Q1. What is the relationship between classes and modules?

Answer: Modules are the reusable code that can be imported in your current application which can classes and it’s objects as well.

Q2. How do you make instances and classes?

Answer:

Class class\_name:

Class related methods and variables and end of class

Obj = class\_name() # obj is an object/instance of that class.

Q3. Where and how should be class attributes created?

Answer: Class attributes should be created inside \_\_init\_\_() function or just within the class where it gets its value initialized when an object of that particular class is created along with some values.

Q4. Where and how are instance attributes created?

Answer: instance attributes are sent via parameters when an object is created for a particular class.

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

Answer: Using self, we can access methods and attributes of a class.

Q6. How does a Python class handle operator overloading?

Answer: operator overloading means giving an extended meaning beyond than operational meaning to a function.

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

Answer: We allow operator overloading when we need to perform some arithmetic operations on our objects created for classes.

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

Answer: \_\_add\_\_(),\_\_sub\_\_() are some of the popular forms of operator overloading.

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

Answer: The two most important concepts are namely classes and it’s objects.