Python Notes

Inheritance

* Inheritance is an OOPS concept where a class (child class) derives attributes and methods from another class (parent class).
* It helps in code reusability and creating a hierarchical structure in programs.
* The child class can;
  + Use methods and attributes from the parent class.
  + Override methods to provide its own implementation.
  + Extend functionality by adding new methods.

Types of Inheritance:

* Single Inheritance:
  + A single child class inherits from a single parent class.
  + The child class gets access to all attributes and methods of the parent class.
* Multiple Inheritance:
  + A child class inherits from more than one parent class.
  + The child class combines functionalities of multiple parents.
  + Python follows the MRO (Method Resolution Order) to handle conflicts.
* Multilevel Inheritance
  + A child class inherits from another child class, forming a chain.
  + The grandchild class gets properties from both parent and grandparent class.
* Hierarchical Inheritance
  + Multiple child classes inherit from a single parent class.
  + Each child class extends or modifies the functionality differently.
* Hybrid Inheritance
  + A combination of multiple inheritance types.
  + It can involve a mix of single, multiple, and multilevel inheritance.

Key Concepts:

1. Super Method()
   1. Used in a child class to call a method from the parent class.
   2. Helps in method overriding while still using parent class functionality.
2. Method Overriding
   1. A child class redefines a method that already exists in the parent class.
   2. The method in the child class overrides the parent’s version.
3. Constructor Inheritance (\_\_init\_\_)
   1. The parent class constructor can be called in the child class using super().\_\_init\_\_().
   2. Helps in reusing initialization logic.
4. MRO (Method Resolution Order)
   1. Python resolves method conflicts in multiple inheritance using C3 Linearization (MRO).
   2. Uses depth-first, left-to-right approach.

Inheritance allows one class to derive properties from another.