Good / Bad

Do Modular → divide your code in methods (functions)
↓
Single responsible

User I/P a, b, c, d → int

O/P => $\frac{a+b}{c+d}$

```
PSVM(){
 sout((a+b)/(c+d))
}
```

```
PSVM(){
  int n = sum(a,b)
  int d = sum(c,d)
 } sout(n/d)
```

(2) Naming Convention

```
for(i=0; i<arr.l; i++) {

}
for(idx=0; idx<arr.l; idx++){

}

String str = "Ethan"

String firstName = "Ethan"
```

Student
MathsDiagram
Class naming Convention

camelCase → methods / variable

ageV AgeX

(3) Comment code

→ doesn't mean comment every line ✗

→ Comment logic

(4) Brute force vs optimized

nge SL → Stack

n.1 ⇒ n^2 → Time | 1 → Space

n2 ⇒ n → Time ✓

n → Space

S.1 → Read Problem & understand

S.2 → Basis of understand → dry run I/P O/P

S.3 → Think of Soln → logic only

↓

Brute force

↓

Sacrifice space to optimize it.

① Greedy algo → not very sacrifice

$T(n) \propto \frac{1}{S(n)}$

Amazon

→ 3 → coding Qns → 2nd

→ 80% → mcqs

1. Pseudo code → Code → O/P
 a
 b
 c
 d

2. Core subject
 OOPS / DBMS / OS / CN (Basic)
 important

3. General Aptitude → very Basic

String firstName = "Sam"
~~int age = 50~~ Incorrect

Sort (firstName)

400 - 500] → approx

300 → 3 coders & 60%

100

Amazon dream