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

ANS)Modules can contain functions but also classes. Modules are imported using the import keyword. Classes, in the other hand, can be defined in your main application code or inside modules imported by your application.

**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.

The simplest class can be created using the class keyword.

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

ANS) 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?**

ANS) instance attributes are not shared by objects. Every object has its own copy of the instance attribute (In case of class attributes all object refer to single copy).

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

ANS) The self argument in Python refers to the object itself. Self is the name preferred by convention by Pythons to indicate the first parameter of instance methods in Python.

**Q6. How does a Python class handle operator overloading?**

ANS) To achieve operator overloading, we define a special method in a class definition. The name of the method should begin and end with a double underscore (\_\_). The + operator is overloaded using a special method named \_\_add\_\_() . This method is implemented by both the int and str classes.

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

ANS) Using a friend function to overload an operator is convenient because it gives you direct access to the internal members of the classes you're operating on.

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

ANS) + operator can be used to add two numbers, concatenate two strings or merge two lists.

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

ANS)Now, there are four fundamental concepts of Object-oriented programming – Inheritance, Encapsulation, Polymorphism, and Data abstraction. It is very important to know about all of these in order to understand OOPs