Q1. What is the concept of a metaclass?

Ans=> A metaclass in Python is a class of a class that defines how a class behaves. A class is itself an instance of a metaclass

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

Ans => MyClass = type('MyClass', (), {})

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

Ans=> class decorators described in the preceding chapter sometimes overlap with metaclasses in terms of functionality. Although they are typically used for managing or augmenting instances, class decorators can also augment classes, independent of any created instances

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

Ans=>same as above