

Object Oriented Programming Concepts (UAI-1)

1 INTRODUCTION

Over last few decades, the entire world has seen tremendous revolution and major credit for it goes to technology revolution of which 'Software Programming' has been an essential part. Programming style, paradigms and languages, everything, have evolved tremendously in last few decades. Programming that started with *machine language*, moved from assembly to higher level languages and many programming paradigms. *Procedural programming* and *object oriented programming* have been two most common programming paradigms.

In this informative chapter, we are talking about the object oriented programming and how it is implemented in Python, which is an object oriented language.

2 WHAT IS OBJECT ORIENTED PROGRAMMING PARADIGM ?

A **programming paradigm** refers to the way or style of writing a program. There have been different programming paradigms such as *procedural programming paradigm*, *modular programming paradigm*, *object oriented programming paradigm* etc.

Before we talk about *object oriented paradigm*, it is important to discuss briefly about the *procedural paradigm*, to appreciate the difference.

Procedural Programming Paradigm

A program written with a procedural programming paradigm is a list of instructions where each statement tells the computer to do something. The focus is on the processing, *i.e.*, the algorithm needed to perform the desired computation. That is, the emphasis is on doing things, *i.e.*, the action and not on data, even though the data is an essential part of the system. So a *procedural program will view a problem/system as a sequence of actions taking place.*

Object Oriented Programming Paradigm

The object-oriented approach, unlike procedural programming, views a problem in terms of objects involved rather than the procedure for doing it. Do you know what an object is? Well, *an object is an identifiable entity with same characteristics and behaviour.*

For instance, we can say 'Orange' is an object. Its characteristics are: it is *spherical shaped*, its *colour is orange* etc. Its behaviour is: it is *juicy, citrus* and it *tastes sweet-sour*.

Similarly, the *(fan on your ceiling)* is also an object. Its characteristics are: *its colour, its motor capacity, number of blades it has*, etc. Its behaviour is: *it can rotate air at a specific speed in rpm (rotations per minutes)*.

To understand this concept more clearly, let us consider an example. Let us *simulate traffic flow at a red light crossing*.

As you know, procedural programming paradigm focuses on the procedures or the working action. Using procedural programming paradigm, the above said problem will be viewed in terms of working or actions happening in the traffic-flow i.e., **moving, halting, turning, accelerating** etc. The OOP paradigm, however, aims at the objects and their interface. Thus in OOP approach, the traffic-flow problem will be viewed in terms of the objects involved. The objects involved are: **cars, trucks, buses, scooters, auto-rickshaws, taxis** etc.

Let us understand another very important term class. A **Class** is a template / blueprint representing a group of objects that share common properties and relationships.

In the above example, cars have been identified as objects. They have characteristics like: *steering wheel, seats, a motor, brakes* etc. and their behaviour is their *mobility*. **Car**, however, is not an object, it is a class (compare with the definition). A specific car "Optra 52xyz" is an object of class type 'car'. An 'Object' is an instance of 'class'. For example, we can say 'parrot' is a class but 'mithu, the parrot' is an object of this class type.

While programming using OOP approach, the **characteristics** of an object are represented by **its data**, and its **behaviour** is represented by its **associated functions**.

OBJECT

An **Object** is an identifiable entity with some characteristics and behaviour.

CLASS

A **Class** is a template / blueprint representing a group of objects that share common properties and relationships.

3 OBJECT ORIENTED PROGRAMMING (OOP) CONCEPTS

The object oriented programming has been developed with a view to overcome the drawbacks of conventional programming approaches. The OOP approach is based on certain concepts, which are given below:

- ❖ Data Abstraction and Data Hiding
- ❖ Data Encapsulation
- ❖ Inheritance
- ❖ Polymorphism

3.1 Data Abstraction and Data Hiding

Abstraction is the concept of simplifying a real world concept into its essential elements.

To understand abstraction, let us take an example. You are driving a car. You only know the essential features to drive a car e.g., *gear handling, steering handling, use of clutch, accelerator, brakes* etc. etc. But while driving do you get into internal details of car like wiring, motor working etc. You just change the gears or apply the brakes etc. What is happening inside is hidden from you. This is abstraction where you only know the essential things to drive a car without including the background details or explanations.