# IT 230 Coding Activity Submission Template

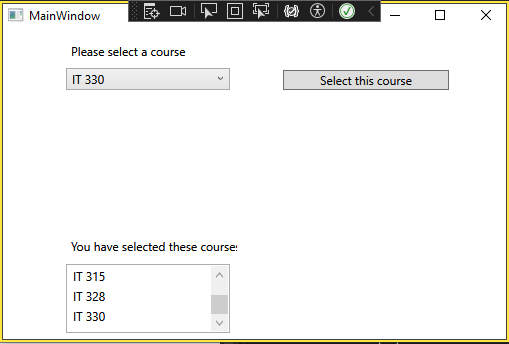
Submit your work on the coding activities for Modules One, Two, Three, Four, and Six in this document. In addition to this document, you should submit a ZIP file containing all your Visual Studio project files and source code that can be run in Visual Studio on a different computer.

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.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace CreateClassesObjs

{

internal class Course

{

private string name; // This references the setName within MainWindow.xaml.cs, lines 41-47.

public void setName(string userChoice) // This method sets the user selected choices or userChoice.

{ name = userChoice; }

public string getName() // This method retreives the selected name(s)/course(s) that was(were) set in userChoice.

{ return name; }

public override string ToString() // This method should return the name field.

{ return name; }

}

}

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.

This allows the program to select multiple courses from a drop-down menu. The menu list is accessed with the private setName and the userChoice is set to the new Course which is returned to a string that is seen in the bottom text box.

The first requirement was the course has a private string field to hold the name of the course. This became “private string name;”. The set name and retrieve name fields following this did not need to be private. The final requirement was the override to string method that allowed the userChoice to be returned and displayed in the WPF window. Visual Studio also prompted parts of the code in the same order as the requirements within the assignment.

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

I believe that I have learned a little more about using private vs public access modifiers. It is often simple enough in theory but can be harder to apply in practice. While this project was laid out step by step, that will not always be the case in the future. With private only being accessible within the same class it was needed to access the course name in MainWindow.xaml.cs. The private modifer allows it’s data to be hidden from the user interface. The following public void is accessible for all classes and returns the userChoice object to setName that leads to the visibility in the WPF window through the rest of the code.