Q1. What is the relationship between classes and modules?

A class can be instantiated but a module cannot. A module will never be anything other than a library of methods.

Q2. How do you make instances and classes?

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?

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?

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?

Self parameter is a reference to current instance of class, and is used to access variables that belongs to class.

Q6. How does a Python class handle operator overloading?

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?

When we have two objects which are in physical representation.

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

Operator Overloading means giving extended meaning beyond their predefined operational meaning

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

Classes and objects are the two main aspects of object oriented programming. A class creates a new type of object where objects are particular instances of a particular class.

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