Introduction to Problem Solving

Algorithm: **An Algorithm is a systematic collection of steps which used to solve any problem. These steps can be implement in programming language for solution. The algorithm should be efficient and effective in nature.**

Efficiency Vs Effectiveness

Effectiveness mean that the algorithm should carry out its correct desired output.

Efficiency mean that the algorithm should be correct with the best possible performance for all types of inputs.

Factors affect the Algorithm:

The set of rules that define how a particular problem can be solved in finite number of steps is known as algorithm. The quality or performance of an algorithm is depends on many internal and external factors. Two or more algorithms can be compared based on these factors.

1. **Internal factors:**
   - ♦ **Time required to run**
   - ♦ **Space (Memory) required to run**
2. **External factors:**
   - ♦ **Size of input to algorithm**
   - ♦ **Speed of computer**
   - ♦ **Quality of Compiler**

Characteristics of good Algorithm:

The effective algorithm should have the following characteristics.

3. **Precisely defined:** **That mean all the terms used in algorithm should be defined in well understandable way.**
4. **Uniqueness:** **every step of algorithm should be unique and it should contribute in problem solving.**
5. **Finiteness:** **the algorithm should not repeat the steps endlessly. It should run up to finite number of times.**
6. **Input:** **The algorithm should requires specific types of input to work upon it.**
7. **Output:** **An algorithm should produce the output as per expectations and it should wholly depend on the input.**

Example-1:

Write an algorithm for the problem- "find the larger number of given two number".

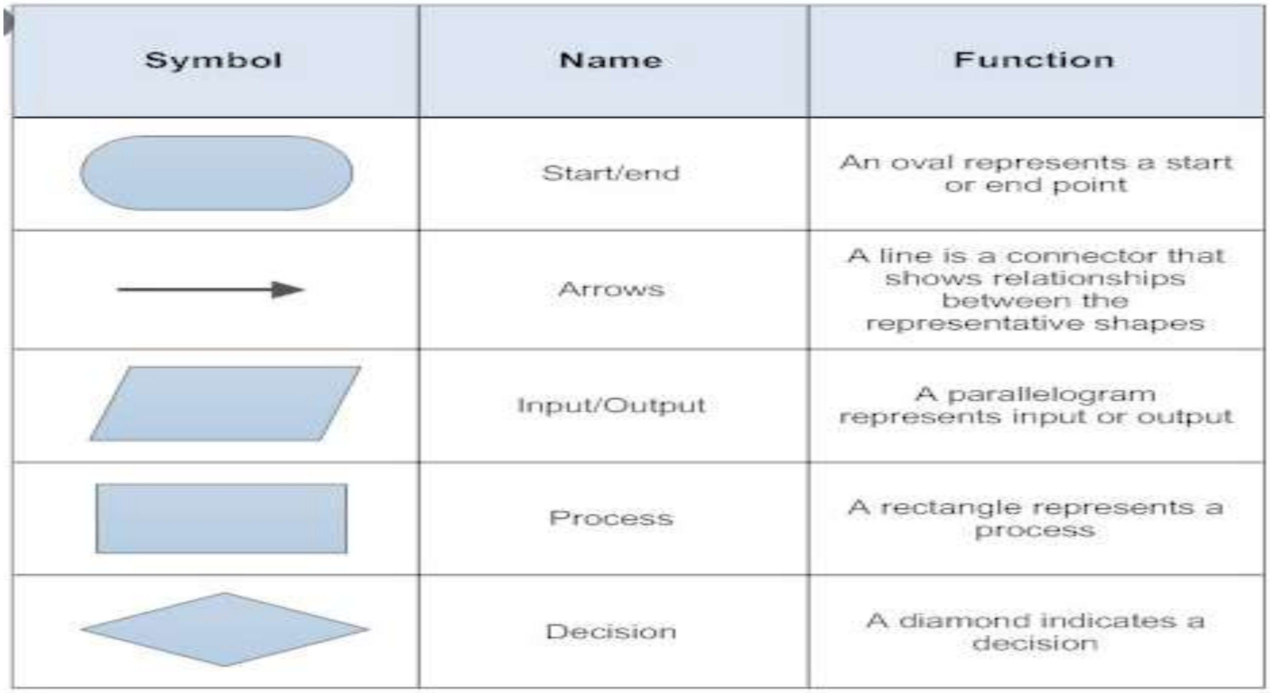
```
Step-1: Start

Step-2: Read two numbers X and Y

Step-3: If X > Y, then print X else Y.

Step-4: Stop
```

Example-2: **Write an algorithm for the problem- "find the sum of given 10 numbers".**

```
Step-1: Start

Step-2: Set Sum=0 and count=1

Step-3: if count<=10 then go to step 4 otherwise go to Step-8

Step-4: Read a number N

Step-5: Add N into Sum as Sum= Sum +N

Step-6: increase count by 1 as Count=count+1

Step-7: Go to Step-3 again.

Step-8: Print Sum

Step-9: Stop
```

Flow Chart:

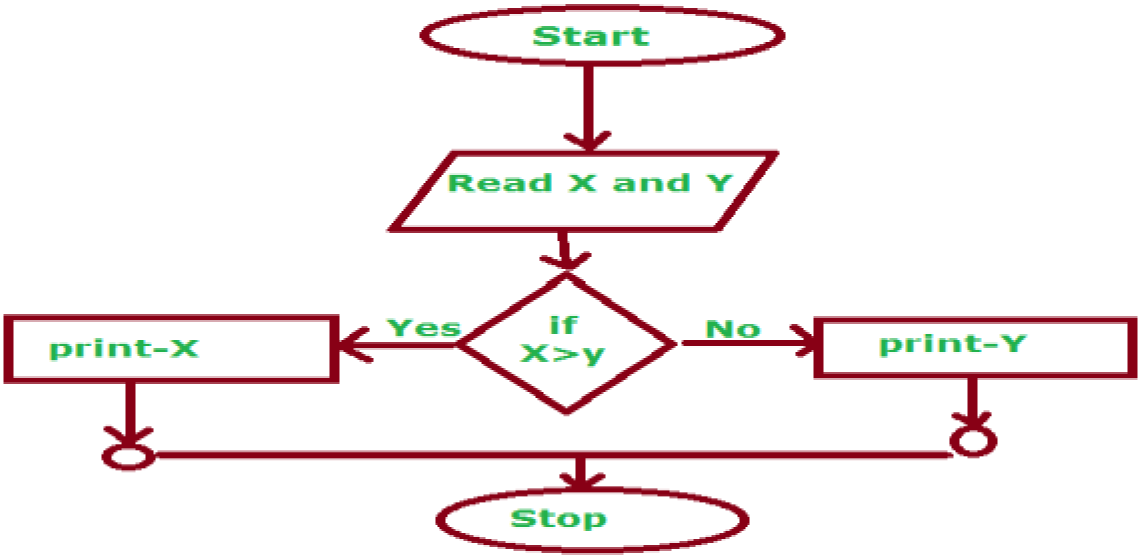
Flowchart is a graphical representation of various steps of problem solution. Flowchart contains some special symbols to represent the specific step of problem solution.



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

Example-1

Draw a flowchart for the algorithm “find the larger number of given two number”.



Pseudocode:

Pseudocode is an informal way to describe the various steps of a problem solution. It does not follow the programming language rules. The main objective of pseudocode is to give idea to understand the problem in easy way.

Advantages of Pseudocode:

- **It written in English like language to understand easily.**
- **It gives idea about the sequence of steps for solution of problem.**
- **Easy to edit and understand.**

Disadvantages of Pseudocode:

- **It is not pictorial representation like flowchart**
- **Pseudocode may be different for different users for same problem because there is no any strict rule to follow.**
- **It can written on paper only.**

Example-1

Write a pseudocode to find that you are eligible for casting the Votes or not eligible.

```
If candidate's age is greater than or equal to 18 year
    Display "Eligible for Voting"
Else
    Display "Not Eligible for Voting"
```

Example-1

Write a pseudocode to find that you are passed of failed in examination.

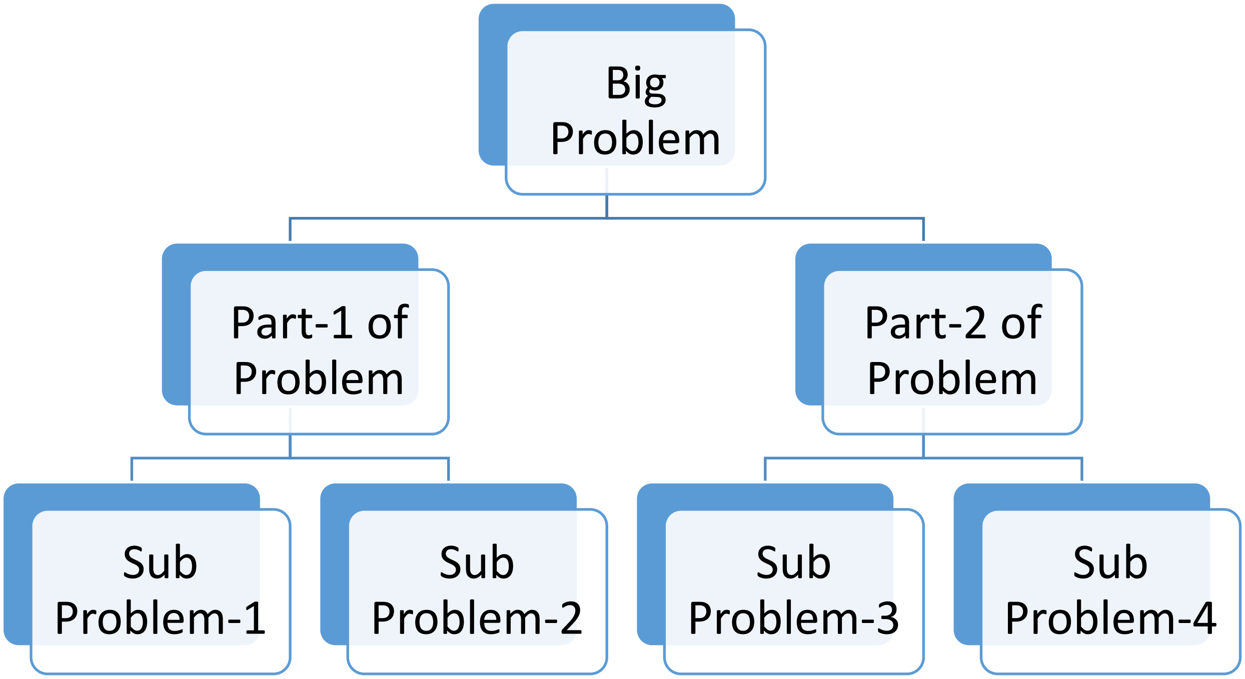
```
If student's percent is greater than or equal to 33
    Display "Passed"
Else
    Display "Failed"
```

Decomposition:

It is a process that break down a big or complex problem into smaller sub – processes. Its helps to understand a complex problem in better way. All it is known as decomposition.

Decomposition involves the following steps:

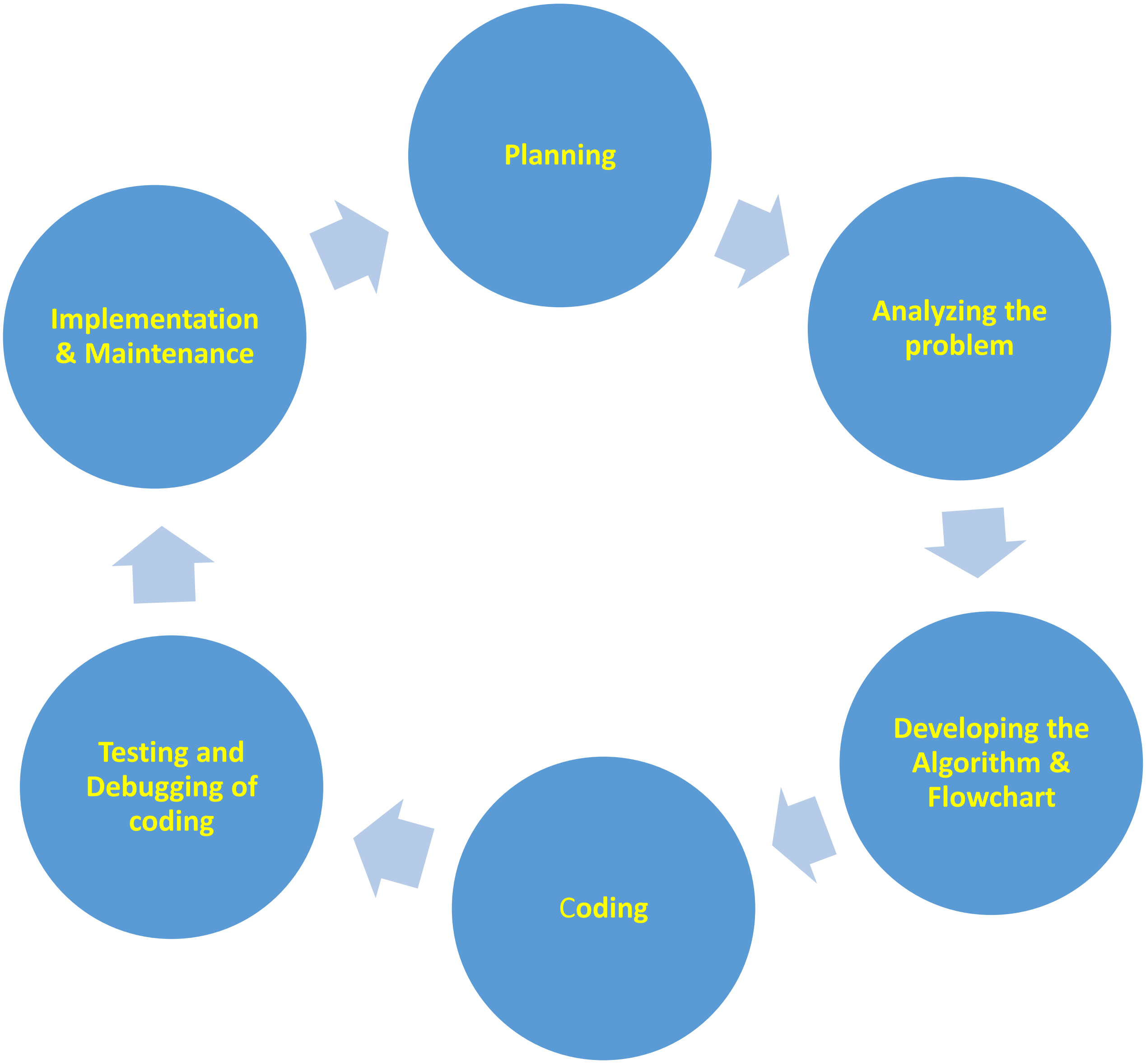
- **Dividing a big or complex task into smaller sub-tasks.**
- **Identify the various elements of complex system.**



Cycle of Problem Solving

The Problem solving cycle includes the following steps.

1. **Planning**
2. **Analyzing the problem**
3. **Developing the Algorithm & Flowchart**
4. **Coding**
5. **Testing and Debugging of coding**
6. **Implementation & Maintenance**


Planning
Analyzing the problem
Developing the Algorithm & Flowchart
Coding
Testing and Debugging of coding
Implementation & Maintenance

1. Planning

The purpose of the Problem solving Planning is to gather all information necessary to control the program solution. It is the phase where developer plan for the step to step solution of problem.

2. Analyzing the problem

In this phase, developer understand the problem very well and analyze the problem. Developer also identify the various components and relationship among components.

3. Developing the Algorithm & Flowchart

Under this phase,

- Developer think about various possible solutions. Finally select one of the best solution for problem.
- Draw the algorithm and flowchart for solution to understand it in easy way.
- Follow the modular approach for systematic solution of problem.
- Identify all required operations those will be useful in coding. Like, minimum inputs, type of input, type of output etc.

4. Coding

The developer always use any programming language to apply the algorithm that is selected as the best solution of problem. Coding should be as module to module.

While writing code for algorithm, following factors should be considered

- Program efficiency
- Program effectiveness
- Speed & performance of program

5. Testing and Debugging

This is the phase where developer traces the errors (bugs) in coding. The error occurred in code are known as bugs and to rectify the bugs is termed as debugging. These testing can be performed by putting various inputs.

6. Documentation

Documentation mean the detailed description about the terms used in coding. There are two types of documentation

- **Internal Documentation: The various comments used in program during coding are considered as internal documentation.**
- **External Documentation: The explanation of flowcharts, algorithm and about input – Output types of program are come under external documentation.**

7. Implementation & Maintenance

This is the last phase of problem solving. After completion of coding and preparation of documentation, developer has to apply the solution at actual site and make it available for all user to work on it.

Maintenance is the process that continues for several years after implementation of project. It includes the modification in existing solution and also adopt new features in project.

Verifying an Algorithm

Verification of an algorithm mean to test and ensure that it is working correctly for sample of inputs. Developer provides some inputs and ensures correctness of output. To verify a big algorithm the intermediate results should be tested.

Mechanism to verify an algorithm

1. **Dry run:**

   In the "dry run" approach, developer manually check the steps of algorithm by providing some inputs. Developer manually calculate the result of steps and compare with the result of algorithm. If any unexpected result found then developer change the code and again check it manually.

   Characteristics of Dry run:

   - **It is carried out during coding, testing, implementation or maintenance phase.**
   - **It is used to identify and rectify the logical errors in coding.**

****Finish****