OOPs CONCEPT IN PYTHON

- Class in Python
- Objects in Python
- Polymorphism in Python
- Encapsulation in Python
- Inheritance in Python
- Data Abstraction in Python

Class in python: A class is a collection of objects. For example: if you have an employee class, then it should contain an attribute and method, i.e. an email id, name, age, salary.

SYNTAX:

```
class ClassName:

    # Class attributes and methods;
```

Object in python: The object is an entity that has state and behavior. It may be any real-world object like the mouse, keyboard, chair, table, pen, etc.

An object consists of:

- **State:** It is represented by the attributes of an object. It also reflects the properties of an object.
- **Behavior:** 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.

Polymorphism in python:

Polymorphism in Python refers to the ability of different objects to respond to the same method or operation in different ways. It allows a single interface to represent different types or classes, making code more flexible and reusable.

For example, different classes may define their own version of a method with the same name, and the behaviour will depend on the object calling the method.

Encapsulation in python:

Encapsulation in Python is an object-oriented programming concept where data (attributes) and methods (functions) that operate on the data are wrapped together in a single unit (a class). It provides data hiding.

Inheritance in python:

Inheritance allows one class (the **child class**) to inherit the attributes and methods of another class (the **parent class**). This promotes code reuse and makes it easier to create and maintain a hierarchy of classes.

Data abstraction in python: Data Abstraction in Python is an object-oriented programming concept that focuses on **hiding implementation details** while exposing only

the essential features of an object or class. The goal is to simplify the interface and reduce complexity for the user of the object.