***Q1. What is the relationship between classes and modules?***

***Ans.***Modules are collections of methods and constants. They cannot generate instances. Classes may generate instances (objects), and have per-instance state (instance variables).

***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.***To define a class attribute, you place it outside of the \_\_init\_\_() method. Use class\_name.class\_attribute or object\_name.class\_attribute to access the value of the class\_attribute.

***Q4. Where and how are instance attributes created?***

***Ans.***Instance attributes are attributes or 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?***

***Ans.***SELF represents the instance of class. This handy keyword allows you to access variables, attributes, and methods of a defined class in Python. The self parameter doesn't have to be named “self,” as you can call it by any other name.

***Q6. How does a Python class handle operator overloading?***

***Ans.***In Python, overloading is achieved by overriding the method which is specifically for that operator, in the user-defined class.

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

***Ans***.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?***

***Ans.***A very popular and convenient example is the Addition (+) operator.

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

***Ans.***Both inheritance and polymorphism are fundamental concepts of object oriented programming.