

Module-1

Abstraction :-

Abstraction is the process of taking away or removing characteristics from something in order to reduce it to a set of essential characteristics.

Abstraction means to focus on the essential features of an element or object in OOP, ignoring its extraneous or ~~object in oop~~ accidental properties.

The essential features are relative to the context in which the object is being used.

It is two types :-

(i) Data Abstraction.

(ii) Process Abstraction.

(i) Data Abstraction :-

When the object data is not visible to the outer world, it creates data abstraction. If needed, access to the object's data is provided through some methods.

(ii) Process Abstraction :-

When we don't need to provide details about all the functions of an object. When we hide the internal implementation of the different functions involved in a user operation, it creates process abstraction.

Encapsulation :-

Encapsulation is also called an information hiding concept. The data and operations are linked to a single unit.

Encapsulation not only bundles essential information of an object together but also restricts access to the data and methods from the outside world.

Information hiding :-

Information hiding is the principle of segregation of the design decisions in a computer program that are most likely to change,

thus protecting other parts of the program from extensive modification if the design decision is changed.

Inheritance :-

Inheritance is the mechanism that permits new classes to be created out of existing classes by extending and refining its capabilities.

The existing classes are called the base classes/parent classes/super-classes and the new classes are called the derived classes/child classes/sub-classes.

It is 5 types of Inheritance.

(i) Single

(ii) Multiple

(iii) Multi level

(iv) Hierarchical

(v) Hybrid

(i) Single Inheritance :-

A subclass derives from a single super-class.

(ii) Multiple Inheritance :-

A subclass derives from more than one super classes.

(iii) Multi level Inheritance :-

A subclass derives from a super-class which in turn is derived from another class and so on.

(iv) Hierarchical Inheritance :-

A class has a number of subclasses each of which may have subsequent subclasses, continuing for a number of levels so as to form a tree structure.

(v) Hybrid Inheritance :-

A combination of multiple and multilevel inheritance so as to form a lattice structure.

Dynamic Binding :- (Late binding)

Dynamic binding or late binding is the mechanism a computer program waits until runtime to bind the name of a method called to an actual subroutine.

It is an alternative to early binding or static binding where this process is performed at compile-time.

Polymorphism :-

polymorphism is originally a Greek word that means the ability to take multiple forms.

In object oriented paradigm, polymorphism implies using operations in different ways, depending upon the instance they are operating upon.

polymorphism allows objects with different internal structures to have a common external interface.

polymorphism is particularly effective while implementing inheritance.

Overview of OOAD :-

Object-oriented analysis and design (OOAD) is a technical approach for analyzing and designing an application, system or business by applying object oriented programming, as well as using visual modeling throughout the software development process to guide stakeholder communication and product quality.

It is based on a set of basic principles, which are as follows -

- (1) The information domain is modeled.
- (2) Behaviour is represented.
- (3) The function is described.
- (4) Data, functional, and behavioural models are divided to uncover greater detail.
- (5) Early models represent the essence of the problem, while later ones provide implementation details.

Object - Oriented Design (OOD) :-

An analysis model created using object-oriented analysis is transformed by object-oriented analysis design into a design model that works as a plan for software creation.

OOD results in a design having several different levels of modularity.

The major system components are partitioned into subsystems and data manipulation operations are encapsulated into objects.

A design pyramid for object-oriented systems is having the following four layers.

(i) The subsystem layer :-

It represents the subsystem that enables software to achieve user requirements and implement technical frameworks that meet user needs.

(ii) The class and object layer :-

It represents the class hierarchies that enable the system to develop using generalization and specialization. This layer also represents each object.

(iii) The Message layer :-

It represents the design details that enable each object to communicate with its partners. It establishes internal and external interfaces for the system.

(iv) The Responsibilities layer :-

It represents the data structure and algorithmic design for all the attributes and operations for each object.

Object Oriented Concepts include:

(1) Attribute: A collection of data values that describe a class.

(2) Class: encapsulates the data and procedural abstractions required to describe the content and behaviour of some real world entity.

In other words,

A class is a generalized description that describes the collection of similar objects.

(3) Objects: Instances of a specific class. Objects inherit a class's attributes and operations.

(4) Operations: also called methods and services, provide a representation of one of the behaviours of class.

(5) Subclass: a specialization of the super class. A subclass can inherit both attributes and operations from a super class.

(6) Superclass: also called a base class, is a generalization of a set of classes that are related to it.