PROBLEM SOLVING

A. Understanding the problem

To gain clarity and understanding of the problem, write it down on paper, reword it in plain English until it makes sense to you.

B. Three steps in Problem Solving

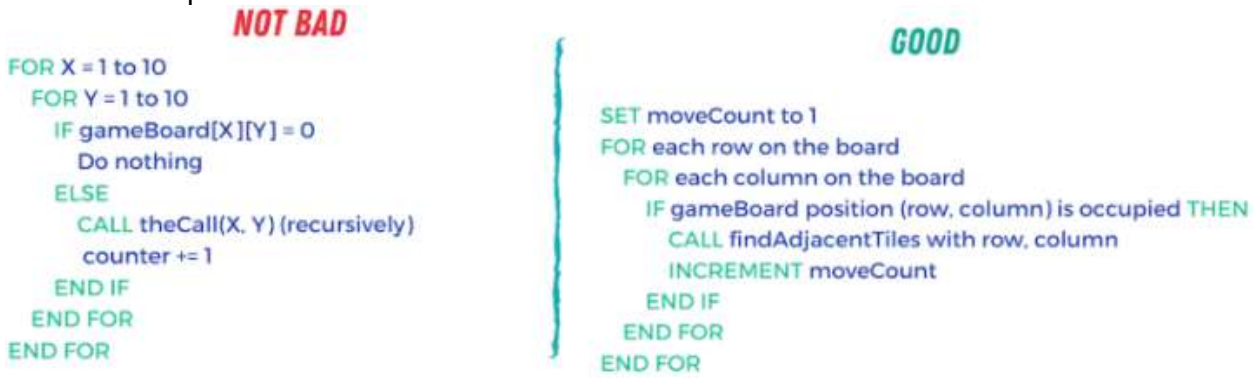
- Plan (SKIP read this)

  Some of the questions you should answer, like:
  - Does your program have a user interface? What will it look like?
  - What inputs will your program have? Will the user enter data or will you get input from somewhere else?
  - What's the desired output?
  - Given your inputs, what are the steps necessary to return the desired output?
- Pseudocode

  Pseudocode is writing out(step-by-step) the logic for your program in natural language instead of code. Example:

```
1  When a user inputs a number
2  Loop from 1 to the entered number
3  If the current number is divisible by 3 then print "Fizz"
4  If the current number is divisible by 5 then print "Buzz"
5  If the current number is divisible by 3 and 5 then print "FizzB
6  Otherwise print the current number
```

  Another example:



- Divide and Conquer (memecah/membagi dan menaklukan)

  You should have identified some subproblems of the big problem you're solving. Many beginners try to solve the big problem in one go. **Don't do this**. If the problem is sufficiently complex, you'll make life a lot harder for yourself. In short, break the big problem down and solve each of the smaller problems until you've solved the big problem.

C. Example

Fizz Buzz Problem

1) make it clearer by rewording it(the problem with your word).
2) make plan, what input & output looks like(will it have an interface or just in console)
3) writing the pseudocode
4) dividing and conquering. as we dividing problem(in the pseudocode), then we solve the subproblem one by one(with coding).

D. How to Think Like a Programmer

Example: write a program that reads ten numbers and figures out which number is the third highest.

If you're stuck, you should reduce the problem to something simpler. Instead of the third-highest number, what about finding the highest overall? Still too tough? What about finding the largest of just three numbers? Or the larger of two?

How to Think:

1) debug

2) reasses Take a step back. Look at the problem from another perspective. Is there anything that can be abstracted to a more general approach? Another way of reassessing is starting anew. Delete everything and begin again with fresh eyes. I'm serious. You'll be dumbfounded at how effective this is.

3) Search (on google) someone has probably solved it. Find that person/solution. In fact, do this even if you solved the problem! (You can learn a lot from other people's solutions). Caveat: Don't look for a solution to the big problem(just for sub-problem).