Chapter 1

Introduction to Problem Solving

Problem Solving

A way of thinking for logically and methodically solving problems – E.g., purposeful, describable, replicable .Problem solving includes skills such as:
– Decomposition
– Pattern Recognition
–Abstraction
– Generalization
– Algorithm Design
– Evaluation

Decomposition - Breaking down a process into a set of smaller sub-processes to allow us to describe, understand, or execute the process better – Dividing a task into a sequence of subtasks – Identifying elements or parts of a complex system.

Example of Decomposition

• When we taste an unfamiliar dish and identify several ingredients based on the flavor, we are decomposing that dish into its individual ingredients
• When we give someone directions to our house, we are decomposing the process of getting from one place to another (e.g., city, interstate, etc.)
• When we break an experiment in chemistry into multiple smaller experiments
• When we design a solution to construct a bridge by considering site conditions, technology available, technical capability of the contractor, foundation, etc.

Pattern Recognition

Noticing or identifying similarities or common differences that will help us make predictions or lead us to shortcuts – We look for patterns when we play games to decide when to do certain things – Based on experience, we develop shortcuts mapping problem characteristics to solution.

Examples of Pattern Recognition

• We look for patterns when choosing a registrar when we checkout

• Drivers look for patterns in traffic to decide whether and when to switch lanes
• People look for patterns in stock prices to decide when to buy and sell
• Scientists and engineers look for patterns in data to derive theories and models
• We look for patterns and learn from them to avoid repeating the same mistake

Abstraction

Preserving information that is relevant in a context, and forgetting or suppressing information that is irrelevant in that context to solve a problem
– We use abstraction to organize things:
• A human is a mammal, a mammal is an animal, and so on
– A "big picture" so we can reason without thinking about the details

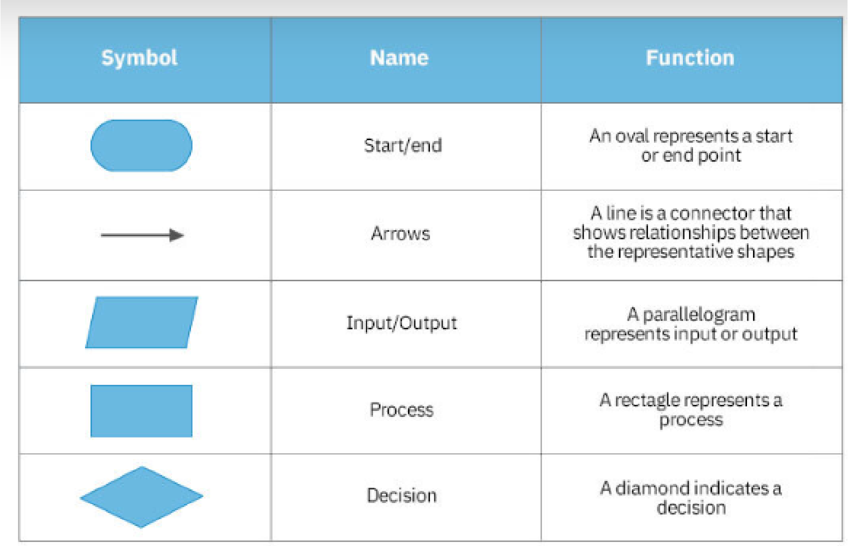
To create efficient and effective ptrogrammer, you should adopt a problem-solving methodology. There are four basic steps in the problem-solving cycle.

- ⇨ Identify and analyse the problem.
- ⇨ Find its solution and develop algorithm of the solution.
- ⇨ Code the solution in a programming language. Test and Debug of the coded solution.
- ⇨ Test and debug the coded solution.

Algorithm – An algoritm is a set of instructions, used to solve problems or perform tasks, based on the understanding of available alternatives.

Pseudocode - It is an artificial and informal language that helps programmers develop algorithms. Pseudocode is a "text-based" detail (algorithmic) design tool.

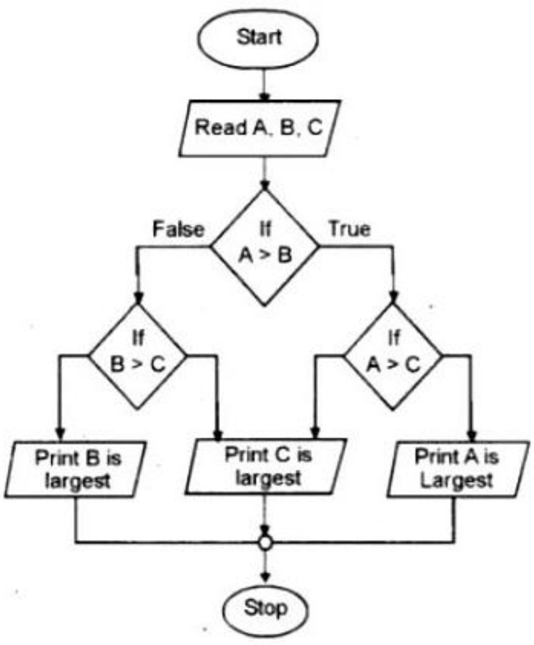
Flowchart - A flowchart is **a diagram that depicts a process, system or computer algorithm**. They are widely used in multiple fields to document, study, plan, improve and communicate often complex processes in clear, easy-to-understand diagrams.



Symbol	Name	Function
⬭	Start/end	An oval represents a start or end point
→	Arrows	A line is a connector that shows relationships between the representative shapes
▱	Input/Output	A parallelogram represents input or output
▭	Process	A rectangle represents a process
◆	Decision	A diamond indicates a decision

Shape used in flowchart

Flow chart of largest of three numbers.



Algorithm to find greatest number of three given numbers:

1. Ask the user to enter three integer values.
2. Read the three integer values in A,B and C (integer variables).
3. Check if A is greater than B.
4. If true, then check if A is greater than C.
    1. If true, then print ‘A’ as the greatest number.
    2. If false, then print ‘C’ as the greatest number.
5. If false, then check if B is greater than C.
    1. If true, then print ‘B’ as the greatest number.
    2. If false, then print ‘C’ as the greatest number.

Exercise

1. Write a flowchart and algorithm to accept marks of 30 students . Calculate the sum and average marks.
2. Define decomposition and abstraction.
3. Define debugging.
4. What is pseudo code?
5. Write the steps of problem-solving cycle.