# IT 230 Coding Activity Submission Template

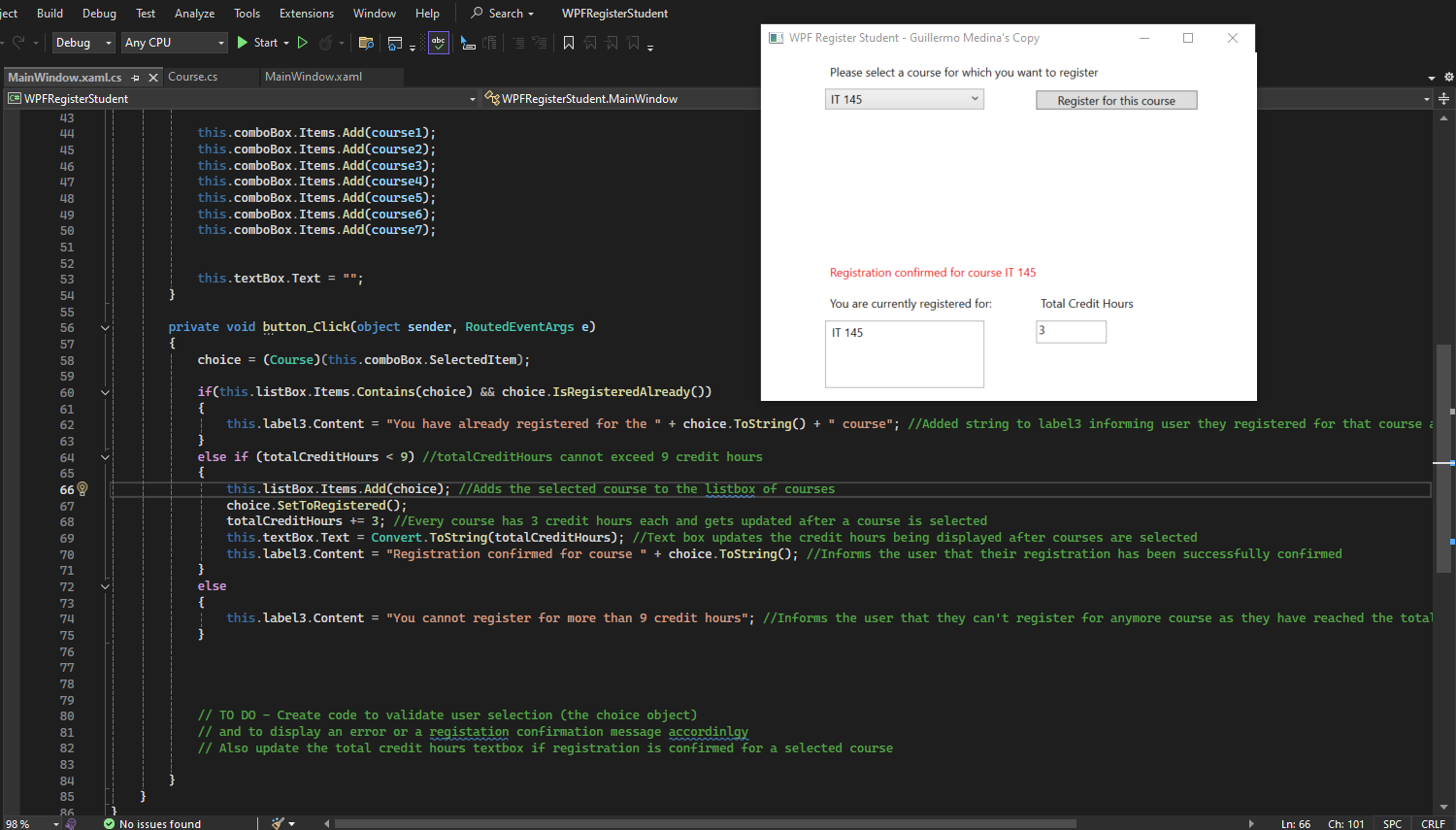
Submit your work on the coding activities for Modules One, Two, Three, Four, and Six in this document. In addition to this document, you should submit a ZIP file containing all your Visual Studio project files and source code that can be run in Visual Studio on a different computer.

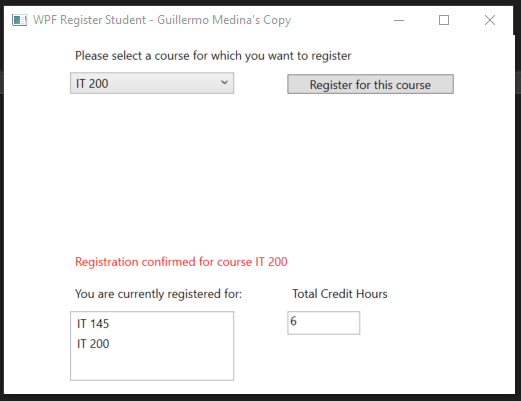
For each coding activity, complete the following steps:

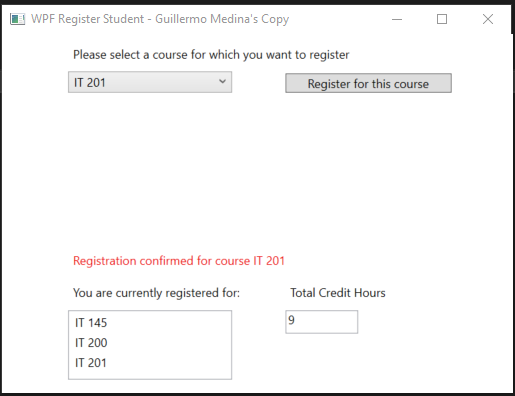
* Download and rename this document to meet the file naming conventions requested in the assignment instructions.
* Fill in the required information below by replacing the bracketed text with the relevant information.
* Submit this document and your ZIP file for grading and feedback. Your ZIP file should follow the same naming conventions.

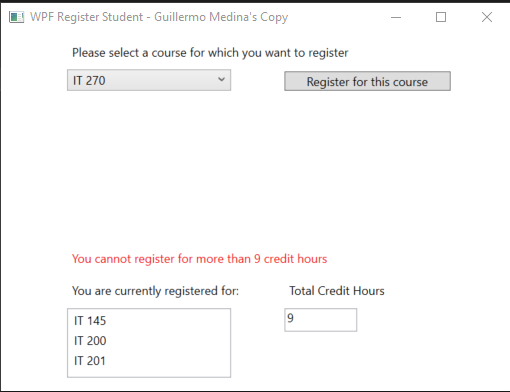
Document your work in the coding activity by completing each of the following items:

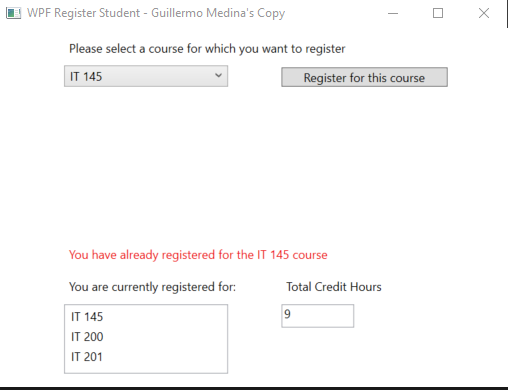
1. Provide a screenshot of the output that resulted from running your program successfully in Visual Studio. See the coding assignment instructions for an example of what should be included in the screenshot. Your screenshot must include the following elements:
   1. Your last name as the first printed text on the screen
   2. Verification that the program is fully functioning and data results are accurate for the given problem











1. Copy and paste the source code text you wrote for this assignment from the \*.cs file into the space below. Only providing the \*.cs files or a screenshot does not meet the requirements for this part of the assignment. Code should be logically organized. It should also follow proper syntax and conventions noted in the Coding Activity Guidelines and Rubric.

private int totalCreditHours = 0; // Variable to track registered credit hours

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

{

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

if(this.listBox.Items.Contains(choice) && choice.IsRegisteredAlready())

{

this.label3.Content = "You have already registered for the " + choice.ToString() + " course"; //Added string to label3 informing user they registered for that course already

}

else if (totalCreditHours < 9) //totalCreditHours cannot exceed 9 credit hours

{

this.listBox.Items.Add(choice); //Adds the selected course to the listbox of courses

choice.SetToRegistered();

totalCreditHours += 3; //Every course has 3 credit hours each and gets updated after a course is selected

this.textBox.Text = Convert.ToString(totalCreditHours); //Text box updates the credit hours being displayed after courses are selected

this.label3.Content = "Registration confirmed for course " + choice.ToString(); //Informs the user that their registration has been successfully confirmed

}

else

{

this.label3.Content = "You cannot register for more than 9 credit hours"; //Informs the user that they can't register for anymore course as they have reached the total credit limit

}

1. Show that you understand the task by explaining the design of your program in the space below. Include the process and steps you took to write your code. Explain how you arrived at the solution to the problem and completed the activity.

Using context clues from the guidelines and rubric, I was able to deduce a couple of things.

Each course carries three credit hours, and it could not have more than nine credit hours, let me know that I needed a variable that would increase by three each time a selection was made. This led me to the conclusion that if statements were also needed in order to do the appropriate checks. I first created a private int value that the xaml.cs file would be able to use called totalCreditHours which will be used a bit later. Following that, I began coding my if statements. I kept cycling back and forth from the xaml.cs file to the regular xaml file in order to make sure the boxes and labels I was editing were the correct ones. For the first if statement I had to make it so that it would check if the choice was in the listbox and was already registered, label3 would inform the user that they are already registered for that course and the credit hours wouldn’t increase. The next else if statement was that the totalCreditHours was less than 9, it would add the course to the listbox, set the course as registered, increase the totalCreditHours by 3, convert totalCreditHours ToString in the textbox, and inform the user that their registration has been confirmed. Last but not least, the else statement informs the user in label3 that they cannot register for more than 9 credit hours only letting them select 3 courses.

1. Reflect on your learning experience and what you learned from completing the activity.

Coding this section gave me a hard time because even though I knew what needed to be applied, I struggled to figure out how to piece together the textbox, labels, items, etc. Once I was able to match up what was on the xaml file to the xaml.cs file, it all started to come together in a way that was intuitive and made sense. Visual Studio is also very good at giving you a direction on where to go, like for example giving me the idea of .Items.Contains for the first if statement. Overall, I think this was a nice experience on how to actually code a longer snippet and really being on our own to try, fail, and succeed at the end of the day.