# IT 230 Coding Activity Submission Template

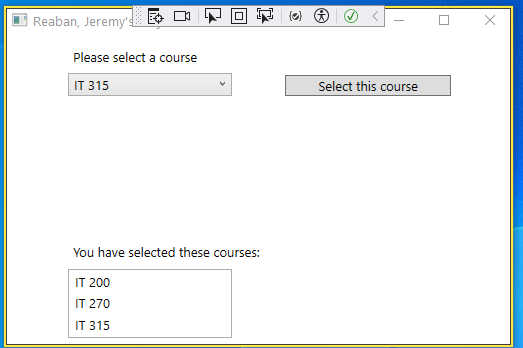
Submit your work on the coding activities for Modules One, Two, Three, Four, and Six in this document. In addition to this document, you should submit a ZIP file containing all your Visual Studio project files and source code that can be run in Visual Studio on a different computer.

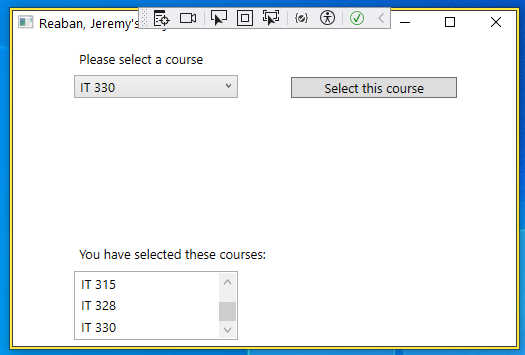
For each coding activity, complete the following steps:

* 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

Showing 3 courses selected and added to the list box



Showing 2 more items added to the list box and the scroll bar scrolled down to show it.

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 courseName;

public void setName(string courseName)

{

this.courseName = courseName;

}

public override string ToString()

{

return courseName;

}

}

}

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.

First, I used the IDE to create a new class in a file named Course.cs

Next, I created the variable courseName for the Course class and made it private, letting it only be used, and then created a method named setName in the class to set the courseName variable.

I did this because the main program simply creates several new instances of the Class object, passing no parameters, and then calls the .setName method. I personally would have made a constructor with a parameter for the courseName and thus each course object could be created with a courseName instead of having to set it. And I would have used the default getter and setter that C# supports instead of writing a method.

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

I am fairly familiar with Java (and C++) so it was interesting to see how C# and Visual Studio handles adding new classes. Although I did not use it in the final form, I saw how you can set the getters and setters by adding { get; set; } after the variable declaration statements.