We will be using this problem solving method throughout the Bootcamp, as a recommended approach for the students to use. The students are free to use any approach that they desire, however we are presenting this approach as one alternative for the students to consider.

The method has 4 steps:

1. **GOAL:** Understand and define what the exercise is asking the student to do. What is the output/result of the exercise, and what does the student's code need to return as a solution?

2. **INPUT:** Understand and list what the exercise provides as input. If the requirement is to write a function, what is being passed to the function as inputs. What examples are provided? Is there any test data that the exercise gives to work with?

3. **STRATEGY:** List, in plain language (not code), the steps that you will take to get from input to output. List the steps in order, numbering them from (1) to (X). Initially, this step should be a complete listing of the steps that the student will follow. However, as the student is working through the actual coding (in step 4 below), they may determine that the order of steps may change, or that steps may need to be added or removed, so they may come back to this step and edit appropriately. By the end of the exercise, this listing should be a plain language reference of what the actual code is doing, step by step. While going back and forth to this step during coding may seem tedious, the value in doing so is that the code becomes self-documenting, and this documentation in the code greatly assists the student to understand what they have done and their thought process, when they come back to the notebook at a later time.

4. **SOLUTION:** Write the code here. If writing a function, start by defining the data structure (variable) to be returned by the function and the return statement at the end of the function. Then code through the steps in the Strategy, with liberal use of print() statements to understand what the code is doing at the various steps of the process.