Q1. What is the concept of a metaclass?

A metaclass is a class that defines the behavior of other classes, often referred to as the "class of a class." It controls how classes are constructed, initialized, and behave. Metaclasses offer a way to customize and extend the behavior of class creation and provide advanced code-level control over class attributes and methods.

Q2. What is the best way to declare a class's metaclass?

class MyClass(metaclass=MyMeta):

Class definition

Q3. How do class decorators overlap with metaclasses for handling classes?

Class decorators and metaclasses can overlap in handling classes by both allowing you to modify or enhance class behavior. However, class decorators are applied after the class is defined, modifying the class itself, while metaclasses are responsible for the creation and definition of the class before it is fully constructed.

Q4. How do class decorators overlap with metaclasses for handling instances?

Class decorators and metaclasses overlap less in handling instances compared to handling classes. Class decorators primarily affect the class itself or its methods, while metaclasses deal with class-level attributes, methods, and construction. Instances are typically not directly impacted by class decorators or metaclasses.