Q1. What is the relationship between classes and modules?

**Ans:** A module in python is simply a way to organize the code, and it contains either python classes or just functions. If you need those classes or functions in your project, you just import them. For instance, the math module in python contains just a bunch of functions, and you just call those needed.

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:** A class attribute is a Python variable that belongs to a class rather than a particular object. It is shared between all the objects of this class and it is defined outside the constructor function, \_\_init\_\_(self,...) , of the class.

Q4. Where and how are instance attributes created?

**Ans:** An instance attribute is a Python variable belonging to only one object. It is only accessible in the scope of the object and it is defined inside the constructor function of a class. For example, \_\_init\_\_(self,..).

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

**Ans:** Self represents the instance of the class. By using the “self” keyword we can access the attributes and methods of the class in python.

Q6. How does a Python class handle operator overloading?

**Ans:** Python operators work for built-in classes. ... For example, the + operator will perform arithmetic addition on two numbers, merge two lists, or concatenate two strings. This feature in Python that allows the same operator to have different meaning according to the context is called operator overloading.

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

**Ans:** Python operators work for built-in classes. ... For example, the + operator will perform arithmetic addition on two numbers, merge two lists, or concatenate two strings. This feature in Python that allows the same operator to have different meaning according to the context is called operator overloading.

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

**Ans:** Operator Overloading means giving extended meaning beyond their predefined operational meaning. For example, operator + is used to add two integers as well as join two strings and merge two lists. It is achievable because ‘+’ operator is overloaded by int class and str class. You might have noticed that the same built-in operator or function shows different behaviour for objects of different classes, this is called Operator Overloading.

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

**Ans:** 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.