**Nonlinear behavior of complexity**

As programming projects became larger, an interesting scenario observed. A task that would take one programmer two months to perform could not be accomplished by two programmers in one month. Fred Brooks in 1975 stated that “The bearing of a child takes nine months, no matter how many women are assigned to the task”.

The reason for this nonlinear behavior was complexity-in particular, the interconnections between software components were complicated, and large quantities of information had to be communicated among the various members of that programming team. Since software construction is inherently a system effort i. e. an exercise interrelationships communication effort is great, and it quickly dominates the decrease in individual task time brought about by portioning. Adding more men not shortens the length (duration) of the schedule.

In this complex situation, it is not simply the complete size of the task undertaken, because size by itself would not be a difficulty to portioning each into several pieces. The unique feature of software system developed using traditional methods, the most complex systems developed is high degree of interconnectedness. Interconnectedness means dependence of one portion of code on another portion. If one task is useful to other part of the program, there must be some communication of information either into or out of the component under consideration. In short, an individual section of code cannot be understood in isolation.

**Introduction to Object-Oriented Paradigm:**

            Development process in software technology has been dynamic since its beginning as new tools and techniques are announced in quick succession. If we see the software evolution of computer, then we find that programming techniques began from machine language at first, then assembly language, then Procedure-oriented and finally object-oriented programming.

            So, object-oriented programming is the most recent concept in computer programming paradigms and it has been defined as “an approach that provides a way of modularizing programs by creating partitioned memory area for both data and functions that can be used as templates for creating copies of such modules on demand”.

OOP approach was invented to overcome the drawbacks of structural programming approach/ procedural oriented programming(POP). Unlike structural programming approach, it uses the bottom up programming approach. OOP treats data as  critical (very important) element in the program development and does not allow data to move freely around the program. OOP ties the data more closely to the function that operates on it and protects data from accidental modification from outside function. OOP allows decomposition of a program into a number of entities called objects and then builds data and functions around these objects. The data of an object can be accessed only by the function associated with that function. You can’t access the data directly. The data is hidden, so it is safe from accidental modification. Data and its functions are said to be encapsulated into a single entity. Data encapsulation, data hiding, inheritance and polymorphism  are key terms in the description of object-oriented languages.

Some main features of OOP are as follows:

i)             Emphasis is on data rather than procedures.

ii)           Programs are divided into objects.

iii)         Functions that operate on the data of an object are tied together in the data structure.

iv)         Data is hidden inside the object and cannot be accessed by external functions.

v)           Objects may communicate with each other through functions.

vi)         Supports inheritance, polymorphism and operator overloading.

vii)       New data and functions can be easily added whenever necessary.

viii)     Follows bottom-up approach in program design.

Languages that supports OOP paradigm are Simula, C++, Smalltalk, Ada, Java, C# etc.

**Differences between OOP and Structured Programming**

OOP is the most recent programming technique whereas Structured Programming is the earlier form of programming and it is also known as Procedural Oriented Programming (POP). The major differences between OOP and Structured Programming Languages are as follows:

|  |  |
| --- | --- |
| **OOP** | **POP** |
| i) It is the most recent programming concept which uses the bottom-up approach. | i) It is the old programming concept that uses the top-down approach. |
| ii) Programs are divided into a number of entities called objects. | ii) Programs are divided into a number of functions. |
| iii) Emphasis is on data rather than on procedures. | iii) Emphasis is on procedures rather than on data. |
| iv) It does not allow data to move freely around the program from one object to another object. | iv) It allows data to move freely around the program from one function to another function. |
| v) Data is hidden inside the object and cannot be accessed by external functions due to encapsulation. | v) Data is open and can be freely accessed by all the programs as encapsulation is not possible. |
| vi) Problem is viewed as a real world entity. | vi) Problem is not viewed as a real world entity. |
| vii) It supports inheritance and polymorphism in programming. | vii) It does not support inheritance and polymorphism. |
| viii) OOP is written by using HLL such as C++, Java, SIMULA, Smalltalk et. | viii) POP is written by using HLL such as C, Pascal, FORTRAN et. |

**Object-oriented programming as a new paradigm:**

            The English word ‘paradigm’ is derived from the Latin word ‘*paradigma’* which means ‘model’. So, in English too, paradigm means ‘any example or model’. Due to its popularity, completeness and robustness, OOP is frequently referred to as a new programming paradigm in the world of computer programming.

Actually, the word ‘paradigm’ was used for the first time in computer programming by Robert Floyd in his speech in 1979 ACM Turing Award lecture. A programming paradigm is a way of conceptualizing what it means to perform computation and how tasks to be carried out on a computer should be structured and organized.

The style of problem solving used in the object-oriented technique is quite similar to the method used to address problems in everyday life. Hence, OOP has been most popular and almost all the standard programs developed today are based on OOP.

OOP has been the new paradigm in computer programming field due to the following concepts/characteristics:

1.      Object: - Objects are the basic run time entities in an object-oriented system. They may represent a person, a place, a bank account, a table of data or any item that the program has to handle. In OOP, programming problem is analyzed in terms of objects and the nature of communication between them.

2.      Classes: - An abstract description of the data and behavior of a collection of similar objects. Classes are user-defined data types and behave like the built-in data types of a programming language.

3.       Data abstraction and encapsulation: - The Data is encapsulated by the method; this mean that no one can deal with the data directly and the only thing that can work with the data is the method of the same class. So, we deal with the data through the method. This is good because:

a.        no one can change the data value directly; this provides a significant level of protection against accidental modification or incorrect use.

b.      All the Member Data being local variable, this is easy to be maintained.

4.      Inheritance: - Inheritance is the process by which one object can acquire the properties of another object. This is important because it supports the concept of classification.

5.      Polymorphism: - Polymorphism allows one name to be used for several related but slightly different purposes. The purpose of polymorphism is to let one name be used to specify a general class of action. Depending upon what type of data it is dealing with, a specific instance of the general case is executed.

6.       Dynamic binding: - Binding refers to the linking of a procedure call to the code to be executed in response to the call and dynamic binding or late binding means that the code associated with a given procedure call is not known until the time of the call at run-time.

7.      Message passing: - A message for an object is a request for execution of a procedure. Message passing involves specifying the name of the object, the name of the function (message) and the information to be sent.