# 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

AI-generated content may be incorrect.

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.Configuration;

namespace CreateClassesObjs

{

internal class Course

{

private string courseName;

// constructor method

public Course()

{

courseName = "";

}

// sets course name of object to the inputted name

public void SetName(string incCourseName)

{

courseName = incCourseName;

}

// gets course name

public string GetName()

{

return courseName;

}

// returns name as string (since it's already string we don't cast or parse anything)

public override string ToString()

{

return 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.

After we make our Course class, the first thing we can do is make a private string variable. This is easy, and we can declare it a global variable.

Next, we create a basic constructor method for the class. We don’t want any parameters in this method.

Then, we create a method called SetName() to assign courseName, our private string variable, to the incoming string variable that is the courses we are adding to the dropdown menu.

After that, we create a method called GetName(), which just grabs the courseName of that course object.

Lastly, we create a method to grab the courseName of the object and convert it to a string with the ToString() method, which will return the courseName as a string variable.

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

I learned how to make C# object classes in relation to WPF applications. This class is the first time I’ve ever messed with WPF applications, so having to go back and forth between the MainWindow class for the UI and the Course class to see what methods I need to create and where I was at in the process was a big learning experience for me.