# IT 230 Coding Activity Submission Template

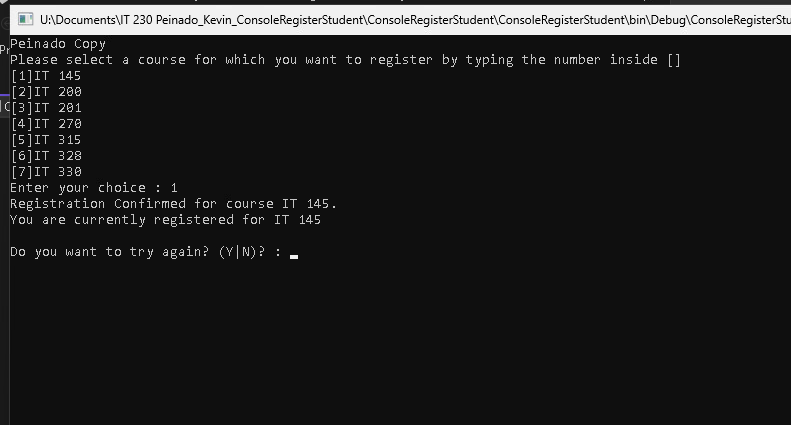
Submit your work on the coding activities for Modules One, Two, Three, Four, and Six in this document. In addition to this document, you should submit a ZIP file containing all your Visual Studio project files and source code that can be run in Visual Studio on a different computer.

For each coding activity, complete the following steps:

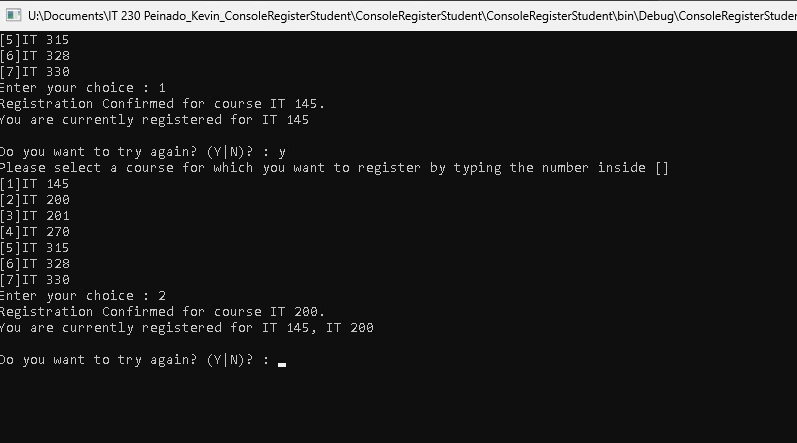
* Download and rename this document to meet the file naming conventions requested in the assignment instructions.
* Fill in the required information below by replacing the bracketed text with the relevant information.
* Submit this document and your ZIP file for grading and feedback. Your ZIP file should follow the same naming conventions.

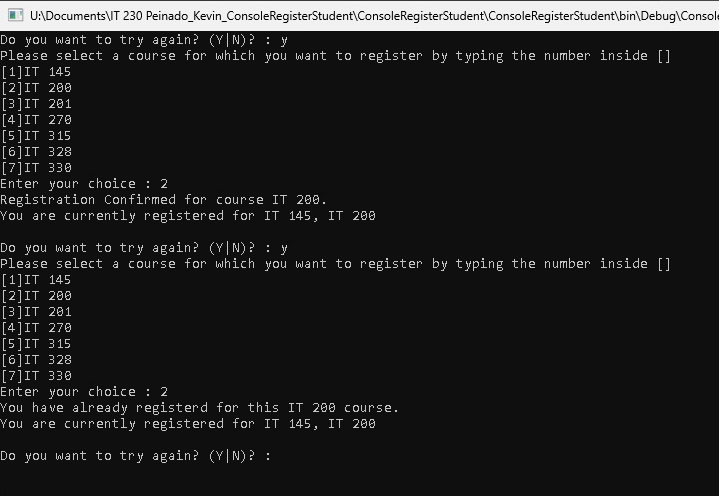
Document your work in the coding activity by completing each of the following items:

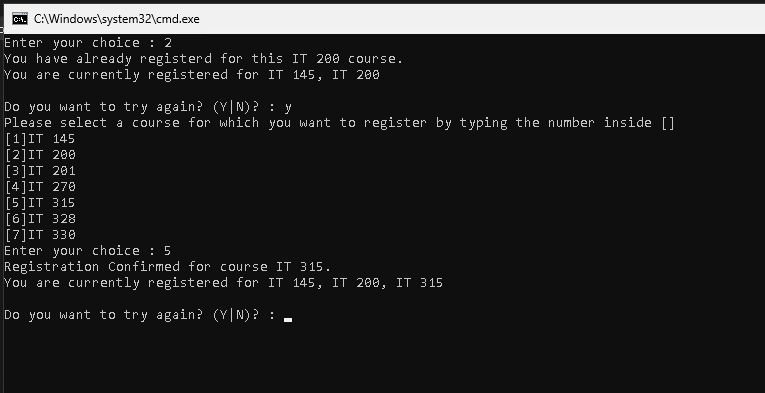
1. Provide a screenshot of the output that resulted from running your program successfully in Visual Studio. See the coding assignment instructions for an example of what should be included in the screenshot. Your screenshot must include the following elements:
   1. Your last name as the first printed text on the screen
   2. Verification that the program is fully functioning and data results are accurate for the given problem

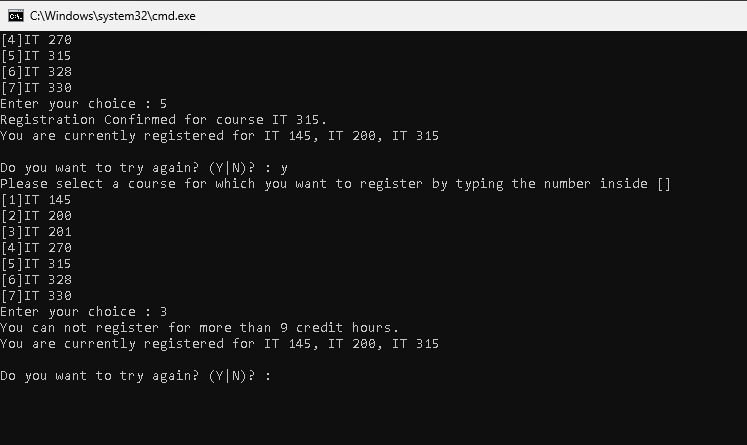
Screenshot 1:

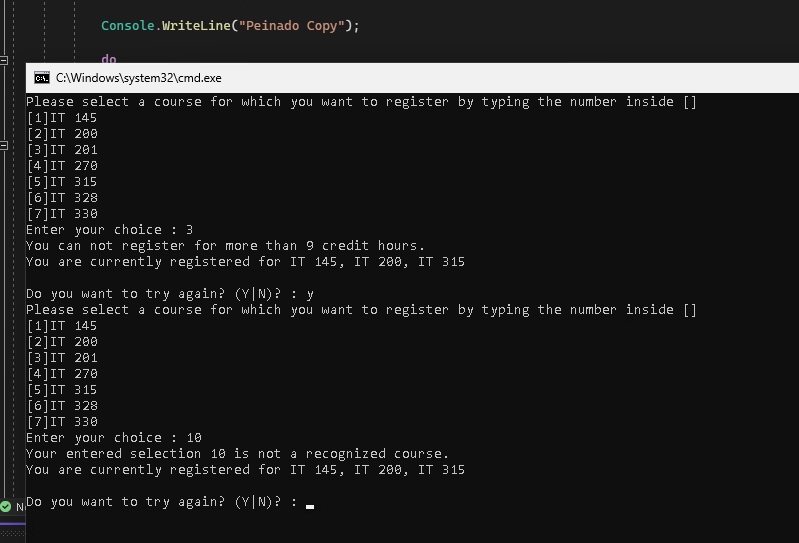
Screenshot 2:

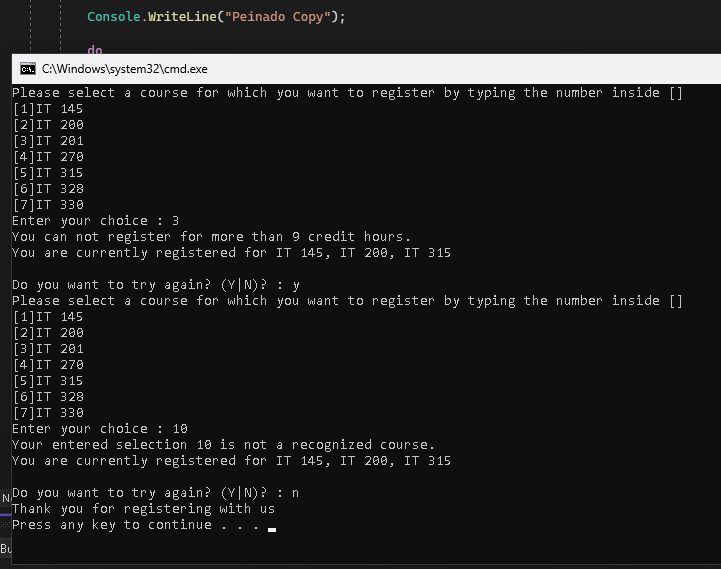


Screenshot 3:

Screenshot 4:

Screenshot 5:

Screenshot 6:

Screenshot 7:

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.

Source code text:

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 = "";

Console.WriteLine("Peinado Copy");

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)

return -1;

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

return -2;

else if (totalCredit + 3 > 9)

return -3;

return 0;

}

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

{

if (firstChoice == 0)

Console.WriteLine("You have not registered for any courses yet.");

else 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)

{

switch (choice)

{

case 1: return "IT 145";

case 2: return "IT 200";

case 3: return "IT 201";

case 4: return "IT 270";

case 5: return "IT 315";

case 6: return "IT 328";

case 7: return "IT 330";

default: return "Unknown 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.

Summary of Fixes in ConsoleRegisterStudent Program:

The ConsoleRegisterStudent program was intended to allow students to register for up to three courses (each worth three credit hours) while enforcing business rules, such as preventing duplicate course registration and ensuring no more than nine credit hours are selected. However, the initial code contained several logical and syntax errors, preventing it from functioning as required. The errors included incorrect validation logic, incorrect range checks, and improper conditional statements.

Steps Taken to Fix the Program:

    1.    Fixed Course Selection Validation:

The original ValidateChoice method allowed selections up to 70 (if (choice < 1 || choice > 70)). In order to fix this I changed this to if (choice < 1 || choice > 7) since only seven courses were available.

    2.    Corrected Duplicate Course Registration Logic:

The condition if (choice == firstChoice && choice == secondChoice && choice == thirdChoice) was incorrect, due to it checked if all three choices were the same instead of checking against each registered course separately. I was able to fixe this by updating the condition to:

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

    return -2;

    3.    Fixed Credit Hour Validation:

The condition if (totalCredit > 9) incorrectly checked if the total was greater than nine after allowing registration. To fix this I updated to if (totalCredit + 3 > 9), preventing students from exceeding the credit limit before registering.

    4.    Fixed Course Registration Return Code:

 The original ValidateChoice method returned -4 by default, which was not handled in the switch statement. Ti fix this I updated it to return 0 for valid selections, ensuring successful course registration.

    5.    Corrected Course Registration Output Formatting:

Modified the WriteCurrentRegistration method to correctly display registered courses when only one or two were selected.

After these fixes were implemented the program successfully registered students for courses while enforcing the correct rules.

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

Fixing the ConsoleRegisterStudent program was a valuable learning experience in debugging and problem solving. The process helped reinforce key programming concepts such as input validation, conditional logic, and control structures. One of the main challenges was identifying logical errors in the course selection validation, specifically with duplicate course registration and credit hour limits. Through careful testing and step-by-step debugging, I was able to refine the conditions and ensure the program followed the rules that were intended for the program.

This exercise also highlighted the importance of thoroughly testing certain logic, such as selecting the same course multiple times or exceeding the credit limit. It also emphasized the value of clear and structured conditional checks to prevent unintended behavior. Overall, this project improved my ability to analyze and debug C# code effectively while reinforcing best practices in validation and user interaction.