Q1. What is the relationship between classes and modules?

A class can be initiated but a module cannot be. A module will never be anything apart from than being a library consisting with various functions. In case we need to instantiate something or have it exist overtime we need to make use of a class.

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?

Class attributes are the owned itself by the class. It can be created with the help if any functions under the same class.

Q4. Where and how are instance attributes created?

Instance attributes are properties attached to an instance of a class. Instance attributes are defined in the constructor.

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

Self keyword is used to represent instance of the class. Using this we can access the attributes and methods of the class.

Q6. How does a Python class handle operator overloading?

This feature that allows the same operator for example ‘+’ to have different meaning according to the context is called operator overloading

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

If we have two objects from a class 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 implement this concept.

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

+ operator.

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

Inheritance and polymorphism.