

IT 230 Coding Activity Submission Template

**Name:** Ryan Hutton

**Date:** 11/19/2019

**Class:** IT 230

**Module:** 4 Coding Activity – Debug/Fix If statements

|  |  |
| --- | --- |
| **1.** | Insert a copy of your project of the ZIP file of all of your Visual Studio project files here so that it can be loaded and run in another Visual Studio: |
| Insert here a copy of your \*.cs source code text you used here (copy and paste source code here, do **not** simply insert \*.cs files):  /\* Ryan Hutton  \* Module 4 Coding Activity - Debug and fix If Statements  \* SNHU IT 230 - Software Development with C#.NET  \* 11/19/2019  \*/  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("Ryan Hutton");  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) // Changed secondChoice to firstChoice to satisfy non-zero precedence rule.  Console.WriteLine("Choices are: {0}, {1}, {2} => There are no choices yet", firstChoice, secondChoice, thirdChoice);  else if (secondChoice == 0)  Console.WriteLine("Choices are: {0}, {1}, {2} => Currently choices are {0}", firstChoice, secondChoice, thirdChoice, firstChoice);  else if (thirdChoice == 0)  Console.WriteLine("Choices are: {0}, {1}, {2} => Currently choices are {0}, {1}", firstChoice, secondChoice, thirdChoice, firstChoice, secondChoice);  else if(thirdChoice != 0 ) // Changed == to != to check if thirdChoice did not contain the value of zero.  Console.WriteLine("Choices are: {0}, {1}, {2} => Currently choices are {0}, {1}, {2}",  firstChoice, secondChoice, thirdChoice, firstChoice, secondChoice, thirdChoice);  }  }  } |
| **2.** | Insert a screenshot here of the output that resulted from running your program, showing your last name as the first printed text to the screen: |
| **3.** | Explain the design of your program, the steps you took to complete it, and how you coded it:  The first thing I did to complete this assignment, was to download the entire project to my computer. Then I uploaded the project into Visual Studio. Next, I examined the code to identify any syntax errors so I can get an executable program. Line 47 has only one “=” sign in the if statement when it is expecting a Boolean statement “else if (secondChoice = 0)” [incorrect]. To correct this, I made the “=” operator a double equal sign. Similarly, in Line 49, there was an extra equal sign in the if statement, so I deleted it. Once the syntax errors were completed, I ran the code to thoroughly figure out the logic. After reading the activity requirements a couple more times, I realized that a couple of if statements needed minor changes to satisfy the requirements. The next change I did was on Line 45, where I change the variable secondChoice to firstChoice to satisfy the non-zero precedence rule because if firstChoice is 0, and all variable must go in order, then there are no choices. I then tested the code for this condition, and it worked. Next, I changed the equality operator in the if statement on Line 51 to != because there was already a test condition in the previous line to check if thirdChoice was zero. If thirdChoice is not zero, then execute this if statement. Lastly, I ran the code and the output worked as expected. |
| **4.** | Reflect on this experience and the lessons you learned from it:  This activity was valuable because it emphasized the importance of understanding and interpreting decisions statements and logic. Another thing I learned, not related to coding specifically is that sometimes the best solution is the simplest one and there is no need to overcomplicate the solution. These skills will help me as a future software developer as they will enhance my problem-solving abilities. |