Q1. What is the relationship between classes and modules?

The difference between a class and a module in python is that a class is used to define a blueprint for a given object, whereas a module is used to reuse a given piece of code inside another program.

Q2. How do you make instances and classes?

Use the class name to create a new instance

Call ClassName() to create a new instance of the class ClassName . To pass parameters to the class instance, the class must have an \_\_init\_\_() method. Pass the parameters in the constructor of the class.

Q3. Where and how should be class attributes created?

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?

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

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

self represents the instance of the class. By using the “self” keyword we can access the attributes and methods of the class in python. It binds the attributes with the given arguments

Q6. How does a Python class handle operator overloading?

Class functions that begin with double underscore \_\_ called as special functions can handle operator overloading

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

Consider that we have two objects which are a physical representation of a class (user-defined data type) and we have to add two objects with binary ‘+’ operator it throws an error, because compiler don’t know how to add two objects. So we define a method for an operator and that process is called operator overloading

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

The most popular form of operator overloading is + operator, we will need to implement \_\_add\_\_() function in the class.

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

Inheritance and polymorphism