Q1. What is the concept of a metaclass?

Ans. Metaclass is used to create a class and are subclasses of type class. Metaclasses can also be used as a simple function , for example they are always responsible for making and returning an object for the new class. Metaclasses may have a methods and data to provide a behaviour for their classed too and a secondary pathway for inheritance

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

Ans Type is default metaclass in python ,but we can create a meta class in the following way

class meta\_class(type):

pass

class demo(metaclass=meta\_class):

pass

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

* Both are used to generate classes whicha re automatically triggered at the end of the class statement.
* Decorators rebinds a class name to callable results whereas Metaclass route class creation through callable results.
* To manage class ,decorators simply augment and return original class objects , Metaclasses augment a class after they create it

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

* We can use both class decorator and metaclasses to manage class instances by inserting a wrapper object to catch the instance creation calls.
* Decorators may rebind the class name to a callabale run on instance creation that retains the original class object.
* Metaclasses can also do the same but they might have some disadvantage as they need to create a new object for that