1. Pre-Code Planning (Before We Start Coding)

What it means

Pre-code planning is **thinking before coding**.
Before writing any program, you need to understand:

- What the **problem** is,
- What **inputs** are given,
- What **output** is expected, and
- The **steps** to solve it.

Real-Life Example

Imagine you're writing a recipe (like "How to make tea").
You don't just start pouring water — you plan first:

1. **Get ingredients (water, milk, tea leaves, sugar).**
2. **Boil water.**
3. **Add tea leaves and milk.**
4. **Pour into a cup.**

That's your **pre-code plan** before you "execute" the recipe — just like we do before writing code!

---

2. Pseudocode (The Blueprint of Program)

What is Pseudocode?

Pseudocode is **simple English-like steps** that describe how your program will work — without using any programming language syntax.

It helps you (and others) **understand the logic** before you write the real code.

Example 1 – Find the Sum of Two Numbers

Problem: Add two numbers entered by the user.

Pseudocode:

```
START
  READ number1
  READ number2
  sum ← number1 + number2
  PRINT "The Sum is", sum
END
```

Explanation:

- We first take input from the user (READ).
- Add them (sum ← number1 + number2).
- Then show the result (PRINT).

This can later be turned into **Python, Java, or C++** easily.

Verify the Algorithm (Check if it Works)

After writing **pseudocode**, **verify** it — test your logic with a few examples to make sure it works correctly.

Example Verification:

Let's verify the above "sum" **pseudocode**.

Test Case 1:
Input → number1 = 4, number2 = 5
Steps → sum = 4 + 5 = 9
Output → "The Sum is 9"

Test Case 2:
Input → number1 = 0, number2 = 7
Steps → sum = 0 + 7 = 7
Output → "The Sum is 7"

Test Case 3 (Edge Case):
Input → number1 = -3, number2 = 3

Steps → sum = -3 + 3 = 0
Output → “The Sum is 0”

If the algorithm gives correct results for different cases — it’s verified.

---

4. Let’s Try a Slightly Bigger Example — Find the Largest Number

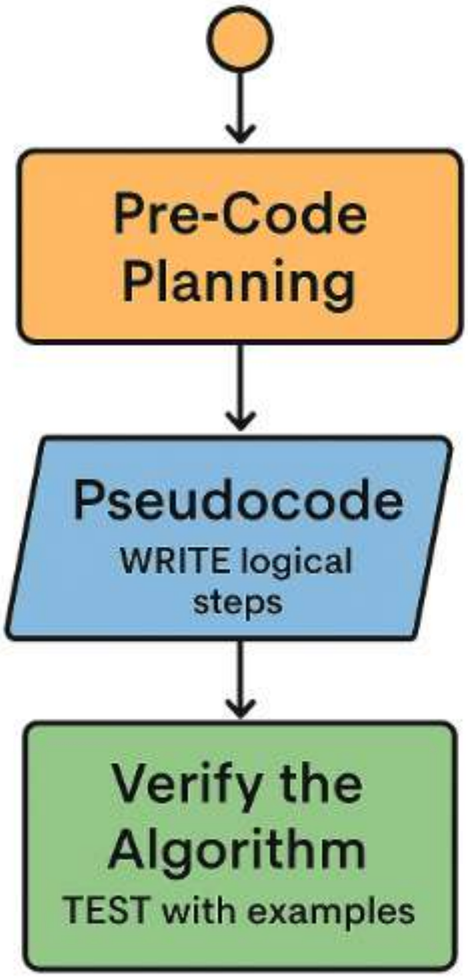
Pseudocode:

```
START
  READ num1, num2, num3
  IF num1 > num2 AND num1 > num3 THEN
      PRINT "num1 is the largest"
  ELSE IF num2 > num3 THEN
      PRINT "num2 is the largest"
  ELSE
      PRINT "num3 is the largest"
  ENDIF
END
```

Verification:

Test Case 1: num1 = 10, num2 = 7, num3 = 5 → Output: “num1 is the largest”
Test Case 2: num1 = 4, num2 = 9, num3 = 6 → Output: “num2 is the largest”
Test Case 3: num1 = 1, num2 = 2, num3 = 3 → Output: “num3 is the largest”

Works for all cases → algorithm verified!



5. Summary

Step	What You Do	Why It's Important
Pre-code Planning	Understand the problem and inputs/outputs	Helps you plan before writing code
Pseudocode	Write logical steps in simple English	Makes logic clear and language-independent
Verification	Test your pseudocode with real examples	Ensures correctness before coding

Pro Tip for New Programmers

“Code is 90% thinking and 10% typing.”
Spend more time planning and verifying your logic — it’ll make coding much easier!