Q1. What is the relationship between classes and modules?

ANS:

A class is more of a unit, and a module is essentially a loose collection of stuff like functions, variables, or even classes.

Q2. How do you make instances and classes?

ANS:

 call the class using class name and pass in whatever arguments its \_\_init\_\_ method accepts.

Q3. Where and how should be class attributes created?

ANS:

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?

ANS:

Instance attributes are defined in the constructor. Defined directly inside a class. Defined inside a constructor using the self parameter. Shared across all objects.

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

ANS:

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?

ANS:

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?

ANS:

Ensures that objects of a class behave consistently with built-in types and other user-defined types. Makes it simpler to write code, especially for complex data types. Allows for code reuse by implementing one operator method and using it for other operators.

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

ANS:

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. As mentioned on top, the plus symbol's practice in dissimilar forms is the largest classic example of the operator level overloading process.

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

ANS:

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.