# 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

Description automatically generated

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("Couch's Copy");

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);

// Wait for user input before closing the console window

Console.WriteLine("Press any key to continue . . .");

Console.ReadKey();

}

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

{

if (secondChoice == 0)

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

else if (firstChoice != 0 && secondChoice != 0 && thirdChoice == 0)

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

else if (firstChoice != 0 && secondChoice != 0 && thirdChoice != 0)

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.

To complete this task, I first thoroughly examined the provided code to understand its intended functionality and identify any errors preventing it from running correctly. The goal was to have a program that sequentially updates and displays different combinations of choices and correctly formats the output based on the current values of these choices. The primary issues I identified were syntactic errors in the conditional statements within the WriteCurrentChoices method. Specifically, there were problems with an assignment operator (=) used instead of a comparison operator (==), and a triple equals operator (===) which is not valid in C#. Additionally, the conditions didn't adequately cover all scenarios required to produce the expected output. By correcting these errors and logically organizing the conditions to handle all specified cases, I ensured the program could accurately display the intended messages. To keep the console window open and view the output, I added a Console.ReadKey() at the end of the run method, which waits for user input before closing the window.

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

From completing this activity, I learned the importance of carefully reviewing code to identify both syntactic and logical errors. Debugging requires attention to detail and a methodical approach to test and verify each part of the code. This exercise reinforced my understanding of C# syntax, especially the correct usage of comparison operators, and the importance of clear and logical condition checking. Additionally, I learned practical techniques for maintaining console applications, such as using Console.ReadKey() to prevent the program from exiting immediately. Overall, this activity enhanced my problem-solving skills and provided valuable experience in debugging.