Logic Formulation

Steps in Problem Solving

1. **Define the problem**
2. **Design the algorithm to solve the problem**

Program Logic Formulation

- Usually defined as the process of coming up with the appropriate methodology in developing a specific program logic that will perform a prescribed computing task or solve a problem using the computer.
- Logic is the systematic study of reasoning that provides standards by which valid reasoning can be categorized.

Algorithm

- It is a list of instructions specifying a precise description of a step by step process that terminates after a finite number of steps for solving an algorithm problem producing the correct answer in the end.
- It is the recipe for solving problems.
- A finite set of an instruction that specifies a sequence of operation to be carried out in order to solve specific problem.
- An ambiguous procedure specifying a finite number of steps to be taken.

Properties of Algorithm

- **Finiteness** – there is an exact number of steps to be taken and has an end.
- **Absence of Ambiguity** – means that every instruction is precisely described and clearly specified.
- **Sequence of Execution** – instructions are performed from top to bottom.
- **Input and Output** – defined the unknowns of the problem is specified and with the expected outcome.
- **Effectiveness** – the solution prescribed is guaranteed to give a correct answer and that the specified process is faithfully carried out.
- **Scope definition** – applies to a specific problem or class of problem.

Algorithm can be expressed as:

- **Pseudocode**
- **Structure Diagram**
- **Flowchart**

Pseudocode

- a compact and informal high–level description of a computer programming algorithm that uses the structural conventions of a programming language, but is intended for human reading rather than machine reading
- a short hand way of describing a computer program rather than use the specific syntax of a computer language
- English–like statements that are used for low–level program design

*Example: Write a program calculating the sum of two numbers

Pseudocode
Version 1

```
PROGRAM Add Two Numbers
  READ two numbers
  ADD the numbers
  WRITE the sum
END PROGRAM
```

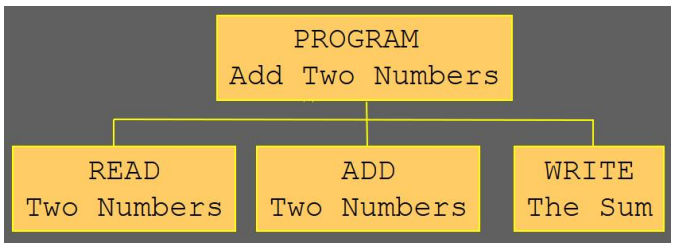
Pseudocode
Version 2

```
PROGRAM Add Two Numbers
  READ First
  READ Second
  Sum = First + Second
  WRITE Sum
END PROGRAM
```

Structure Diagram

- A structure diagram is a conceptual modeling tool used to document the different structures that make up a system such as a database or an application.
- It shows the hierarchy or structure of the different components or modules of the system and shows how they connect and interact with each other.
- It is a tool used to guide developers to ensure that all parts of the system work as intended in relation to all the other parts.

*Example: Write a program calculating the sum of two numbers



- **Rules for Structure Diagram**
  - ✓ A module which resides above others is referred to as a Calling module
  - ✓ A module which resides below another is referred to as a Called module
  - ✓ A module can be both a calling and called module
  - ✓ A called module can only be called by one calling module