# 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

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.

int ValidateChoice(int choice, int firstChoice, int secondChoice, int thirdChoice, int totalCredit)

{

if (choice < 1 || choice > 7) //was choice > 70 changed to choice > 7

return -1;

else if (choice == firstChoice || choice == secondChoice || choice == thirdChoice) //changed &&(AND) operators to ||(OR) operators

return -2;

else if (totalCredit == 9) // was totalCredit > 9 changed to totalCredit == 9

return -3;

return 0; //was return -4 changed to return 0

}

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 small program is a way for a student to register for three courses of their choice. It takes their input, recognizes the input to match the class associated with their selected number, and outputs it as a successful registration. In the “int ValidateChoice” section there were a couple of errors preventing the project from running and giving an output. if (choice < 1 || choice > 70) is incorrect because there is no options above 7. else if (choice == firstChoice && choice == secondChoice && choice == thirdChoice) I changed the AND operators (&&) to OR operators(||) to fix an issue where you could register for the same course multiple times. else if (totalCredit > 9) was changed to == 9 because since we have totalCredit +=3, we want to make sure a student cannot go above 3 course when reaching 9 total credits. Lastly, return -4 was incorrect because there was no case -4 to run, and as such case 0 was never able to run because it wasn’t being returned properly so I changed it to return 0.

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

This program was very intricate and complex to look through and figure out. As I began looking through it, ideas of if statements and adding more cases came to my mind but I needed to keep it simple and try to solve the main problems at hand. This activity was really interesting because we are essentially taking on the role of a code tester/editor looking through a colleague’s code. In the end I realized the issues lied in the int ValidateChoice(int choice, int firstChoice, int secondChoice, int thirdChoice, int totalCredit). Most errors like choice > 7, == 9, and return 0 where all pretty easy fixes and allows you to run the program. However, there is one that is sneaky until you actually go into the program and test the different choices, intentionally trying to break the program. Without changing the logical operators AND to OR, you can register for the same class over and over. This isn’t what we want so once you realize that was a problem, it’s pretty easy to realize where the issue was. Overall, this project helped my debugging process more.