**Object-Oriented Programming (OOP)**

is a programming paradigm that revolves around the concept of objects, which are instances of classes encapsulating both data and behavior. OOP promotes abstraction by bundling related data and functions into cohesive units, facilitating the modeling of real-world entities effectively. It encourages code reusability through mechanisms like inheritance and composition, allowing the creation of hierarchies and sharing of functionality among classes. Encapsulation ensures that the internal details of objects are hidden, exposing only the necessary interfaces for interaction. Additionally, OOP supports polymorphism, enabling objects of different classes to be treated uniformly through interfaces or base classes.

**Procedural-Oriented Programming (POP)**

**o**n the other hand, is a programming paradigm that focuses on procedures or functions that operate on data directly. In POP, the code is organized around procedures, with less emphasis on abstraction and object-oriented principles. While POP also promotes code reusability through functions, it lacks built-in mechanisms for encapsulation, inheritance, and polymorphism. This paradigm tends to be more straightforward and suitable for simpler projects where the focus is on procedural flow rather than complex object interactions. However, POP can lead to code duplication and maintenance challenges as the project scales, compared to the more structured approach offered by OOP.