Module 1c: Introduction to Problem Solving and Python Fundamentals

Premanand S

Assistant Professor,
School of Electronics and Engineering,
Vellore Institute of Technology, Chennai

premanand.s@vit.ac.in

July 19, 2025

Terminologies to brush up,

- Problem
- Programmers
- User
- Software

What is PROBLEM? (Programming)

- A problem is a specific challenge or task that needs to be solved through code. It could be anything from automating a task, fixing a bug, or developing a new feature.

Why do we need to solve the PROBLEM?

- Create the solution
- Improve the efficiency
- Enhance user experience
- Learning growth
- Drive business success
- Personal satisfaction

Problem solving in some domains,

- Engineering - Designing, testing, refining
- Medicine - Diagnosis illness, treatment, patient care
- Business - Strategic planning, managing operations, improving customer satisfaction, and driving innovation
- Information Technology - software, hardware, and networks
- Education - Curriculum, classroom and student management
- Everyday life - decision making, society

Why machines in problem-solving?

- Computer technology, completing any task quickly and accurately with skills
- Technology – not advanced enough to solve problems on own, hence we need to give step-by-step instruction to proceed
- Eg: Zomato, Swiggy – Ordering our favorite, Bookmyshow – Booking cinema in no time, Amazon, Netflix, Hotstar – Entertainment all in one place.
- Eg: Groceries in Netflix

Anti Virus - Scenario

- Problem - A user's computer is frequently getting infected with viruses and malware from the internet. This causes slow performance, data corruption, and security threats.
- User - The user is frustrated and wants a solution. But the user doesn't know how to remove or prevent viruses — they just want the computer to be safe.
- Programmer - They understand the problem and start analyzing:
  - What kinds of viruses are common?
  - How do they enter the system?
  - What patterns can identify them?
  - What should be done to remove or quarantine them?
- Software includes,
  - A virus scanner
  - A real-time protection system
  - A quarantine and deletion mechanism

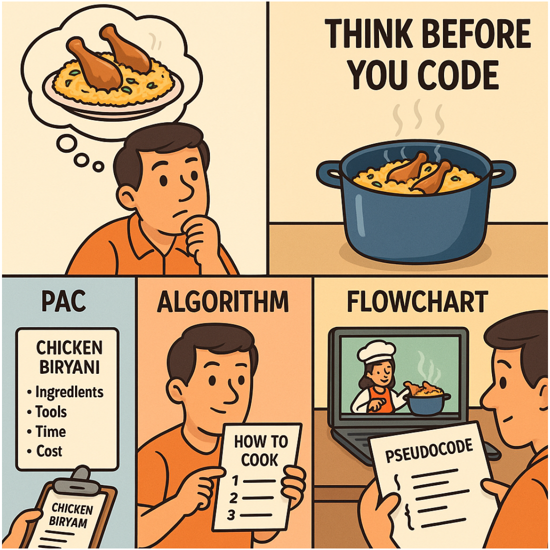
Topics to be covered in Pre-Programming Phase,

- Problem Solving: Definition and Steps
- **Problem Analysis Chart (PAC)**, Interactivity Chart (IC), Input Process Output (IPO) chart, Coupling Diagram, and Data Dictionary
- Developing an Algorithm
- Flowchart
- Pseudo-code

Program Development Cycle

- Analyze – Problem definition
- Design – Problem solution
- Choose recipes – Flowchart, Pseudocode, Charts, and Algorithms
- Code – Converting algorithm to programming language
- Test & Debug – Real-time errors
- Documentation – Problem-solving to product development

Thinking Before You Code



4-Step Pre-Code Process

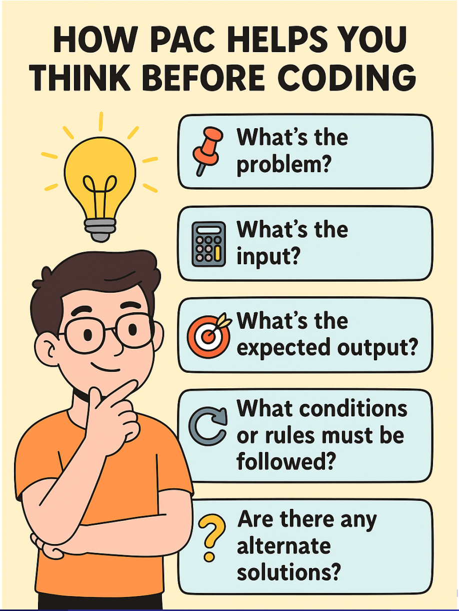
- Problem Analysis Chart (PAC)
- Algorithm
- Flow Chart
- Pseudocode

Problem Analyzing Chart (PAC)

Problem Analyzing Chart (PAC)

- A Problem Analysis Chart (PAC) is a tool used to break down and analyze a problem systematically.
- It helps in organizing thoughts and data related to the problem in a structured format.
- The primary purpose of a PAC is to gain a clear understanding of the problem and its components, which is crucial for developing effective solutions.

PAC – Problem Analysis Chart



HOW PAC HELPS YOU THINK BEFORE CODING

- What's the problem?
- What's the input?
- What's the expected output?
- What conditions or rules must be followed?
- Are there any alternate solutions?

PAC Analogy: Building a House

What type of house is needed? **Real-world House Analogy**	Problem Definition **PAC Component**	Understand the task or goal clearly. **Purpose**
What materials will be used?	Input	List the required data, tools, or resources.
What should it look like?	Expected Output	Describe the final outcome or result expected.
What's the design constraint?	Process / Rules	Specify logic, formulas, or restrictions to follow.
Any better way to do it?	Alternate Solutions	Explore other efficient or logical approaches.

PAC Example – Feeling Hungry Scenario

PAC Component	**Details / Questions**
Problem Statement	Feeling hungry and want to eat something.
Input	Current time, available food, dietary preferences, hunger level, time available.
Process	1. Assess how hungry you are (snack or full meal). 2. Check available food. 3. Consider dietary needs. 4. Decide meal based on time.
Output	Decision on what to eat and possibly where/how to get it.
Alternative Solutions	1. Cook a quick snack. 2. Order food online. 3. Eat a ready-to-eat meal. 4. Wait until next mealtime.

PAC – Pedestrian Signal Control System

PAC Component	**Details / Questions**
Problem Statement	Automatically control a pedestrian signal at a crosswalk to ensure safe crossing.
Input	Pedestrian button press (Yes/No), current traffic light state (Red/Green), time of day.
Process	1. Detect if the pedestrian has pressed the button. 2. Check current traffic light. 3. Wait for a safe time gap or switch traffic light to red. 4. Show "WALK" sign for a few seconds. 5. Switch back to "DON'T WALK" after timer ends.
Output	Display: "WALK" or "DON'T WALK" to pedestrians, and update traffic signal if needed.
Alternative Solutions	1. Manual control via traffic police. 2. Sensor-based detection (instead of button). 3. Time-based cycles without input.

PAC – Even or Odd Number Check

PAC Component	**Details / Questions**
Problem Statement	Determine whether a given number is even or odd.
Input	A single integer (e.g., 4, 15, -8, etc.).
Process	1. Take the number as input. 2. Check if the number % 2 equals 0. 3. Decide based on the result.
Output	A message stating: “Even” or “Odd”.
Alternative Solutions	1. Use % (modulus) operator. 2. Use bitwise AND (number & 1). 3. Use integer division logic: `(n // 2) * 2 == n`.

PAC – Find the Largest of Three Numbers

PAC Component	**Details / Questions**
Problem Statement	Determine the largest among three given numbers.
Input	Three numbers (e.g., a, b, c).
Process	1. Compare `a` with `b`. 2. Compare the larger of `a` and `b` with `c`. 3. Store/display the largest value.
Output	The largest number among the three.
Alternative Solutions	1. Use nested `if/else` conditions. 2. Use built-in `max(a, b, c)` function (Python). 3. Sort the numbers and pick the last one: `sorted([a, b, c])[-1]`.

PAC – User Login Authentication System

PAC Component	Details / Questions
Problem Statement	Build a system that checks if a user-entered username and password match stored credentials.
Input	Username and password entered by the user.
Process	1. Prompt user for input (username and password). 2. Check if the username exists in the database. 3. If yes, compare the entered password with stored password (or hashed version). 4. If matched → Login successful. 5. If not → Display error and limit login attempts.
Output	Display success message or grant access if credentials match; otherwise, show error message.
Alternative Solutions	1. Add two-factor authentication (2FA). 2. Use OTP instead of password. 3. Use biometric authentication (e.g., fingerprint or face recognition).

Why PAC Matters

- Pass, Think and Plan
- Clarity

Assignment

1. What is a Problem Analysis Chart (PAC)? Why is it useful before writing code?
2. List and explain the main components of a PAC.
3. How does PAC help in breaking down complex problems?
4. What is the difference between "Input" and "Process" in a PAC? Give an example.
5. How do "Alternate Solutions" in PAC encourage better problem solving?

Assignment

For each of the following problems, fill in the PAC table:

1. ATM cash withdrawal
2. Online exam login and submission system
3. Elevator control system
4. Smart irrigation system that waters plants when soil is dry
5. A system to detect whether a student is eligible for a scholarship

Assignment

6. Think of a daily life problem not discussed in class. Describe it and complete a PAC chart.
7. Take a simple Python program you wrote. Reconstruct the PAC for it.
8. Why is the "Alternate Solutions" section important? Give an example.
9. Which of the following is NOT a PAC component? a) Input b) Process c) Loop d) Output
10. In the PAC, the "Output" represents: a) Tools used b) Final result c) Logic d) Errors

mail me: er.anandprem@gmail.com / premanand.s@vit.ac.in
ring me: +91 73586 79961
Follow me: Linkedin
Medium Blogs
Analytics Vidhya: Blogs

Don't just code — think, plan, and solve