# IT 230 Coding Activity Submission Template

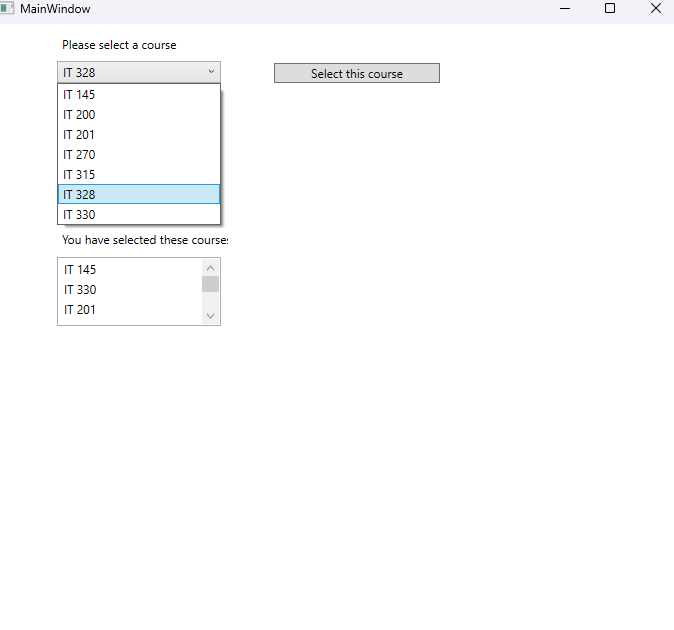
Submit your work on the coding activities for Modules One, Two, Three, Four, and Six in this document. In addition to this document, you should submit a ZIP file containing all your Visual Studio project files and source code that can be run in Visual Studio on a different computer.

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

{

class Course

{

private string Name = "null";

public void setName(string courseName) // Setter

{

this.Name = courseName;

}

public string getName // Getter

{

get

{

return this.Name;

}

}

public override string ToString() // ToString() Override

{

return this.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 was a rather simple class to make, but very important practice. Each course object needs a name, so I first initialized a variable ‘Name’ to private so that it could be set or gotten by methods. Then, I began work on the methods themselves. I needed to first create a setter, which was rather easy after practicing in Java. I simply assigned the instanced variable to the input name. Then, I needed a getter (also practiced in java) that simply returns the instanced name. I’ll need to do more reading on get{} within properties, but it does compile and work so far. Then, I simply made the ToString() override do the exact same thing as the getter and moved on.

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

Exploring the small differences in class creation between Java and C# has been quite fun so far, and, just as well, this class has been super helpful in expanding my knowledge of classes as a whole. The discussion this week really helped me understand how static variables work as compared to instanced variables, and this assignment was really good practice with writing classes and file organization.