# UNIT: 1
# Flowchart and Algorithm

## ❖ Flowchart
### Definition and Importance of flowchart:

**Flowchart:** Flowchart is Graphical or Pictorial Representation of problem.

## ❖ Importance (Advantages) of Flowchart:

1. Flowcharts are better way of communicating the logic of the system.
2. Using Flowchart it is easy to analyze Problem.
3. The flowcharts are very useful during program development phase.
4. Flowchart represents problem solution using different standard Symbol.
5. It provides a good communication interface between programmer and user.
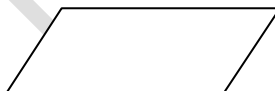6. It helps in debugging and testing process.

## ❖ Symbols of flowchart:

### 1. Terminal:
   o This Symbol is used for the Starting and Ending Process of Flowchart.
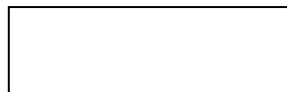
### 2. Input / Output
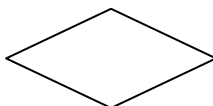   o This Symbol is used for Receiving input Data From user and displaying output to the user

### 3. Process
   o This Symbol is used for Computing.
   o It is Use for Some Arithmetic Calculation.

### 4. Decision Making
   o This Symbol is used for making decision base on Condition.
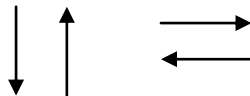   o This Symbol has one incoming arrow for entry and Two outgoing arrow.

**5.Subroutine**
  o  This Symbol is used for user defined function or sub program.

**6.Flow / Direction**
  o  Flow line Symbol.
  o  This Symbol Indicates flow of data from one Symbol to Another Symbol.

**7.Page Break**
  o  When flowchart is do not fit into single page then Page Break and Page Continue Symbol is used.
  o  Page Break Symbol Indicate break in current page.

**8.Page Continue**
  o  Page Continue Symbol Indicate Continue flowchart in another page.

## ❖ Guidelines for preparing Flowchart:

1. Standard symbols should be used while drawing flowchart.
2. Ensure that flowchart has START (or BEGIN) and STOP (or END).
3. Flowchart should be neat, clean and easy to follow. There should be no any ambiguity.
4. The usual direction of flowchart is from top to bottom or from left to right.
5. The terminal symbol, that is, START/BEGIN or STOP/END should have only one flow line.
6. Only one flow line should come out from process symbol.
7. Only one flow line should enter a decision symbol.

## ❖ Flowchart structure: Sequence, selection, repetition:

1. Sequence
   Steps that execute in sequence are represented by symbols that follow each other top to bottom or left to right. Top to bottom is the standard.
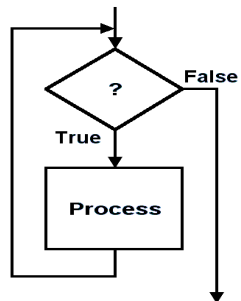
```
┌──────────────┐
│  Process 1   │
└──────────────┘
       │
       ▼
┌──────────────┐
│  Process 2   │
└──────────────┘
       │
       ▼
```
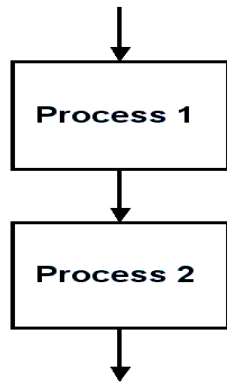
2. Selection
   Once the condition is evaluated, the control flows into one of two paths.

```
        ◇ ? ──── No
       Yes         │
        │          ▼
   ┌─────────┐  ┌─────────┐
   │Process 1│  │Process 2│
   └─────────┘  └─────────┘
```
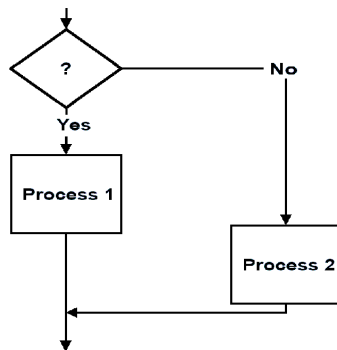
3. Repetition
   Either the processing repeats or the control leaves the structure.

```
       ◇ ? ──── False
      True        │
       │          ▼
   ┌─────────┐
   │ Process │
   └─────────┘
```

## ❖ Limitation of flowchart:

1. Complex Logic: It is very difficult to draw a flowchart for very hard and complex problem
2. Difficulty in Modifications: If a minor change is required then it is difficult to change flowchart and may be required to redraw flowchart again.
3. Difficult to understand for people who don't know flowchart symbols.

## ❖ Definition and Importance of Algorithm:

**Algorithm:** Algorithm is stepwise solution of any problem or program.

### Importance (advantages) of Algorithm:

1.  It is a step-wise representation of a solution to a given problem, which makes it easy to understand.
2.  An algorithm uses a definite procedure.
3.  It is not dependent on any programming language, so it is easy to understand for anyone even without programming knowledge.
4.  Every step in an algorithm has its own logical sequence so it is easy to debug.
5. By using algorithm, the problem is broken down into smaller pieces or steps, so it is easier for programmer to convert it into an actual program.

## Limitation of Algorithm

1.  An algorithm is Time consuming. Developing algorithm for complex problems would be time consuming and difficult to understand.
2.  Understanding complex logic through algorithms can be very difficult.
3.  Difficult to show Branching and Looping in Algorithms.
4.  Big tasks are difficult to put in Algorithms.

**Examples:**
**Flowchart and algorithm to calculate area of circle.**
**Flowchart:**



Fig. Flowchart to print Area of Circle
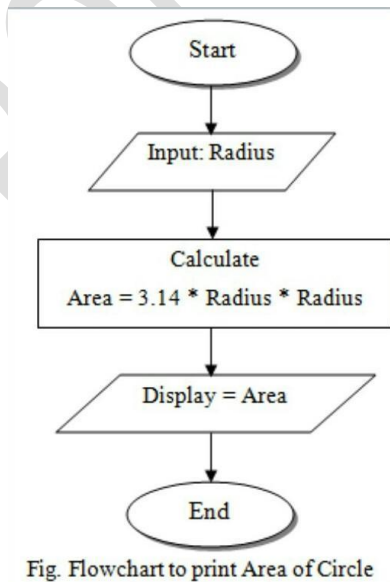
### Algorithm:

1. START
2. INPUT: r
3. Calculate: Area= 3.14 * r *r
4. PRINT:  Area
5. STOP

## ❖ Developing and writing algorithm using pseudo codes:

**Pseudo code:**

**Pseudo code is not a programming language**; it is a simple way of describing a set of instructions that does not have to use specific syntax.

Example:

If student's grade is greater than or equal to 60

Print "passed"

else

Print "failed"