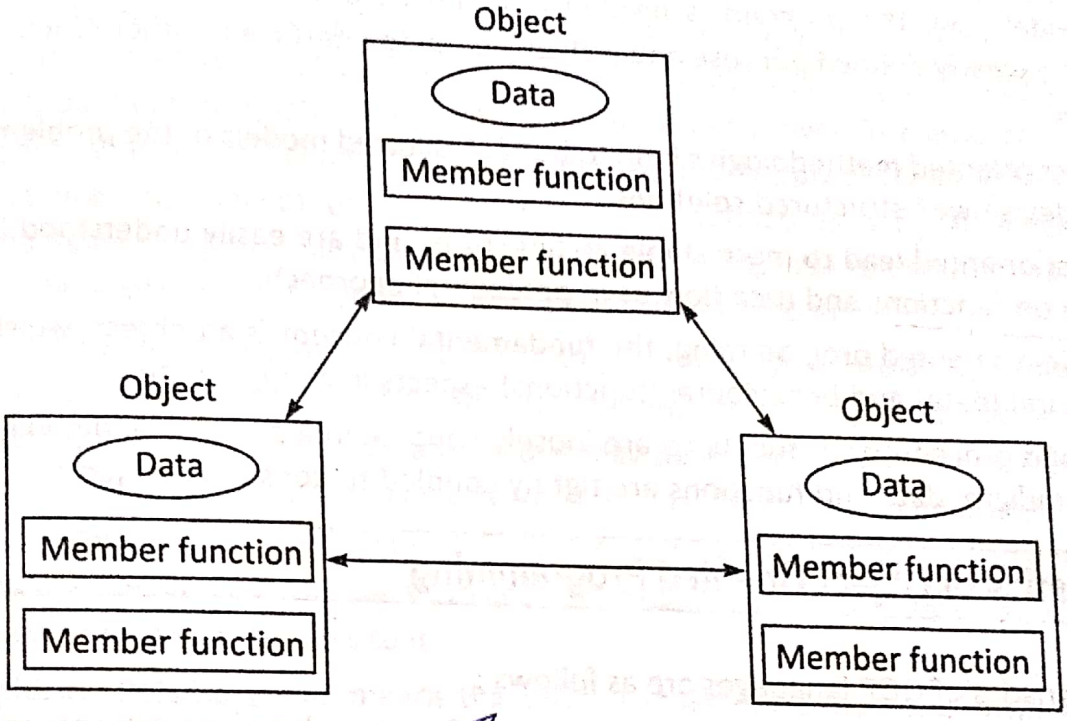
Object Oriented Methodologies

Unlike the traditional paradigms, where the emphasis was on functions, object oriented programming emphasis on both data as well as on functions. Data and functions are encapsulated in a single unit called objects. In structured and procedure programming language, the problem is divided into functions, whereas in OOP the problem is divided into objects.



Object Oriented Paradigm

Object oriented programming is a concept that was created because of the need to overcome the problems that were found with using structured programming techniques. While structured programming uses an approach which is top-down. OOP uses an approach which is bottom up. OOP paradigm is shown in below mentioned figure.

In contrast to procedural paradigm which has a large single store where all procedures work, in object oriented paradigm, procedures operate on abstract values called objects which can be created and destroyed dynamically. This programming paradigm is based on the idea of communicating between objects to simulate the temporal evolution of a set of real world phenomena. Data as well as operations are encapsulated in objects. Objects interacts by means of message passing and create the functionality of a larger program. They take in certain data, process it and pass it to another object. The set of functions through which they interact is called the interface. Information hiding is used to protect the internal properties of an object.

In object oriented paradigm, objects are grouped into classes. Objects in classes are similar enough to allow programming of the classes, as they are opposed to programming of individuals objects. Classes are organised into inheritance hierarchies. This provides for class extension or specialisation. Inheritance allows new objects to be defined in terms of other existing objects. To make an OO design, one is required to decide which classes are needed, then provide a full set of operations for each class.

✔ Need of Object Oriented Programming

Object oriented programming was developed because of limitations discovered in other programming paradigms, especially its close pre-decessor, procedural paradigm. A program in procedural paradigm is a collection of instructions. When program becomes larger, a single list of instructions becomes unwidely. So, the program is divided into functions or procedures and each functions or procedure has a clearly defined purpose and a clearly defined interface to other functions or procedures in the program.

(1) Object oriented methodologies help to build structured models of the problem domain at hand and devise well structured solutions.

(2) Object oriented lead to more stable architectures and are easily understood than those based solely on functions and data flow as in procedural approach.

(3) In Object oriented programming, the fundamental concept is an object, which combines both structural (data) and behavioural (functions) aspects in a single entity.

(4) Data and procedures or functions are loosely coupled in a procedural paradigm whereas in an OO paradigm, data and functions are tightly coupled to constitute objects.

✔ Characteristics of Object Oriented Programming

The characteristics of OOP languages are as follows :

(1) Objects : These are runtime states of a conceptual framework encapsulating typed data and typed operations that correspond to a real world entity or thing for the purpose of computational modeling. Objects are the basic runtime entities in an object oriented system. Programming problem is analysed in terms of objects and nature of communication between them. When a program is executed, objects interact with each other by sending messages. Different objects can also interact with each other without knowing the details of their data or code.

(1) Classes : These are static (compile-time) definitions of a new type of a collection of data and associated operations (procedures or functions) from which runtime instances called objects can be created. A class is a collection of objects of similar type. Once a class is defined, any number of objects can be created which belong to that class. Class defines the abstract characteristics of a thing (object), including the thing's characteristics (its attributes, fields or properties) and thing's behaviour (the thing's it can do or methods, operations or features). One might say that a class is a blueprint or factory that describes the nature of something.

✓ Properties of OOP

The various properties of an object oriented programming are as follows :

(1) Inheritance : The ability to declare and define new classes as specialization from existing classes is called inheritance. Specialization is defined in terms of added data and/or procedures or methods. Inheritance is the process by which objects can acquire the properties of objects of other class. In OOP, inheritance provides reusability, like adding features to an existing class without modifying it. This can be achieved by deriving a new class from the existing one. The new class will have combined features of both the classes.

2. Data Abstraction (Information Hiding) : Abstraction refers to the act of representing essential features without including the background details or explanations. An abstraction denotes the essential characteristics of an object that distinguish it from all other kind of objects and thus provides crisply defined conceptual boundaries, relative to the perspective of the viewer. The goal of abstraction is to isolate those aspects that are important for some purpose and suppress those aspects that are not important. Classes use the concept of abstraction and are defined as a list of abstract attributes. Abstraction is simplifying complex reality by modeling classes appropriate to the problem and working at the most appropriate level of inheritance for a given aspect of the problem.

3. Encapsulation : Storing data and functions in a single unit (class) is known as encapsulation. This is complementary to abstraction and is used to describe the process of defining the code and data, which form an object within the object, some data and code may be accessible directly as the interface of the object, while other parts of the object remain private. It is usual for all data to be private, access to values being through the return of member functions. Data cannot be accessible to the outside world and only those functions which are stored in the class can access it. Encapsulation conceals the functional details of a class from objects that send messages to it.

4. Polymorphism : Polymorphism means the ability to take more than one form. An operation may exhibit different behaviours in different instances. The behaviour depends on the datatypes used in the operation. Polymorphism is extensively used in implementing inheritance. Polymorphism allows the programmer to treat derived class members just like then parent class members. Polymorphism in OOP is the ability of objects belonging to different data types to respond to method calls of methods of the same name, each one according to an appropriate type specific behaviour.

Advantages of OOP

The major advantages of OOP's are as follows :

1. Simplicity : Software objects model real world objects, so the complexity is reduced and the program structure is very clear.

2. Modularity : Each object forms a separate entity whose external workings are decoupled from other parts of the system. OOP provides a clear modular structure for programs which makes it good for defining abstract data types where implementation detail are hidden and the unit has clearly defined interface.