**IT 230 Coding Activity Submission Template**

**Name: Brantly McKendree**

**Date: Sept. 2, 2020**

**Class:** IT 230

**Module: Module 7 Final**



|  |  |
| --- | --- |
| **1.** | Insert a copy of your of the ZIP file of all of your Visual Studio project files here so that it can be loaded and run in another Visual Studio: |
| Insert here a copy of your \*.cs source code text you used here (copy and paste source code here, do **not** simply insert \*.cs files):  **MainWindow.xaml.cs**  using System;  using System.CodeDom;  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;  public MainWindow()  {  InitializeComponent();  }  public 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.textBox.Text = "";    }  private void button\_Click(object sender, RoutedEventArgs e)  {  choice = (Course)(this.comboBox.SelectedItem); // Creates a selected item to be tested and validated.    if (choice.IsRegisteredAlready() == true) //Checks if the class has been registered for.  {  this.label3.Content = ($"You have already registered for {choice}."); // Displays message in error box.  }    else if (Course.creditHours >= 9) // Credit hour validation  {  this.label3.Content = ("You can not register for more than 9 credit hours."); // Displays message in error box.  }  else  {  choice.SetToRegistered(); // Sets the course to IsRegisteredAlready = true and adds 3 hours to credit hours.  this.listBox.Items.Add(choice); // Adds choice to list box.  this.label3.Content = ($"Registration confirmed for course {choice}."); // Displays message confirming registration.  this.textBox.Text = Convert.ToString(Course.creditHours); // Converts the creditHours to a string to be input into textbox.  }    }  }  }    **Course cs**  using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace WPFRegisterStudent  {  class Course  {  public string name = "";  public bool isRegisteredAlready = false;  public static int creditHours;  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; // Makes the course boolean IsRegistered to true for validation.  creditHours += 3; // Adds credit hours if the user selection passes validation.  }  public override string ToString()  {  return getName();  }  }  } |
| **2.** | Insert a screenshot here of the output that resulted from running your program, showing your last name as the first printed text to the screen:    Graphical user interface, text, application  Description automatically generated  Graphical user interface, text, application, email  Description automatically generated |
| **3.** | Explain the design of your program, the steps you took to complete it, and how you coded it:    The first step I took was creating a selected item within the private void button click in MainWindow.xaml.cs. choice = (Course)(this.comboBox.SelectedItem); // Creates a selected item to be tested and validated. This code allowed me to test the inputs that are entered when choosing a course to register for. The next step was to create a branch of logic to test the registered courses.  if (choice.IsRegisteredAlready() == true) //Checks if the class has been registered for.  {  this.label3.Content = ($"You have already registered for {choice}."); // Displays message in error box.  }  This branch allows the course being registered for to be tested and if it has already been registered it will not be able to be registered again. This also displays the error message that lets the user know they have registered for the class already. The next part of the branch is as follows:  else if (Course.creditHours >= 9) // Credit hour validation  {  this.label3.Content = ("You can not register for more than 9 credit hours."); // Displays message in error box.  }  This part of the branch tests the credit hours and if they have already registered to 9 credit hours, they will not be able to register for more. This allows displays the error message for too many credit hours.  The final part of the branch allows the course to register and continue through the process of registration for the student. It is as follows.  else  {  choice.SetToRegistered(); // Sets the course to IsRegisteredAlready = true and adds 3 hours to credit hours.  this.listBox.Items.Add(choice); // Adds choice to list box.  this.label3.Content = ($"Registration confirmed for course {choice}."); // Displays message confirming registration.  this.textBox.Text = Convert.ToString(Course.creditHours); // Converts the creditHours to a string to be input into textbox.  }  This will display positive registration and it adds credit hours and sets the course to registered. This is done through the method created in the Course class. |
| **4.** | Reflect on this experience and the lessons you learned from it:    The main lessons I learned from this assignment were how to use methods correctly and Boolean operations within the program. I also learned a lot about how to write to an application and how to make it work. The branches have to be written in a certain way and if they are place incorrectly and written incorrectly then there will be bugs and it causes problems when the program executes. |