Q1. What is the relationship between classes and modules?

Solution - Modules are collections of methods and constants. They cannot generate instances. Classes may generate instances (objects), and have per-instance state (instance variables).

Q2. How do you make instances and classes?

Solution - To create instances of a class, you call the class using class name and pass in whatever arguments its __init__ method accepts, a class can be created by using the keyword class, followed by the class name.

Q3. Where and how should be class attributes created?

Solution - 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. Instance attributes are defined in the constructor. Defined directly inside a class.

Q4. Where and how are instance attributes created?

Solution – An instance attribute is a Python variable belonging to one, and only one, object. This variable is only accessible in the scope of this object, and it's defined inside the constructor function, __init__(self,..) of the class.

Q5. What does the term "self" in a Python class mean?

Solution - The self-parameter is a reference to the current instance of the class, and is used to access variables that belongs to the class.

Q6. How does a Python class handle operator overloading?

Solution - The operator overloading in Python means provide extended meaning beyond their predefined operational meaning. Such as, we use the "+" operator for adding two integers as well as joining two strings or merging two lists. We can achieve this as the "+" operator is overloaded by the "int" class and "str" class.

Q7. When do you consider allowing operator overloading of your classes?

Solution - When one or both operands are of a user-defined class or structure type, operator overloading makes it easier to specify user-defined implementation for such operations. This makes user-defined types more similar to the basic primitive data types in terms of behaviour.

Q8. What is the most popular form of operator overloading?

Solution - The most frequent instance is the adding up operator '+', where it can be used for the usual addition and also for combining two different strings.

Q9. What are the two most important concepts to grasp in order to comprehend Python OOP code?

Solution - Both inheritance and polymorphism are fundamental concepts of object oriented programming. These concepts help us to create code that can be extended and easily maintainable.