# 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;

namespace DebugFixMethods

{

class Program

{

static void Main(string[] args)

{

(new Program()).Run();

}

void Run()

{

int choice = 0;

WritePrompt();

choice = ReadChoice();

WriteChoice(choice);

}

void WritePrompt()

{

Console.WriteLine("Nikki Malmanger's Copy");

Console.WriteLine("Please select a course for which you want to register by typing the number inside []");

Console.WriteLine("[1]IT 145\n[2]IT 200\n[3]IT 201\n[4]IT 270\n[5]IT 315\n[6]IT 328\n[7]IT 330");

Console.Write("Enter your choice : ");

}

int ReadChoice()

{

string s = Console.ReadLine();

return (int.Parse(s));

}

void WriteChoice(int choice)

{

Console.WriteLine($"Your choice is {choice}");

}

}

}

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, we need to capitalize the first letter in each method and each word that’s in it. Make sure any calls to methods also follow the same conventions. This isn’t really a debug necessarily, but methods are supposed to be formatted like this in C#.

Next, in the ReadChoice() method, after the user types in a number, the string of their input must be returned as an integer. To do this, we can just parse the string variable to be an integer within the return statement.

Then, in the WriteChoice() method, we have to specify the datatype of the parameter choice, which is meant to be an integer based off what we have in the Run() method. Finally, we just have to fix the console print statement to include choice in the output. Personally, I just replace what’s in the brackets with the variable name, put a $ sign before the quotes of the output statement, and remove anything else past the quotes (not including the closing parenthesis and semicolon).

Lastly, test the program to make sure it’s running correctly. I used the same number as the example picture provided among a few others. And that’s it!

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

I learned to parse variables without needing to make new ones! Usually, I create a variable of the datatype I want and cast the old variable with it, but parsing makes it a lot easier, especially when you don’t need to re-use that variable for anything else.