# 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.
2. using System;
3. namespace ConsoleRegisterStudent
4. {
5. class Program
6. {
7. static void Main(string[] args)
8. {
9. new Program().run();
10. }
11. void run()
12. {
13. int choice;
14. int firstChoice = 0, secondChoice = 0, thirdChoice = 0;
15. int totalCredit = 0;
16. string yesOrNo = "";
17. Console.WriteLine("Teacher's Copy");
18. do
19. {
20. WritePrompt();
21. choice = Convert.ToInt32(Console.ReadLine());
22. switch (ValidateChoice(choice, firstChoice, secondChoice, thirdChoice, totalCredit))
23. {
24. case -1:
25. Console.WriteLine("Your entered selection {0} is not a recognized course.", choice);
26. break;
27. case -2:
28. Console.WriteLine("You have already registered for this {0} course.", ChoiceToCourse(choice));
29. break;
30. case -3:
31. Console.WriteLine("You cannot register for more than 9 credit hours.");
32. break;
33. case 0:
34. Console.WriteLine("Registration Confirmed for course {0}.", ChoiceToCourse(choice));
35. totalCredit += 3;
36. if (firstChoice == 0)
37. firstChoice = choice;
38. else if (secondChoice == 0)
39. secondChoice = choice;
40. else if (thirdChoice == 0)
41. thirdChoice = choice;
42. break;
43. }
44. WriteCurrentRegistration(firstChoice, secondChoice, thirdChoice);
45. Console.Write("\nDo you want to try again? (Y|N)? : ");
46. yesOrNo = Console.ReadLine().ToUpper();
47. } while (yesOrNo == "Y");
48. Console.WriteLine("Thank you for registering with us");
49. }
50. void WritePrompt()
51. {
52. Console.WriteLine("Please select a course for which you want to register by typing the number inside []");
53. Console.WriteLine("[1] IT 145\n[2] IT 200\n[3] IT 201\n[4] IT 270\n[5] IT 315\n[6] IT 328\n[7] IT 330");
54. Console.Write("Enter your choice : ");
55. }
56. int ValidateChoice(int choice, int firstChoice, int secondChoice, int thirdChoice, int totalCredit)
57. {
58. if (choice < 1 || choice > 7)
59. return -1;
60. else if (choice == firstChoice || choice == secondChoice || choice == thirdChoice)
61. return -2;
62. else if (totalCredit + 3 > 9)
63. return -3;
64. return 0;
65. }
66. void WriteCurrentRegistration(int firstChoice, int secondChoice, int thirdChoice)
67. {
68. if (firstChoice != 0) Console.Write("You are currently registered for {0}", ChoiceToCourse(firstChoice));
69. if (secondChoice != 0) Console.Write(", {0}", ChoiceToCourse(secondChoice));
70. if (thirdChoice != 0) Console.Write(", {0}", ChoiceToCourse(thirdChoice));
71. Console.WriteLine();
72. }
73. string ChoiceToCourse(int choice)
74. {
75. switch (choice)
76. {
77. case 1: return "IT 145";
78. case 2: return "IT 200";
79. case 3: return "IT 201";
80. case 4: return "IT 270";
81. case 5: return "IT 315";
82. case 6: return "IT 328";
83. case 7: return "IT 330";
84. default: return "";
85. }
86. }
87. }
88. }

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.

So this code needed to be corrected so that the end user can choose from the list of courses and register them based on the corresponding number next to the course. Then the user needed to be able to register for more courses if need be and the program store the previously registered courses as well. The code provided had a few glaring issues that needed to be addressed for it to execute correctly. The first thing I noticed was that the method ‘VallidateChoice` was checking for a range that was greater than 70 instead of between 1 and 7. I then noticed that the expression choice==firstChoice && choice == secondChoice && choice == thirdChoice was not set up to correctly tell if the course was already registered for. Then I noticed that totalCredit > 9 didn’t actually restrict the registration to a max of 9 credit hours. So that had to be changed too. Then lastly I noticed that ValidateChoice should have actually returned a 0 instead of a -4.

So I used the following statement to correct these items:.

if (choice < 1 || choice > 7)

return -1;

else if (choice == firstChoice || choice == secondChoice || choice == thirdChoice)

return -2;

else if (totalCredit + 3 > 9)

return -3;

return 0;

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

I learned a lot about how to correctly write if and else if statements that had specific conditions to check for based on user input. I also learned a lot about the validation method as it basically looks back into what the user has entered and determines whether not it is valid. I also learned about correcting the duplicate course registration logic as it was throwing off the whole program due to a simple character being wrong.