# IT 230 Coding Activity Submission Template

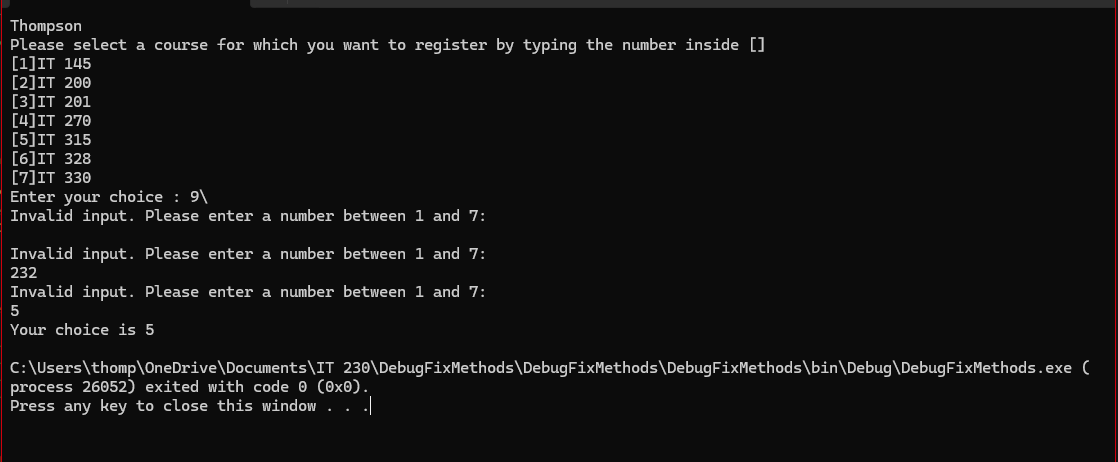
Submit your work on the coding activities for Modules One, Two, Three, Four, and Six in this document. In addition to this document, you should submit a ZIP file containing all your Visual Studio project files and source code that can be run in Visual Studio on a different computer.

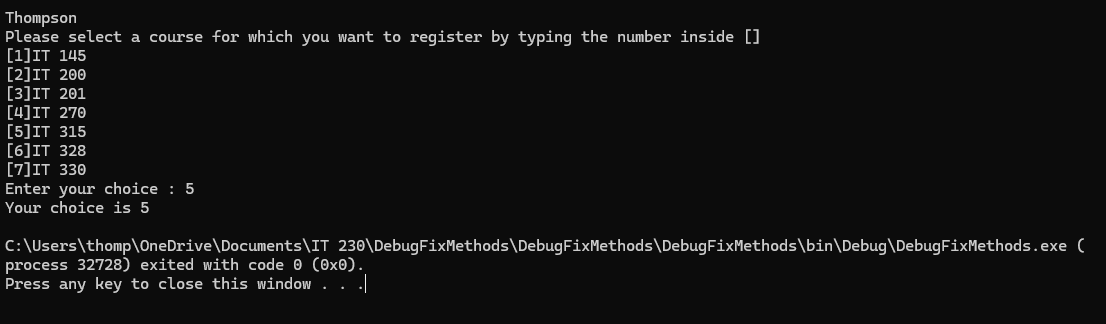
For each coding activity, complete the following steps:

* 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; //Only this line is needed as all else can be called under System

namespace DebugFixMethods

{

class Program

{

static void Main(string[] args)

{

(new Program()).Run(); //Needed to be "Run" as method names are case sensitive

}

void Run()

{

int choice = 0;

WritePrompt();

choice = ReadChoice();

WriteChoice(choice);

}

void WritePrompt()

{

Console.WriteLine("Thompson");

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 ReadChoice() //Rewritten to properly convernt input string to int and validate input

{

while (true)

{

string input = Console.ReadLine(); // Get user input

int choice;

// Validate if the input can be parsed to an integer and is within the range 1 to 7

if (int.TryParse(input, out choice) && choice >= 1 && choice <= 7)

{

return choice; // Return valid input

}

else

{

Console.WriteLine("Invalid input. Please enter a number between 1 and 7:"); // Prompt user again for valid input

}

}

}

void WriteChoice(int choice) //Parameter needs type declaration

{

Console.WriteLine("Your choice is {0}", choice);

}

}

}

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.

The program had a few key errors that needed to be resolved for the program to properly run:

-The program only needed the first line “using System” at the head of the program because all other lines after were unnecessarily calling other methods that can be called directly when needed.

-The “Run” method called under Main needed to be capitalized to both match formatting standards as well match the method definition.

-The ReadChoice method needed to be rewritten so that it wasn’t trying to return a string instead of an integer, and also needed an error exception coded in so that it could handle input outside of the expected parameters.

-The WriteChoice method was also missing the type declaration for the choice parameter.  
  
Once these were resolved, the program was able to be run successfully and the desired output obtained.

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

This exercise gave me a good refresher on debugging and improving a more complex program. Working out the logic needed to rewrite the ReadChoice method threw me through a loop at first, but after referencing some other similar use cases I was able to work out the necessary logic to get my desired result. Originally, I tried to write a nested if statement but realized that wasn’t the most concise or efficient way to handle this logic and so I ended up writing this as a while loop. I look forward to the next challenge and being able to apply my abilities in more complex ways.