Q1. What is the concept of a metaclass?

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. A class in Python defines how the instance of the class will behave

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

add a \_\_metaclass\_\_ attribute when you write a class

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

Decorators simply augment and return the original class object. Metaclass augment a class after they create it.

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

Decorators and metaclasses can manage instances by inserting a wrapper proxy object to catch instance creation calls