Ch-4 Introduction to Problem-Solving in Programming

What is Problem-Solving?

Problem-solving is a **step-by-step process** of understanding and finding a solution to a task using logical reasoning and programming.

Steps in Problem-Solving

Step	Description
1. Analyze the Problem	Understand the input, expected output, and rules. Break the problem into parts.
2. Develop an Algorithm	Write clear steps to solve the problem logically.
3. Coding	Translate the algorithm into a programming language (e.g., Python).
4. Testing	Check the code using different inputs to see if it gives correct output.
5. Debugging	Fix any errors (bugs) in the code (syntax or logic errors).

What is an Algorithm?

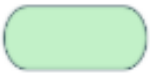
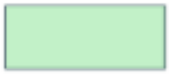
An **algorithm** is a **finite set of well-defined steps** to solve a problem.

Example: Algorithm to find the sum of two numbers

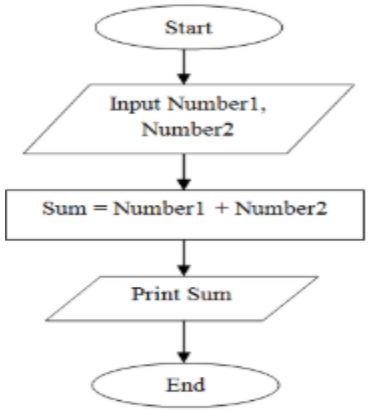
1. Start
2. Input two numbers A and B
3. Calculate Sum = A + B
4. Display Sum
5. End

Flowchart Representation

Flowchart Symbols:

Symbol	Name	Function
	Oval	Represents the start or end of a process
	Rectangle	Denotes a process or operation step
	Arrow	Indicates the flow between steps
	Diamond	Signifies a point requiring a yes/no
	Parallelogram	Used for input or output operations

Example: Sum of Two Numbers



Pseudocode Representation

Pseudocode = **English-like instructions** that resemble programming

Example:

START
INPUT A, B
SUM ← A + B
OUTPUT SUM
END

Decomposition

What is Decomposition?

Breaking a complex problem into **smaller sub-problems** or modules.

Benefits:

- Easier to understand and solve
- Allows team-based coding
- Promotes reuse of logic

Example:

Problem: Create a Student Report
Decomposition:

- Input student details
- Calculate total and average
- Determine grade
- Display report

Summary Table

Concept	Description
Problem-Solving	Stepwise approach to solve tasks
Algorithm	Sequence of steps
Flowchart	Visual representation
Pseudocode	English-like instructions
Decomposition	Divide large problems into sub-tasks