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

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

**Q2. How do you make instances and classes?**

Ans :- To create instances of a class, you 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.

**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:- 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.

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

Ans:- A very popular and convenient example is the Addition (+) operator. Just think how the '+' operator operates on two numbers and the same operator operates on two strings. It performs “Addition” on numbers whereas it performs “Concatenation” on strings.

**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