Object Oriented Programming Paradigm **OOP**

The Object Oriented programming paradigm plays an important role in human computer interface. It has different components that takes real world objects and performs actions on them, making live interactions between man and the machine. Following are the components of OOPP –

- This paradigm describes a real-life system where interactions are among real objects.
- It models applications as a group of related objects that interact with each other.
- The programming entity is modeled as a class that signifies the collection of related real world objects.
- Programming starts with the concept of real world objects and classes.
- Application is divided into numerous packages.
- A package is a collection of classes.
- A class is an encapsulated group of similar real world objects.

**Objects**

Real-world objects share two characteristics – They all have state and behavior. Let us see the following pictorial example to understand Objects.

In the above diagram, the object ‘Dog’ has both state and behavior.

An object stores its information in attributes and discloses its behavior through methods. Let us now discuss in brief the different components of object oriented programming.

**Data Encapsulation**

Hiding the implementation details of the class from the user through an object’s methods is known as data encapsulation. In object oriented programming, it binds the code and the data together.
and keeps them safe from outside interference.

**Public Interface**

The point where the software entities interact with each other either in a single computer or in a network is known as public interface. This help in data security. Other objects can change the state of an object in an interaction by using only those methods that are exposed to the outer world through a public interface.

**Class**

A class is a group of objects that has mutual methods. It can be considered as the blueprint using which objects are created.

Classes being passive do not communicate with each other but are used to instantiate objects that interact with each other.

**Inheritance**

Inheritance as in general terms is the process of acquiring properties. In OOP one object inherit the properties of another object.

**Polymorphism**

Polymorphism is the process of using same method name by multiple classes and redefines methods for the derived classes.

**Example**

![Diagram of Animal hierarchy](image)

**Object Oriented Modeling of User Interface Design**

Object oriented interface unites users with the real world manipulating software objects for designing purpose. Let us see the diagram.
Interface design strive to make successful accomplishment of user’s goals with the help of interaction tasks and manipulation.

While creating the OOM for interface design, first of all analysis of user requirements is done. The design specifies the structure and components required for each dialogue. After that, interfaces are developed and tested against the Use Case. Example – Personal banking application.

The sequence of processes documented for every Use Case are then analyzed for key objects. This results into an object model. Key objects are called analysis objects and any diagram showing relationships between these objects is called object diagram.