**PRINCIPLES OF PROGRAM ANALYSIS**

This chapter starts with the basic information regarding the fundamental knowledge required to solve various problems. Algorithm design is one of the primary steps in solving problems. Algorithms are set of steps or instructions required and designed to solve a specific problem.

**SOFTWARE DEVELOPMENT LIFE CYCLE**

Developing good software is a tedious process which keeps on going i.e. under development for a long time before the software or the program takes the final shape. This process is often termed as Software Development Life Cycle (SDLC). Here, the output of one stage becomes the input of next stage.

The various steps involved in the Software Development Life Cycle are as follows:

* Analyze the problem with precision
* Create a prototype and experiment with it until all requirements are finalized
* Design an algorithm for the task using the tools of the data structure
* Verify the algorithmic steps
* Analyze the algorithm for checking its requirements
* Code the algorithm to any suitable programming language
* Test and evaluate the code
* Refine and repeat the preceding steps until the software is complete
* Optimize the code to improve performance
* Maintain the application that you have designed so that it meets the upcoming client's and users need

### Program Design

Program design is an important stage of software development. This phase takes the help of algorithms and different concepts of data structures to solve the problem(s) that is proposed.

### Different Approaches to Designing Algorithms

Complex code can be subdivided into smaller units called modules. The advantage of modularity is that it allows the principle of separation of concerns to be applied into two phases are -

* While dealing with details of each module in isolation
* While dealing with overall characteristics of all modules and their relationships

Modularity enhances design clarity, which in turn eases implementation and readability. Debugging, testing, documenting and maintenance of product also increase due to modularity.

### What Do You Mean by Complexity of Algorithm?

Complexity is an essential concept in Data structure. When you talk about complexity is related to computer, you call it as computational complexity. It can be termed as the characterization of time and space requirements for solving a problem using some specific algorithm. These requirements are expressed regarding one single parameter which is used to represent the size of the problem.

Example:

Let 'n' denotes the size of the problem. Then the time required for a specific type of  algorithm for solving a problem can be expressed as:

f : R -> R, where f is the function and f(n) is the most significant amount of time needed. Thus you can conclude that analysis of any program requires two vital concepts:

* Time Complexity
* Space Complexity

Time Complexity of a program can be defined as the amount of time the computer takes to run a program to its completion. On the other hand, the space complexity of an algorithm can be defined as the memory that it needs to run that algorithm to its completion.