# 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 program

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 DebugFixIFStmt

{

class Program

{

static void Main(string[] args)

{

// runs programs

(new Program()).Run();

}

void Run()

{

// my name!

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

// initialize choices

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

// first line

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

WriteCurrentChoices(firstChoice, secondChoice, thirdChoice);

// second line

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

WriteCurrentChoices(firstChoice, secondChoice, thirdChoice);

// third line

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

WriteCurrentChoices(firstChoice, secondChoice, thirdChoice);

// fourth line

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

WriteCurrentChoices(firstChoice, secondChoice, thirdChoice);

}

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

{

// if all choices equal 0, then there are no choices

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

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

// if the second choice is 0, then prints out the first choice

else if (secondChoice == 0)

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

// if the third choice is 0, then prints out the first and second choice

else if (thirdChoice == 0)

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

// if none of the choices are 0, then prints all of the choices

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.

This program is meant to output all the choices and then point out the choices that are not equal to 0. To do this, multiple if else statements are created to cover as many possibilities there are, including if-statements where multiple things need to be true.

To start debugging, first we need to change the name in console write line statement to ours.

Next, we need to make sure everything is spelled correctly. We can see that in the if-statements, there should be two = signs to compare one thing to another. We can also see that methods aren’t properly capitalized.

Lastly, we need to figure out where in the logic in the if-statements is incorrect. We can see that there are two separate if-statements where secondChoice and thirdChoice equal 0, which is redundant.

For the first if-statement, the statement should be true if ALL choices equal 0. For the second if-statement, the statement should be true if secondChoice AND thirdChoice equal 0. For the third if-statement, the statement should be true if thirdChoice equals 0. Finally, the last statement should be true if ALL choices DO NOT equal 0.

Now you can save it and test it out!

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

So, I had no idea in console print statements you could put numbers in brackets and add variables after what’s in quotes to make them equal the numbers. I always thought what was in quotes would print as such, and that the program couldn’t read things like that (without the $ sign before the quotes). This is an interesting way to print out console statements and can make it easier to read without having a bunch of variable names within the quotes of the print statement.