Q1. What is the relationship between classes and modules?

**Ans:** Modules are basically to modularise specific type of functionality and separate one chunk of activity with other, however class is an OOPs concept which can use all Oops functionality. We can say class is a template which have some common attributes and functionalities in form of modules.

Q2. How do you make instances and classes?

**Ans:** p1=MyClass(), here p1 is the instance of MyClass class.

Q3. Where and how should be class attributes created?

**Ans:** in \_\_init\_\_ method, using self.attributename.

Q4. Where and how are instance attributes created?

**Ans:** instance attributes can be created instance.attributename.

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

**Ans:** self parameter is a reference to the current instance of the class, and is used to access variables that belongs to class.

Q6. How does a Python class handle operator overloading?

**Ans:** by using magic methods, we can do operator overloading.

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

**Ans:** when you want to give special meaning to operator or want to extend functionality of operator.

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

**Ans:** most popular form of operator overloading is addition operator.

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

**Ans:** inheritance and encapsulation.