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 (instance variables).

Q2. How do you make instances and classes?

**Ans: Instance** : To create instances of a class, you call the class using class name and pass in whatever arguments its \_\_init\_\_ method accepts.

**Class:** 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.

Q4. Where and how are instance attributes created?

**Ans:** 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.

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:** Python operators work for built-in classes. But the same operator behaves differently with different types. For example, the + operator will perform arithmetic addition on two numbers, merge two lists, or concatenate two strings.

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

**Ans:** the + operator will perform arithmetic addition on two numbers, merge two lists, or concatenate two strings. We need this feature in Python when allows the same operator to have different meaning according to the context.

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

**Ans:** The most popular form 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:**Classes and objects are the two main aspects of object oriented programming.