegistered()

{

isRegisteredAlready = true;

}

public override string ToString()

{

return getName();

}

}

}

MainWindow.xaml

<Window x:Class="WPFRegisterStudent.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:WPFRegisterStudent"

mc:Ignorable="d"

Title="WPF Register Student - Baynes, Tyler" Height="395.902" Width="525" Loaded="Window\_Loaded">

<Grid>

<ComboBox x:Name="comboBox" HorizontalAlignment="Left" Margin="66,37,0,0" VerticalAlignment="Top" Width="164" IsDropDownOpen="True"/>

<Button x:Name="button" Content="Register for this course" HorizontalAlignment="Left" Margin="283,39,0,0" VerticalAlignment="Top" Width="166" Click="button\_Click"/>

<ListBox x:Name="listBox" HorizontalAlignment="Left" Height="69" Margin="66,276,0,0" VerticalAlignment="Top" Width="164"/>

<TextBox x:Name="textBox" HorizontalAlignment="Left" Height="23" Margin="283,276,0,0" TextWrapping="Wrap" Text="0" VerticalAlignment="Top" Width="73" IsReadOnly="True"/>

<Label x:Name="label" Content="Please select a course for which you want to register" HorizontalAlignment="Left" Margin="66,7,0,0" VerticalAlignment="Top" Width="383"/>

<Label x:Name="label1" Content="You are currently registered for:" HorizontalAlignment="Left" Margin="66,245,0,0" VerticalAlignment="Top" Width="176"/>

<Label x:Name="label2" Content="Total Credit Hours" HorizontalAlignment="Left" Margin="283,245,0,0" VerticalAlignment="Top" Width="106"/>

<Label x:Name="label3" Content="" HorizontalAlignment="Left" Margin="66,213,0,0" VerticalAlignment="Top" Width="383" Foreground="#FFEE0E0E"/>

</Grid>

</Window>

MainWindow.xaml.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Linq.Expressions;

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;

namespace WPFRegisterStudent

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

Course choice;

private int totalCreditHours;//Initializes int creditHours

//Course handles already registered exceptions within the isAlreadyRegistered bool

public MainWindow()

{

InitializeComponent();

}

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

{

//Assigns names to coursex values

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");

//Adds items to comboBox

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);

//Credit hours text box

this.textBox.Text = $"{totalCreditHours}";

}

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

{

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

if (choice == null )

{

label3.Content = "Please make a selection.";

}

if (choice.IsRegisteredAlready())

{

//If choice.IsRegisteredAlready returns true:

label3.Content = "Course is already registered.";

//If already registered, nothing else is needed.

}

else

{

if (totalCreditHours + 3 <= 9)

{

choice.SetToRegistered();

label3.Content = choice.getName() + " has been registered.";

totalCreditHours += 3;

this.listBox.Items.Add(choice);

this.textBox.Text = $"{totalCreditHours}";

}

else if (totalCreditHours + 3 >= 10)

{

label3.Content = "This exceeds the maximum number of credits(9).";

}

}

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

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

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

}

}

}

How the program appears upon opening. A screenshot of a computer

AI-generated content may be incorrect.

The first course has been added with the credit hours updating. A screenshot of a computer

AI-generated content may be incorrect.

Trying to add the first course again. This is stopped by using the bool logic of isAlreadyRegistered. A screenshot of a computer

AI-generated content may be incorrect.

Adding the second course and updating the credit hours. A computer screen with a message

AI-generated content may be incorrect.

Adding the third course and updating the credit hours.. A screenshot of a computer

AI-generated content may be incorrect.

Adding a fourth course but is stopped because of the check in the button\_Click if statement using totalCreditHours + >= 10. A screenshot of a computer

AI-generated content may be incorrect.

A shot of the dropdown activating. A screenshot of a computer

AI-generated content may be incorrect.