# IT 230 Coding Activity Submission Template

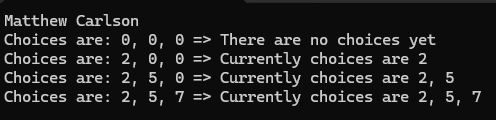
Submit your work on the coding activities for Modules One, Two, Three, Four, and Six in this document. In addition to this document, you should submit a ZIP file containing all your Visual Studio project files and source code that can be run in Visual Studio on a different computer.

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 DebugFixIFStmt

{

class Program

{

static void Main(string[] args)

{

(new Program()).run();

}

void run()

{

int firstChoice = 0, secondChoice = 0, thirdChoice = 0;

System.Console.WriteLine("Matthew Carlson");

firstChoice = 0; secondChoice = 0; thirdChoice = 0;

WriteCurrentChoices(firstChoice, secondChoice, thirdChoice);

firstChoice = 2; secondChoice = 0; thirdChoice = 0;

WriteCurrentChoices(firstChoice, secondChoice, thirdChoice);

firstChoice = 2; secondChoice = 5; thirdChoice = 0;

WriteCurrentChoices(firstChoice, secondChoice, thirdChoice);

firstChoice = 2; secondChoice = 5; thirdChoice = 7;

WriteCurrentChoices(firstChoice, secondChoice, thirdChoice);

}

void WriteCurrentChoices(int firstChoice, int secondChoice, int thirdChoice)

{

if (firstChoice == 0) // Corrected to 'firstChoice' from 'secondChoice'

Console.WriteLine("Choices are: {0}, {1}, {2} => There are no choices yet", firstChoice, secondChoice, thirdChoice);

else if (secondChoice == 0) // Added missing '='

Console.WriteLine("Choices are: {0}, {1}, {2} => Currently choices are {0}", firstChoice, secondChoice, thirdChoice, firstChoice);

else if (thirdChoice == 0) // Removed Extra '='

Console.WriteLine("Choices are: {0}, {1}, {2} => Currently choices are {0}, {1}", firstChoice, secondChoice, thirdChoice, firstChoice, secondChoice);

else // Removed 'if' from else statement

Console.WriteLine("Choices are: {0}, {1}, {2} => Currently choices are {0}, {1}, {2}",

firstChoice, secondChoice, thirdChoice, firstChoice, secondChoice, thirdChoice);

}

}

}

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 assignment was straight forward. I glanced over the run() method to check for any errors, and upon finding none, moved to the WriteCurrentChoices() method. The input looked good, so I moved to the if statements.

The first error was that the first if statement was relying on secondChoice, and therefore was not running as intended, and was instead printing the first statement ‘there are no choices yet’ twice, due to it acting twice and instead of the second if statement.

The second error was a missing ‘=’ in the ‘equal to’ operator. That was a simple mistake to make, and a simple one to solve.

The third error was an *extra* ‘=’ in the ‘equal to’ operator. It needs to be two, or it will just not work.

The fourth and final error was the ‘if’ part of the third ‘else if’ statement. Due to it acting upon the same requirements as the previous statement, it was simply never reached. However, changing it to simply ‘else’ caused it to print when no other conditions were met (when all choices were made) as was intended.

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

This activity was a good refresher for ‘if’ and ‘else if’ statements in C#, just as well as it was good practice with operators and simple coding. Being able to spot out errors and correct them efficiently is important to maintaining and creating a program.

Besides, just like I mentioned previously, I really enjoy debugging. It’s good practice, I get to see how other people write code, what is efficient and what isn’t...etc.