1. What is the concept of a metaclass?

Ans: Metaclass is a class of a class that defines how a class behaves. A class is an instance of Metaclass, and any instance of Class is an Instance of type metaclass. Example: int, str, float, list, tuple, etc type is of metaclass type.

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

Ans: A way to declare a class' metaclass is by using metaclass keyword in class definition.

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

Ans: The decorator approach: Syntax differs from ordinary class statements, easier to construct bunch classes dynamically and point class is an instance of type.
The __metaclass__ approach: Syntax same as ordinary class statement, 'Magic' takes place behind the scenes, tricky to construct Bunch classes dynamically, point class is an instance of MetaBunch.
Decorators are simpler than using __metaclass__, particularly if the decorator syntax is available.

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

Ans: Anything that can be done with a class decorator can also be done with a custom metaclass.