8 Steps to Solving a Programming Problem

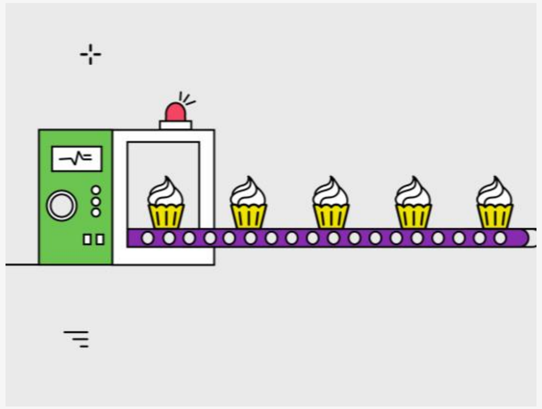
TestLeaf

Always Ahead

First 4 Steps

Describe it manually

Work through with at least multiple sets of sample data



Write pseudocode

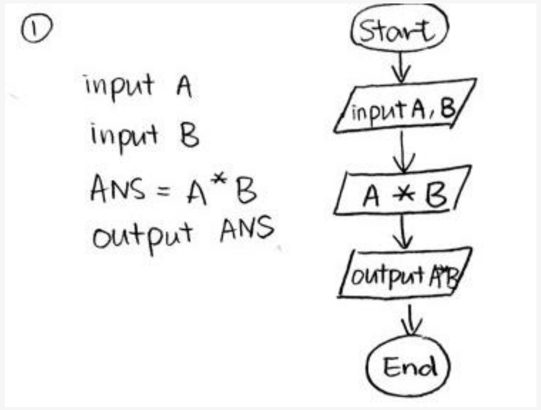
Focus on the logic and steps.

Understand the Problem

Read the problem at least few times.

Simplify your solution

Simplify and optimize your steps



```
①
input A
input B
ANS = A*B
output ANS
```

1 : Understand the Problem

You can't solve a problem you don't understand

- ✓ What is the intended goal of this problem?
- ✓ What are the inputs to this program?
- ✓ What is the expected output of the program?
- ✓ Have I worked on similar problem before?
- ✓ I did not understand the problem, then read it or ask for details.

Program to find duplicate numbers

Ask right questions to get more details.

✓ What is the intended goal of this problem?

Identify duplicate numbers in an input

✓ What are the inputs to this program?

Ask : Array

✓ What is the expected output of the program?

None may be just printing dups will do.

✓ Have I worked on similar problem before?

No

✓ I did not understand the problem, then read it or ask for details.

2: Work through the problem manually

Create 3 sample data

Think of at least three sets of sample data.

Corner / Edge case

It occurs only at an extreme operating parameter

Draw the steps

It is easy to gloss over the steps – one by one.

Run steps with 3 data

Validate if the steps are good for all data – else redefine

3: Simplify & Optimize the steps

TestLeaf
Always Ahead

Patterns

Look for patterns and see if there's anything you can generalize

Is it complex?

Complex problems can be broken into smaller sub problems

Number of steps

Consider the costs of implementing different solutions

Alternate Solutions

Watch out for alternate steps by the performance

4: Write pseudocode

1

Every Step to a Line

Write pseudocode line by line

2

Syntax – Not necessary

Don't get caught up with the syntax

3

Find your logic

Focus on the logic and steps.

4

Relook at logic

Revise the logic on each line.

Last 4 Steps

Simplify your code

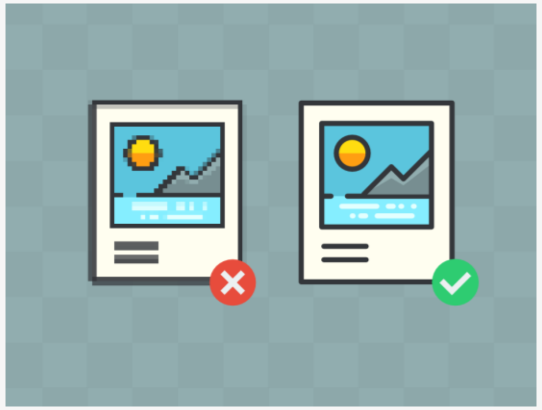
Code better and optimize the code.

Practice, Practice !

With each problem you solve, the better a developer you become

Pseudo to Java Code

Translate your pseudo to Java Program step by step.



Debug your code

Debug your code to confirm the expected works.

5: Convert Pseudo -> Code

TestLeaf
Always Ahead

Translate

Translate each line into real java code with comments.

Skip

Unknown? Don't worry ! Write comments and Move On.

Repository

Refer to your Java Class – Methods Repository for your correct syntax.

Validate

Check if values and code are behaving as expected.

6: Simplify & Optimize your code

Goals?

What are your goals for simplifying and optimizing?

Readable?

Someone couldn't read your code, then it require improvement!

Performance?

How much milliseconds it takes to run your code?

Reusable?

Repeating some steps a lot? See if you can define in another method.

Edge Cases?

Does your code cover the edge cases?

7: Debug

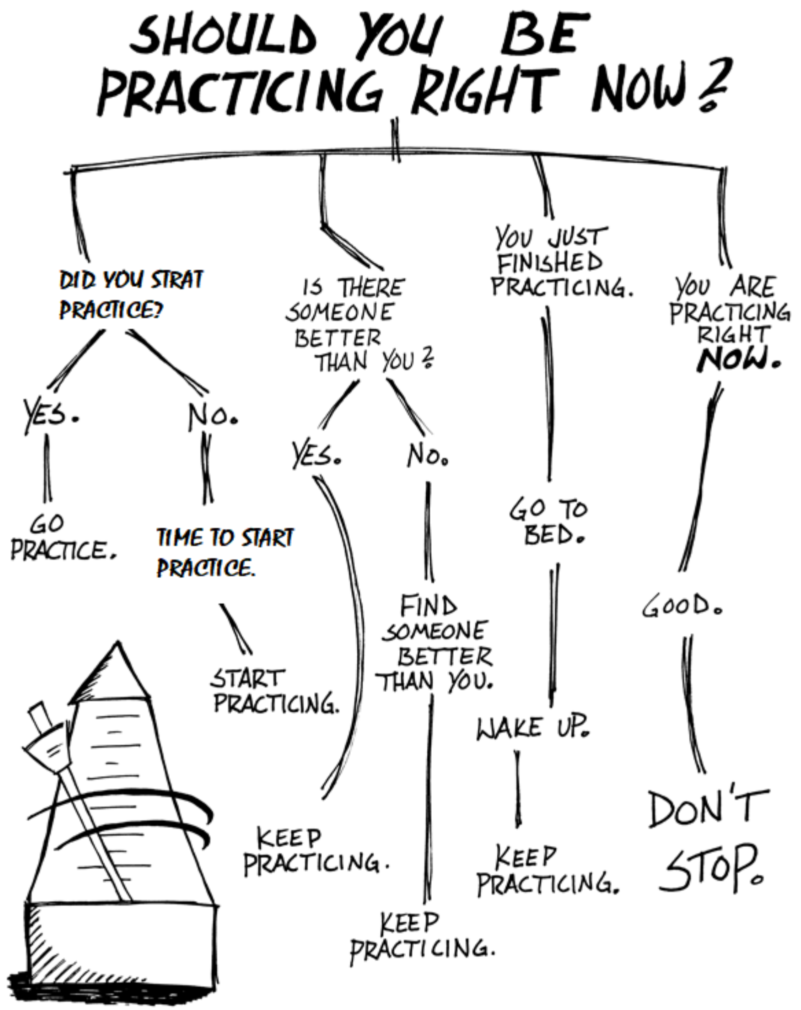
1. Check the console to see what the (error) message says
2. Comment lines of code and output what you coded so far to quickly see if the code is behaving as expected.
3. Use other sample data if there are scenarios you did not think of and see if the code will still work.
4. Save different versions of my file if you am trying out a completely different approach

8: Practice, Practice, Practice

With each problem you solve, the better a developer you become.

Celebrate each success and be sure to remember how far you've come.

Remember that programming, like with anything, comes easier and more naturally with time.


SHOULD YOU BE PRACTICING RIGHT NOW?
DID YOU STRAT PRACTICE?
YES.
GO PRACTICE.
NO.
TIME TO START PRACTICE.
START PRACTICING.
IS THERE SOMEONE BETTER THAN YOU?
YES.
KEEP PRACTICING.
NO.
FIND SOMEONE BETTER THAN YOU.
KEEP PRACTICING.
YOU JUST FINISHED PRACTICING.
GO TO BED.
WAKE UP.
KEEP PRACTICING.
YOU ARE PRACTICING RIGHT NOW.
GOOD.
DON'T STOP.