1.

In object-oriented programming, a metaclass is a class whose instances are classes. Just as an ordinary class defines the behavior of certain objects, a metaclass defines the behavior of certain classes and their instances. Not all object-oriented programming languages support metaclasses.

2.

In order to set metaclass of a class, we use the __metaclass__ attribute. Metaclasses are used at the time the class is defined, so setting it explicitly after the class definition has no effect.

3.

Just like with metaclasses, because the decorator returns the original class, instances are made from it, not from a wrapper object. In fact, instance creation is not intercepted at all. ... Decorators can be used to manage both instances and classes, and they intersect with metaclasses in the second of these roles.

4.

Class attributes are the variables defined directly in the class that are shared by all objects of the class. Instance attributes are attributes or properties attached to an instance of a class. Just like with metaclasses, because the decorator returns the original class, instances are made from it, not from a wrapper object. In fact, instance creation is not intercepted at all. ... Decorators can be used to manage both instances and classes, and they intersect with metaclasses in the second of these roles.