

IT 230 Coding Activity Submission Template

**Name:** Ryan Hutton

**Date:** 12/13/2019

**Class:** IT 230 – Software Development with C#.NET

**Module:** Final Project Part II

|  |  |
| --- | --- |
| **1.** | Insert a copy of the ZIP file of all 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):  namespace WPFRegisterStudent  {  /// <summary>  /// Interaction logic for MainWindow.xaml  /// </summary>  public partial class MainWindow : Window  {  Course choice;  private int creditHours = 0; // Added this variable to track registered credit hours - RH  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);  }  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  COMPLETED 12/13/2019 - RH  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*RH CODE ADDED FOR PROJECT IN THIS FUNCTION BELOW\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  if (this.listBox.Items.Contains(choice) && choice.IsRegisteredAlready())  {  this.textBlock1.Text = "You have already registered for this " + choice.ToString() + " course"; // Add string to custom text block and print it to window.  }  else if (creditHours < 9) // Total credit hours cannot exceed 9 credit hours  {  this.listBox.Items.Add(choice); // Add each select to the output box of course.  choice.SetToRegistered();  creditHours += 3; // Each course has 3 credit hours each and is updated.  this.textBox.Text = Convert.ToString(creditHours); // TextBox is the credit hour text box displayed.  this.textBlock1.Text = "Registration confirmed for course " + choice.ToString(); // Print string when each eligible course is added.  }  else  this.textBlock1.Text = "You cannot register for more than 9 credit hours.";  }  private void comboBox\_SelectionChanged(object sender, SelectionChangedEventArgs e)  {  //comboBox is the method for the dropbox list of courses to choose from.  //It does not need to be defined below to execute this project.  }    private void listBox\_SelectionChanged(object sender, SelectionChangedEventArgs e)  {  //listBox is the method where the courses will output to.  //It does not need to be defined below to execute this project.  }  private void textBox\_TextChanged(object sender, TextChangedEventArgs e)  {  //This textBox method is the creditHours text box.  //It does not need to be defined below to execute this project.  }  }  } |
| **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: |
| **3.** | Explain the design of your program, the steps you took to complete it, and how you coded it:  This program utilizes several classes and objects to create a Visual Studio WPF application. In the WPF XAML window, three main objects are used to manipulate the data. The first object is comboBox which contains the list of courses to register for. The next object is listBox, which is where the courses will output to when the user selects each course. The textbox object updates the creditHours. I added a fourth object to the WPF window, named textBlock1, which is used to update the confirmation or error messages.  The requirements of the assignment were to create code to validate the user selection and to display an error or confirmation message accordingly. Also, we were required to update the total credit hours for confirmed registration. In order to code this assignment, I mistakenly attempted to dive in and start coding because I underestimated how much I had to think about the code. After humbling myself and seeking help, I decided to go back to the drawing board and write pseudocode. I also experimented with the XAML file to learn about how I can customize the application window. Immediately into me writing pseudocode, everything suddenly clicked, and I set up the conditional statements to drive the code with the appropriate user selection. The conditional checks if the course chosen is already registered by invoking the Contains method within the Items class, or if there are more than 3 courses registered. If everything checks out, then the course is added with a confirmation message and the total credit hours are updated with the creditHours variable, which is of type int. This is done by 1) adding the choice variable and invoking choice.SetToRegistered(), 2) updating the variable creditHours, 3) converting the creditHours variable to a string and assigning it the textBox, and 4) concatenating choice.ToString() to the confirmation message and assigning it to the textBlock1 I created. |
| **4.** | Reflect on this experience and the lessons you learned from it:  This was a good reminder on the importance of writing pseudocode to solve a problem. Another thing I learned is that it is sometimes easier to step away from the computer and solve the problem on paper. The assignment also strengthened my skills with creating WPF applications and manipulating the window to improve the user’s experience withing a graphical user interface. These skills will be very useful in coding for user experience in a software application. |