**Problem Solving Methodology**

* Understand the Problem
* Explore the concrete Examples
* Break It Down
* Solve/Simplify
* Look Back and Refractor

\*\*\*\*Understand that it is about the process not necessarily the solution\*\*\*\*

**Understand the Problem**

Steps:

1. Can I restate the problem in my own words?
2. What are the inputs that go into the problem?
3. What are the outputs that should come from the solution to the problem?
4. Can the outputs be determined from the inputs?
   1. In other words – Do I have enough information to solve the problem?
      1. You may not be able to answer this question until you set about solving the problem.
5. How should I label the important pieces of data that are part of the problem?

**Explore Examples**

Steps:

1. Start with simple examples.
   1. Have simple use cases.
2. Explore example with empty inputs.
3. Explore example of invalid inputs.

**Break It Down**

Steps:

1. Comment out your code.
   1. E.g., // loop over the string.
2. Write the steps that you may take.
   1. Walk through the steps .

**Solve/Simplify**

Steps:

1. Solve the Problem
   1. If you can’t 🡪 Solve a simpler Problem.
2. Simplify
   1. Find the core difficulty in what you’re trying to do
   2. Temporarily ignore that difficulty
   3. Write a simplified solution
   4. Incorporate the difficult part back in

**Look Back & Refractor**

Steps:

1. Can you check the result?
2. Can you derive the result differently?
3. Can you understand it at a glance?
4. Can you use the result or method for some other problem?
5. Can you improve the performance of your solution?
6. Can you think of other way to refractor?
7. How have other people solved this problem?