**Q1. What is the relationship between classes and modules?**

**Answer:** 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?**

**Answer:** Instances are objects of a class in Python. In other words, users can define an instance of a particular class as an individual object.

Users can define the Instance methods inside a Python class, similar to how they define a regular function. For ex:

Class Employee:

def \_\_init\_\_(self, employee\_name, employee\_salary):

self.name = employee\_name

self.salary = employee\_salary

**Q3. Where and how should be class attributes created?**

**Answer:** Class attributes belong to the class itself they will be shared by all the instances. Such attributes are defined in the class body parts usually at the top, for legibility.

**Q4. Where and how are instance attributes created?**

**Answer:** Instance attributes are attributes or properties attached to an instance of a class. Instance attributes are defined in the constructor.

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

**Answer:** 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?**

**Answer:** In Python a feature that allows the same operator to have different meaning according to the context is called **operator overloading**. To handle this python have special functions defined by the Python interpreter They are called **"double underscore"** functions because they have a double underscore prefix and suffix, such as \_\_init\_\_() or \_\_add\_\_().

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

**Answer:** Let we have two objects which are the physical representation of a user-defined data type class. And we have to add two objects using the "+" operator, and it gives an error. This is because the compiler does not know how to add two objects. So, we will define the function for using the operator, and that process is known as "operator overloading".

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

**Answer:** The most popular 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?**

**Answer:** 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.