# IT 230 Coding Activity Submission Template

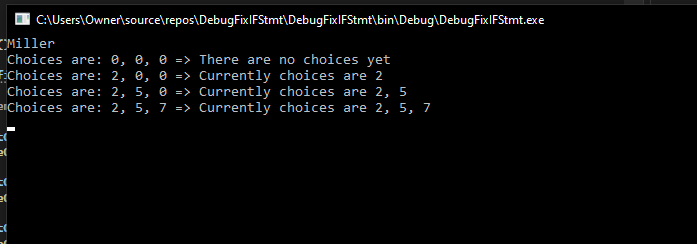
Submit your work on the coding activities for Modules One, Two, Three, Four, and Six in this document. In addition to this document, you should submit a ZIP file containing all your Visual Studio project files and source code that can be run in Visual Studio on a different computer.

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("Miller"); // changed Teacher's Copy to Miller.

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);

Console.ReadLine();

}

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

{

if (firstChoice == 0) // Corrected secondChoice to firstChoice.

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

else if (secondChoice == 0) // There was only one equal sign, corrected to ==.

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

else if (thirdChoice == 0) // Removed the third equal, corrected === to ==.

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

else // Added a space between the else if and the the start of the parentheses. This line was not showing up,

// so I used else in place of else if. This meant the (thirdChoice == 0) was not longer needed and prevented the code from running.

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.

I started off by reading through the provided code. There were not any glaring problems such as broken links and highlighted sections of errors. I noticed in the WriteCurrentChocies program it included secondChoice and thirdChoice twice respectively. The first if statement needed to be corrected to include firstChocie instead of secondChoice. The firstChoice had not been called at all. This still left the thirdChoice called twice, but I had not yet focused on that. Instead, I corrected the must equal statements, where one was a single equal and one was a triple. They were both corrected to be double, ==. I then added a space in-between the else if and parentheses in the last if else statement.

After debugging the program would open and close without displaying its output long enough to see. I added a ReadLine to ensure the output would be displayed before closing. I then realized the last line was not being printed and changed the statement to an else instead of if else. The (thirdChoice == 0) was now no longer needed and was hinting it required a semi-colon, but this section just needed to be removed. The program then put out the desired results.

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

It is good to double check that the operators are entered correctly and to check for simple mistakes. This can include spelling, spacing, capitalization, or even missing components. It is good to click on sections, such as firstChoice, to easily and quickly see where there are referenced elsewhere in the program. This made it very easy to see that it was not called in anything but the writeLine sections of the program. I do not think a lot of the hints provided in the tooltips are helpful yet. The solutions and hints I have encountered from it so far are not actually helpful yet, but I am hoping to see that change in the future. Changing the final else if statement should include all other options and end the cycle. This meant it was no longer iterating through for options that were not perfectly equal.