# IT 230 Coding Activity Submission Template

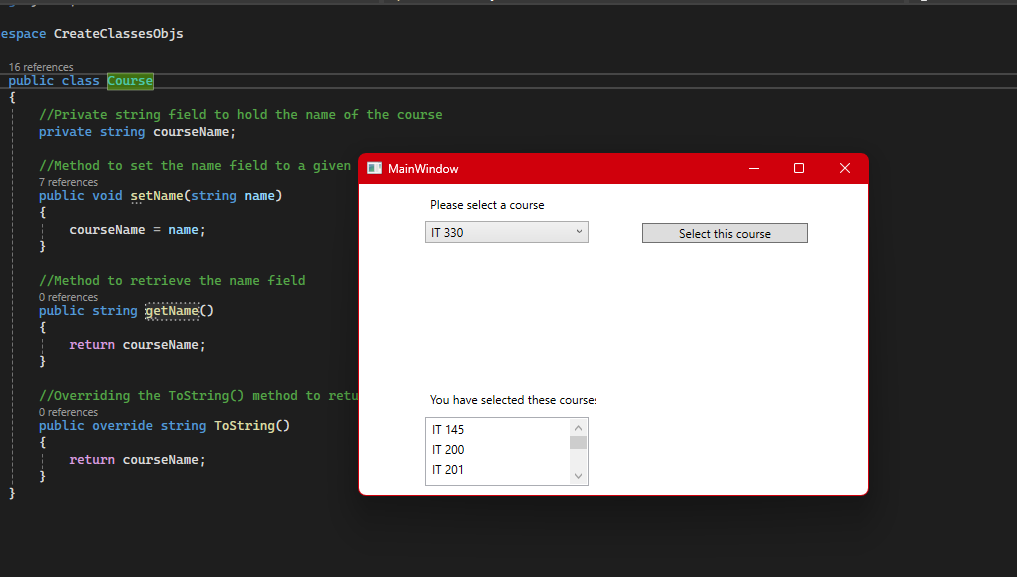
Submit your work on the coding activities for Modules One, Two, Three, Four, and Six in this document. In addition to this document, you should submit a ZIP file containing all your Visual Studio project files and source code that can be run in Visual Studio on a different computer.

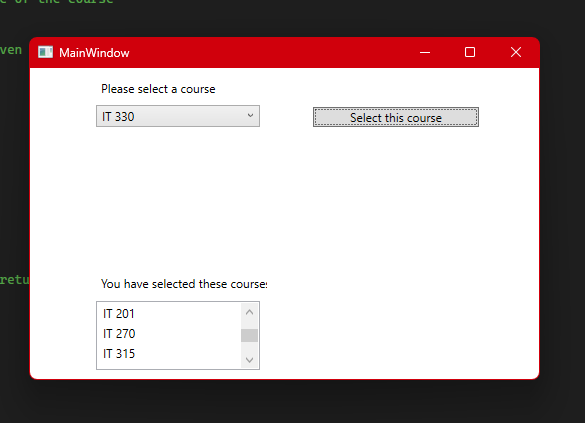
For each coding activity, complete the following steps:

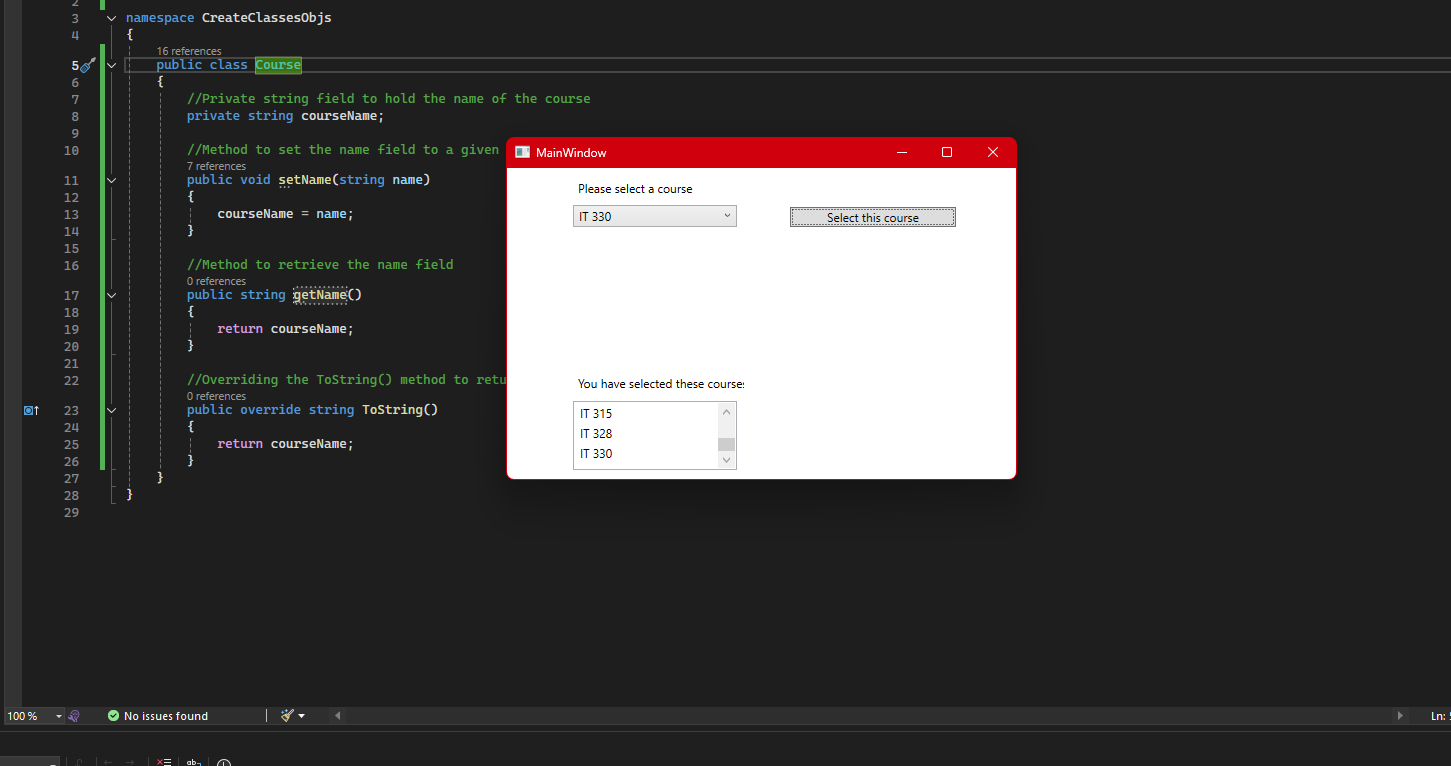
* Download and rename this document to meet the file naming conventions requested in the assignment instructions.
* Fill in the required information below by replacing the bracketed text with the relevant information.
* Submit this document and your ZIP file for grading and feedback. Your ZIP file should 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.
2. using System;
3. namespace CreateClassesObjs
4. {
5. public class Course
6. {
7. //Private string field to hold the name of the course
8. private string courseName;
9. //Method to set the name field to a given string value
10. public void setName(string name)
11. {
12. courseName = name;
13. }
14. //Method to retrieve the name field
15. public string getName()
16. {
17. return courseName;
18. }
19. //Overriding the ToString() method to return the name field
20. public override string ToString()
21. {
22. return courseName;
23. }
24. }
25. }
26. 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 file is what the program references (specifically in the MainWindow.xaml.cs file) in order to assign the name of the course to each course option seen in the menu the user interacts with. This class creates a private string courseName that is used to store the string that is passed in to the setName method and is subsequently retrieved by the getName method. This class also overrides the ToString method to ensure that it correctly displays the course name passed through into the setName method for each respective course option created in MainWindow.xaml.cs.
27. Reflect on your learning experience and what you learned from completing the activity.

It was really cool to be able to be given most parts of a program and then have to finish it out given the context from the rest of the files. I also really enjoyed having more of an official interface in play rather than just a program that ran in the console. I am interested in seeing what the next part of the project will entail.