# Python OOPs Concepts

Python is an object-oriented programming language. You can easily create and use classes and objects in Python.

Major principles of object-oriented programming system are given below:

* Object
* Class
* Method
* Inheritance
* Polymorphism
* Data Abstraction
* Encapsulation

## Object

Object is an entity that has state and behavior. It may be anything. It may be physical and logical. For example: mouse, keyboard, chair, table, pen etc.

Everything in Python is an object, and almost everything has attributes and methods. All functions have a built-in attribute \_\_doc\_\_, which returns the doc string defined in the function source code.

## Class

Class can be defined as a collection of objects. It is a logical entity that has some specific attributes and methods. For example: if you have an employee class then it should contain an attribute and method i.e. an email id, name, age, salary etc.

**Syntax:**

1. class ClassName:
2. <statement-1>
3. .
4. .
5. .
6. <statement-N>

## Method

Method is a function that is associated with an object. In Python, method is not unique to class instances. Any object type can have methods.

## Inheritance

Inheritance is a feature of object-oriented programming. It specifies that one object acquires all the properties and behaviors of parent object. By using inheritance you can define a new class with a little or no changes to the existing class. The new class is known as derived class or child class and from which it inherits the properties is called base class or parent class.

It provides re-usability of the code.

## Polymorphism

Polymorphism is made by two words "poly" and "morphs". Poly means many and Morphs means form, shape. It defines that one task can be performed in different ways. For example: You have a class animal and all animals talk. But they talk differently. Here, the "talk" behavior is polymorphic in the sense and totally depends on the animal. So, the abstract "animal" concept does not actually "talk", but specific animals (like dogs and cats) have a concrete implementation of the action "talk".

## Encapsulation

Encapsulation is also the feature of object-oriented programming. It is used to restrict access to methods and variables. In encapsulation, code and data are wrapped together within a single unit from being modified by accident.

## Data Abstraction

Data abstraction and encapsulation both are often used as synonyms. Both are nearly synonym because data abstraction is achieved through encapsulation.

Abstraction is used to hide internal details and show only functionalities. Abstracting something means to give names to things, so that the name captures the core of what a function or a whole program does.

## Object-oriented vs Procedure-oriented Programming languages

|  |  |  |
| --- | --- | --- |
| **Index** | **Object-oriented Programming** | **Procedural Programming** |
| 1. | Object-oriented programming is an approach to problem solving where computation is done by using objects. | Procedural programming uses a list of instructions to do computation step by step. |
| 2. | It makes development and maintenance easier. | In procedural programming, It is not easy to maintain the codes when project becomes lengthy. |
| 3. | It simulates the real world entity. So real world problems can be easily solved through oops. | It doesn't simulate the real world. It works on step by step instructions divided in small parts called functions. |
| 4. | It provides data hiding. so it is more secure than procedural languages. You cannot access private data from anywhere. | Procedural language doesn't provide any proper way for data binding so it is less secure. |
| 5. | Example of object-oriented programming languages are: C++, Java, .Net, Python, C# etc. | Example of procedural languages are: C, Fortran, Pascal, VB etc. |