Q1. What is the relationship between classes and modules?

A class defines the blueprint of object creation, and modules are collection of instances that can be reused.

Q2. How do you make instances and classes?

To create instances of a class we need to call the class name and pass the acceptable arguments its \_\_init\_\_ accepts.

To create a class, new instances of an object are created allowing creation of new instances of that type.

Q3. Where and how should be class attributes created?

Class attributes are created directly in the class and shared by objects of the class.

Q4. Where and how are instance attributes created?

It is defined in the constructor, directly inside a class instance attributes are created using self -parameter.

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

It represents instance of a class, meaning we can access the attributes and methods of the class in python.

Q6. How does a Python class handle operator overloading?

It helps to give extended meaning beyond the pre-defined operational limits.

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

When the meaning of python operator is required to be changed operator overloading is accepted.

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

The \_add\_ operator also known as the magic function

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

The two most important concepts to grasp in order to comprehend python OOP code is Inheritance and Encapsulation.