

# Problem Solving and Computational Thinking

# **“Problem solving is a skill that can be developed via practice”**

- Define the Problem
  - What exactly is the problem that we are trying to solve?
- Identify the Problem
  - How and why did the problem happen?
- What are all the possible solutions?
  - The ideal solution could be one of the many possible solutions.
- A decision is to be made.
  - Any decision is usually better than no decision at all.

- Assign responsibility to carry out the decision.
  - If a team then who will do what and when.
  - If alone, still decide when are you going to do it
- Set a schedule.
  - Without schedule and deadline, its just a discussion.
- Task self/someone else to take definite action to implement the solution and resolve the problem.

# Core Components of Computational Thinking

- Decomposition
  - Break down complex problems into smaller, simpler problems.
- Pattern recognition
  - Make connections between similar problems and experience.
- Abstraction
  - Identify important information while ignoring unrelated or irrelevant details.
- Algorithms
  - Creates sequential rules to follow in order to solve a problem.

# Algorithm and Data Structures

# Algorithm

- A “**finite sequence**” of “**well defined**” computational steps that transforms “**input**” into “**output**”.
- Basic constructs of an algorithm.
  - Linear Sequence – statements that follow one after the other.
  - Conditional – “if then else”
  - Loop – sequence of statements that are repeated a number of times.

# Data Structure

- A *data structure* is a way to **store** and **organize** data in order to facilitate **access** and **modifications**.
- No single data structure works well for all purposes.  
So, it is important to know the strengths and limitations of several of them.

# Linear Data Structures

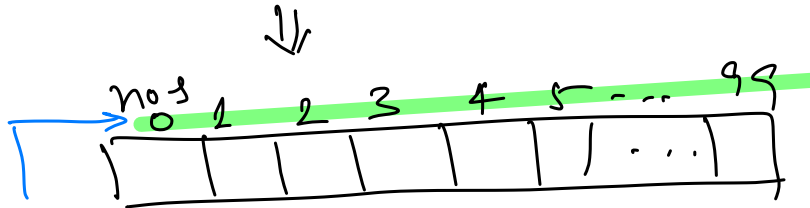
## Array

- Need for an array?

↳ When same processing is to be done on a set of data.

Sum of 100 numbers.

int nos[100];



Sum of 3  
numbers

int no1, no2,  
no3;

no1    no2  
3    4


no3  
5

int sum;

sum    sum = no1 +  
712    no2;  
↖ ↗  
sum = sum + no3;

1  
index/subscript  $\Rightarrow$  unique number given to identify elements of array.

int sum;

sum = 0; 

for i = 0 to 99

sum = sum + nos[i];

} Pseudo code

$\Downarrow$  in JAVA

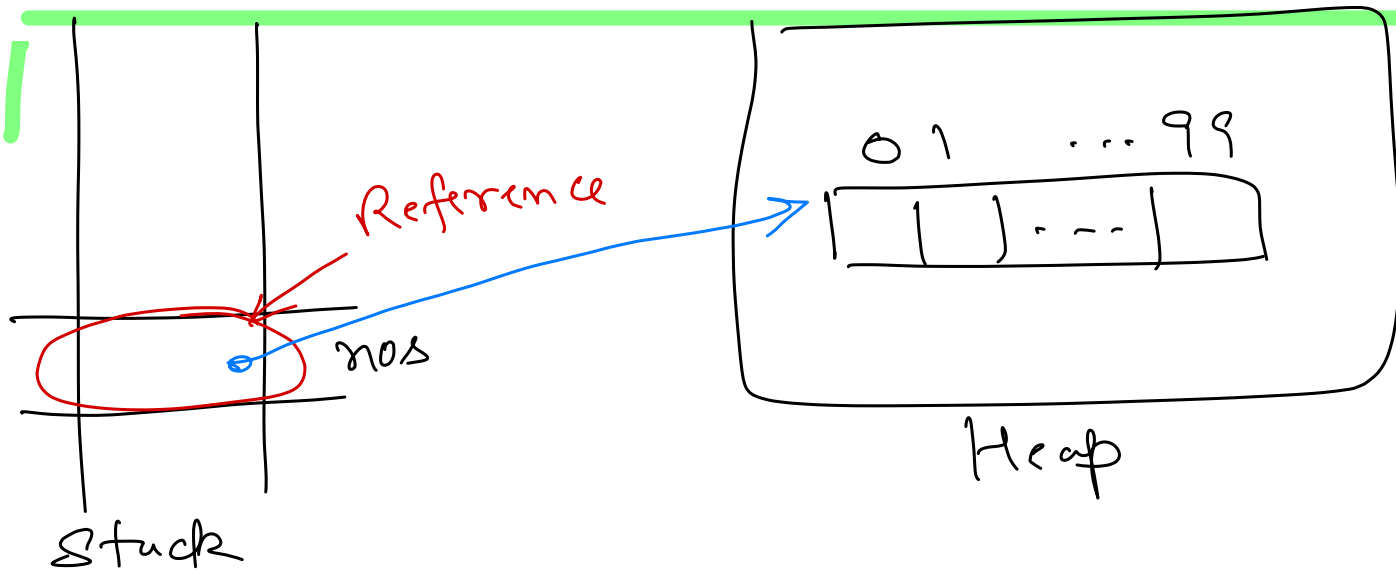
int[] nos = new int[100];

int sum = 0;

for (int i = 0; i < 100; ++i)

sum = sum + nos[i];





## Properties of Array

- Data Structure that stores multiple elements, all of the same type.
- All elements of an array are stored sequentially in memory, one after another.

# Pros and Cons of Array

- Advantages
  - Efficient lookup OR Random access.
  - Efficient in adding and removing elements at the end of array
- Disadvantages
  - Fixed size. Resizing of array is inefficient.
  - Insertion and deletion of elements, in middle of array is inefficient.