# 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

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

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.

**Course.cs:**

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.cs:**

using System;

using System.Collections.Generic;

using System.Windows;

namespace WPFRegisterStudent

{

public partial class MainWindow : Window

{

private List<Course> courses;

private int totalCredits = 0;

public MainWindow()

{

InitializeComponent();

}

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

{

// Initialize the courses

courses = new List<Course>

{

new Course("IT 145"),

new Course("IT 200"),

new Course("IT 201"),

new Course("IT 270"),

new Course("IT 315"),

new Course("IT 328"),

new Course("IT 330")

};

// Populate the ComboBox with courses

foreach (var course in courses)

{

comboBox.Items.Add(course.getName());

}

textBox.Text = "";

}

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

{

Course selectedCourse = GetSelectedCourse();

if (selectedCourse != null)

{

if (RegisterCourse(selectedCourse))

{

UpdateUI();

}

}

else

{

MessageBox.Show("Please select a course.");

}

}

private Course GetSelectedCourse()

{

string selectedCourseName = comboBox.SelectedItem as string;

foreach (var course in courses)

{

if (course.getName() == selectedCourseName)

{

return course;

}

}

return null;

}

private bool RegisterCourse(Course selectedCourse)

{

if (selectedCourse.IsRegisteredAlready())

{

MessageBox.Show("This course is already registered.");

return false;

}

if (totalCredits + 3 > 9)

{

MessageBox.Show("You cannot register for more than 9 credit hours.");

return false;

}

selectedCourse.SetToRegistered();

totalCredits += 3;

MessageBox.Show($"Successfully registered for {selectedCourse.getName()}.");

return true;

}

private void UpdateUI()

{

textBox.Text = $"Total Credits: {totalCredits}";

listBox.Items.Clear();

foreach (var course in courses)

{

if (course.IsRegisteredAlready())

{

listBox.Items.Add(course.getName());

}

}

}

}

}

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.

**To complete this assignment, I made sure that the course.cs section could properly read the situations relating to selecting courses. In MainWindow.xaml.cs, I also made sure that the user’s choice was validated, and that the proper information would display after selecting your course. It was very important to make sure that each course worked and that they would display properly.**

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

**I learned a lot about referencing multiple different sections of code. I had to do a bit of back and forth between course.cs, MainWindow.xaml, and MainWindow.xaml.cs. It was important that the information was consistent and that I used the same exact terms for each section. If one thing was wrong in one, I would need to go back and make sure that it was consistent. It tripped me up for about 10 minutes, but luckily, I was able to find it easily. It was a nice experience to be able to manage more than one window of code at once.**