

OOPS-Features

Features of OOP

The basic features of Object Oriented Programming are:

- 1)Objects
- 2)Classes
- 3)Data Abstraction and Encapsulation
- 4)Inheritance
- 5)Polymorphism
- 6)Dynamic Binding
- 7)Message Passing

Objects:They are basic run-time entities in an object-oriented system containing data and code to manipulate that data. They may represent a person, a place, a table of data or any item that the program has to handle. They may also represent user-defined data such as vectors, time and lists. Programming problem is analyzed in terms of objects and the nature of communication between them.

Program objects should be chosen such that they match closely with the real-world objects.

Objects take up space in the memory and

have an associated address like a record in Pascal, or a structure in C. They can interact without having to know details of each

other's data or code. It is sufficient to know the type of message accepted, and the type of response returned by the objects.

Classes:The entire set of data and code of an object can be made a user defined data type with the help of a class. Objects are

variables of the type class. Once a class is defined, we can create any number of objects belonging to that class. Each object is

associated with the data of type class with which they are created. A class thus is a collection of objects of similar type. Classes

are user-defined data types and behave like built-in types of a programming language.

Data Abstraction and Encapsulation:The wrapping up of data and functions into a single unit(called class) is known as encapsulation.

Data encapsulation is the most striking feature of a class. The data is not accessible to the world, and only those functions which

are wrapped in the class can access it. These functions provide the interface between the object's data and the program. This

insulation of the data from direct access by the program is called data hiding or information hiding. Abstraction refers to the act of representing essential features without including the background details or explanations. Classes use the concept of abstraction and are defined as a list of abstract attributes such as size, weight and cost, and functions to operate on these attributes. They encapsulate all the essential properties of the objects that are to be created.

Inheritance:It is the process by which objects of one class acquire the properties of objects of another class. It supports the concept of hierarchical classification. It provides the idea of reusability. This means that we can add additional features to an existing class without modifying it. This is possible by deriving a new class from the existing one. The new class will have the combined features of both the classes. This feature allows the user to tailor the class in such a way that it does not introduce any undesirable side-effects into the rest of the classes.

Polymorphism:It is a Greek term which means the ability to take more than one form. An operation may exhibit different behaviors in different instances. The behavior depends upon the types of data used in the operation. The process of making an operator exhibit different behaviors in different instances is known as operator overloading. Using a single function name to perform different types of tasks is known as function overloading. It plays an important role in allowing objects having different internal structures to share the same external interface. This means that a general class operations may be accessed in the same manner even though specific actions associated with each operation may differ. It is extensively used in implementing inheritance.

Dynamic Binding:It refers to linking a procedure call to the code to be executed in response to the call. Dynamic binding means that the code associated with a given procedure call is not known until the time of the call at run-time. It is associated with polymorphism and inheritance. A function call associated with a polymorphic reference depends on the dynamic type of that reference. At run-time, the code matching the object under current reference will be called.

Message Passing:Objects communicate with one another by sending and receiving information much the same way as people pass messages to

one another. The concept of message passing makes it easier to talk about building systems that directly model or simulate their real world counterparts. A message for an object is a request for execution of a procedure, and therefore will invoke a function(procedure) in the receiving object that generates the desired result. Message passing involves specifying the name of the object, the name of the function(message) and the information to be sent. Objects have a life cycle. They can be created and destroyed. Communication with an object is feasible as long as it is alive.