Q1. What is the concept of a metaclass? ¶

```
A metaclass is a class that defines the behavior and structure of other classes.
In other words, it is a class that creates and controls other classes.
```

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

```
One of the best way is by inheritence
class BaseClass(metaclass=MyMeta):
    pass

class SubClass(BaseClass):
    pass

```

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

```
Class decorators: Class decorators operate on the class object after it is defined.
They can add, modify, or remove class attributes, methods, or even the entire class.
Metaclasses: Metaclasses have more control over the class creation process.
They are responsible for creating the class object and can customize various aspects,
including attribute creation, method binding, inheritance, and more.
```

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

In [ ]:

```
Class decorators can modify the __init__ method of a class,
which affects how instances are initialized.
Metaclasses have control over the creation process of instances. By overriding the __ca
you can customize how instances are created.
```