# IT 230 Coding Activity Submission Template

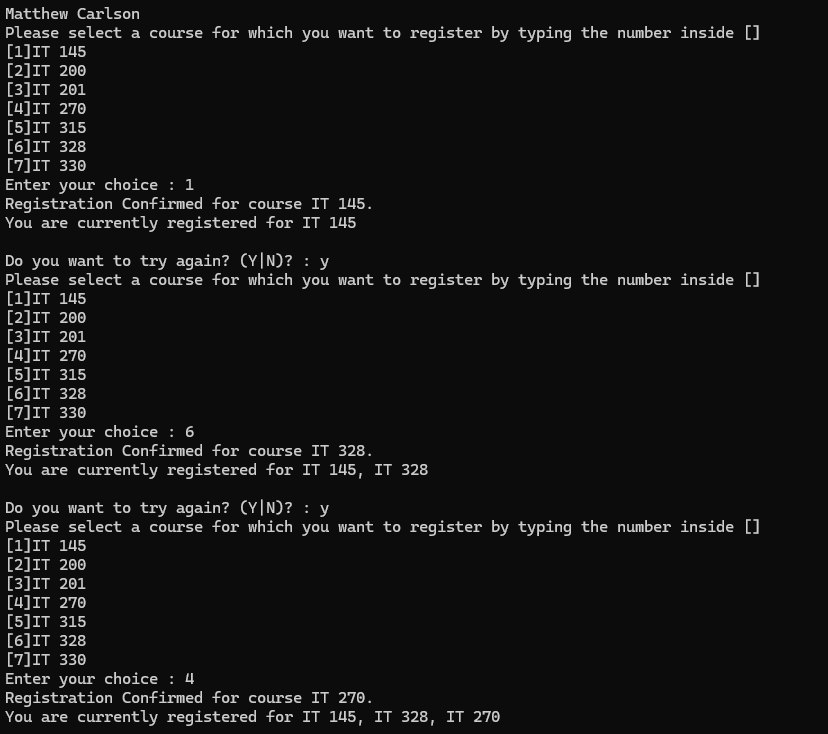
Submit your work on the coding activities for Modules One, Two, Three, Four, and Six in this document. In addition to this document, you should submit a ZIP file containing all your Visual Studio project files and source code that can be run in Visual Studio on a different computer.

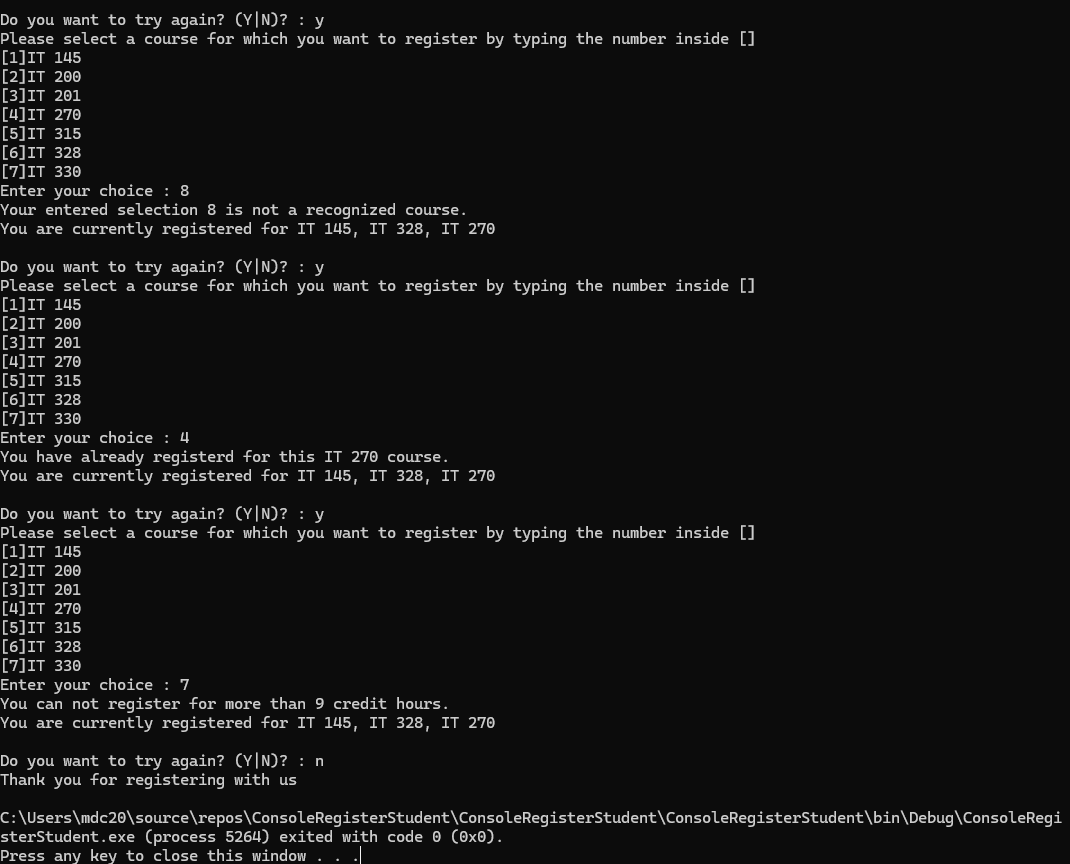
For each coding activity, complete the following steps:

* 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 ConsoleRegisterStudent

{

class Program

{

static void Main(string[] args)

{

(new Program()).run();

}

void run()

{

int choice;

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

int totalCredit = 0;

string yesOrNo = "";

System.Console.WriteLine("Matthew Carlson");

do

{

WritePrompt();

choice = Convert.ToInt32(Console.ReadLine());

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

{

case -1:

Console.WriteLine("Your entered selection {0} is not a recognized course.", choice);

break;

case -2:

Console.WriteLine("You have already registerd for this {0} course.", ChoiceToCourse(choice));

break;

case -3:

Console.WriteLine("You can not register for more than 9 credit hours.");

break;

case 0:

Console.WriteLine("Registration Confirmed for course {0}.", ChoiceToCourse(choice));

totalCredit += 3;

if (firstChoice == 0)

firstChoice = choice;

else if (secondChoice == 0)

secondChoice = choice;

else if (thirdChoice == 0)

thirdChoice = choice;

break;

}

WriteCurrentRegistration(firstChoice, secondChoice, thirdChoice);

Console.Write("\nDo you want to try again? (Y|N)? : ");

yesOrNo = (Console.ReadLine()).ToUpper();

} while (yesOrNo == "Y");

Console.WriteLine("Thank you for registering with us");

}

void WritePrompt()

{

Console.WriteLine("Please select a course for which you want to register by typing the number inside []");

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");

Console.Write("Enter your choice : ");

}

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

{

if (choice < 1 || choice > 7) // Changed '70' to '7'

return -1;

else if (choice == firstChoice || choice == secondChoice || choice == thirdChoice) // Changed AND operator '&&' to OR operator '||'

return -2;

else if (totalCredit >= 9) // Changed '>' to '>='

return -3;

return 0; // Changed return from -4 (unused) to 0

}

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

{

if (secondChoice == 0)

Console.WriteLine("You are currently registered for {0}", ChoiceToCourse(firstChoice));

else if (thirdChoice == 0)

Console.WriteLine("You are currently registered for {0}, {1}", ChoiceToCourse(firstChoice), ChoiceToCourse(secondChoice));

else

Console.WriteLine("You are currently registered for {0}, {1}, {2}", ChoiceToCourse(firstChoice), ChoiceToCourse(secondChoice), ChoiceToCourse(thirdChoice));

}

string ChoiceToCourse(int choice)

{

string course = "";

switch (choice)

{

case 1:

course = "IT 145";

break;

case 2:

course = "IT 200";

break;

case 3:

course = "IT 201";

break;

case 4:

course = "IT 270";

break;

case 5:

course = "IT 315";

break;

case 6:

course = "IT 328";

break;

case 7:

course = "IT 330";

break;

default:

break;

}

return course;

}

}

}

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 was a very relatively simple project to fix up. It was just like the last one, only a little bit larger and required just a bit more attention to detail. My first step, of course, was reading through the program, noting what input each method took, and figuring that out. When everything looked relatively clean, I turned my gaze to ValidateChoice() because it was the only place I found issues.

My first spotted issue was the end return. It was returning –4 for some reason, despite –4 being unused in the WritePrompt() switch. I decided that, based on WritePrompt()’s case 0, it was meant to be 0.

Then, it was just top to bottom. I changed ‘choice > 70’ to ‘choice > 7’ because...well, there are only 7 choices, not 70.

I changed the ‘AND’ operators in the first else to ‘OR’ because otherwise it would only trigger if choice was equal to *every other choice.*

Then, I turned the ‘>’ in the second else to ‘>=’. This will check if the number of credits *is* 9, rather than only checking if it went above it. One could reliably use ‘==’ instead, but using ‘>=’ ensures that if the number somehow magically goes above 9 (it really, really shouldn’t...) it will still be caught.

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

Once again, this project was a great practice experience for handling operator issues within C#. I have to admit that I was a *little* bit disappointed by the amount of work present in this one (we didn’t need to handle input validation or even write a switch ourselves) but it was still good practice. I do wish there were a few more errors sprinkled in, though---this project barely took me 15 minutes.