**Python OOPs Concepts**

In Python, object-oriented Programming (OOPs) is a programming paradigm that uses objects and classes in programming.

The main concept of OOPs is to bind the data and the functions that work on that together as a single unit so that no other part of the code can access this data.

## Main Concepts of Object-Oriented Programming (OOPs)

* Class
* Objects
* Method
* Inheritance
* Polymorphism
* Data abstraction
* Encapsulation

**Class**

A class is a collection of objects. A class contains the blueprints or the prototype from which the objects are being created. It is a logical entity that contains some attributes and methods.

**Some points on Python class:**

* Classes are created by keyword class.
* Attributes are the variables that belong to a class.
* Attributes are always public and can be accessed using the dot (.) operator.

**Objects**

The object is an entity that has a state and behaviour associated with it. It may be any real-world object like a mouse, keyboard, chair, table, pen, etc. Integers, strings, floating-point numbers, even arrays, and dictionaries, are all objects. More specifically, any single integer or any single string is an object.

**Eg**:- The number 12 is an object, the string “Hello, world” is an object, a list is an object that can hold other objects, and so on.

**An object consists of:**

* **State:** It is represented by the attributes of an object. It also reflects the properties of an object.
* **Behaviour:** It is represented by the methods of an object. It also reflects the response of an object to other objects.
* **Identity:** It gives a unique name to an object and enables one object to interact with other objects.

The \_\_init\_\_( ) Function (constructor)

All classes have a function called \_\_init\_\_( ), which is always executed when the class is being initiated.  
Use the \_\_init\_\_( ) function to assign values to object properties,  
or other operations that are necessary to do when the object is being created

## Methods

Methods are functions defined inside the body of a class. They are used to define the behaviors of an object.

## **Inheritance**

Inheritance is the capability of one class to derive or inherit the properties from another class. The class that derives properties is called the derived class or base class and the class from which the properties are being derived is called the base class or parent class. The benefits of inheritance are:

* It represents real-world relationships well.
* It provides the reusability of a code. We don’t have to write the same code again and again. Also, it allows us to add more features to a class without modifying it.
* It is transitive in nature, which means that if class B inherits from another class A, then all the subclasses of B would automatically inherit from class A.

**Parent class  
  
child class**

**Polymorphism:**

Polymorphism contains two words "poly" and "morphs". Poly means many, and morph means shape.  
By polymorphism, we understand that one task can be performed in different ways.