**Data Structure**

**&**

**Algorithm**

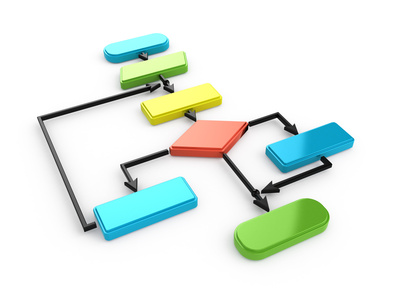
**Class IX**

**Lab 8**

|  |
| --- |
| Lab Objectives:Algorithm Basics |

# What is Algorithm?

## Algorithm is a step-by-step procedure, which defines a set of instructions to be executed in a certain order to get the desired output.



an algorithm can be implemented in more than one programming language. From the data structure point of view, following are some important categories of algorithms −

## ***Search***− Algorithm to search an item in a data structure.

## ***Sort***− Algorithm to sort items in a certain order.

## ***Insert***− Algorithm to insert item in a data structure.

## ***Update***− Algorithm to update an existing item in a data structure.

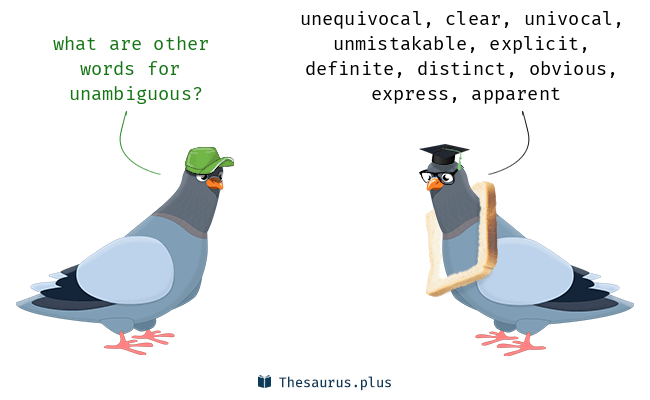
## ***Delete***− Algorithm to delete an existing item from a data structure.

* 

# Characteristics of an Algorithm

## Not all procedures can be called an algorithm. An algorithm should have the following characteristics −

## ***Unambiguous***− Algorithm should be clear and unambiguous. Each of its steps (or phases), and their inputs/outputs should be clear and must lead to only one meaning.



## IMG_256***Input***− An algorithm should have 0 or more well-defined inputs.

## IMG_256***Output***− An algorithm should have 1 or more well-defined outputs, and should match the desired output.

## ***Finiteness***− Algorithms must terminate after a finite number of steps.

## ***Feasibility***− Should be feasible with the available resources.

## ***Independent***− An algorithm should have step-by-step directions, which should be independent of any programming code.

# How to Write an Algorithm?

## There are no well-defined standards for writing algorithms. Rather, it is problem and resource dependent. Algorithms are never written to support a particular programming code.

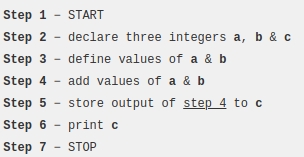
## As we know that all programming languages share basic code constructs like loops (do, for, while), flow-control (if-else), etc. These common constructs can be used to write an algorithm.

## We write algorithms in a step-by-step manner, but it is not always the case. Algorithm writing is a process and is executed after the problem domain is well-defined. That is, we should know the problem domain, for which we are designing a solution.

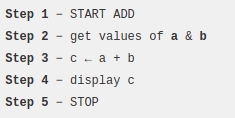
# Example

## Let's try to learn algorithm-writing by using an example.

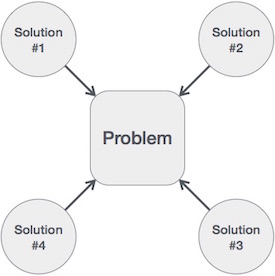
## **Problem**− Design an algorithm to add two numbers and display the result.



## Algorithms tell the programmers how to code the program. Alternatively, the algorithm can be written as −



## We design an algorithm to get a solution of a given problem. A problem can be solved in more than one ways.



## Algorithm helps us to choose the best solutions among them.

## For choosing the best solutions algorithm provides many searching and sorting techniques. We will discuss it further.