Q1. What is the relationship between classes and modules?

**Ans. A python module is nothing but a package to encapsulate reusable code.. Modules can contain functions but also classes. Modules are imported using the *import* keyword.**

**Classes, in the other hand, can be defined in your main application code or inside modules imported by your application. Classes are the code of Object Oriented Programming and can contain properties and methods.**

Q2. How do you make instances and classes?

**Ans. To create instances, you call the class using class name and pass in whatever arguments its \_\_init\_\_ method accepts. Where as to create a class, we use the keyword class.**

Q3. Where and how should be class attributes created?

**Ans. 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?

**Ans An instance attribute is a Python variable belonging to one, and only one, object. This variable is only accessible in the scope of this object and it is defined inside the constructor function, \_\_init\_\_(self,..) of the class.**

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

**Ans. The first argument of every class method, including init, is always a reference to the current instance of the class. By convention, this argument is always named self. In the init method, self refers to the newly created object; in other class methods, it refers to the instance whose method was called.**

Q6. How does a Python class handle operator overloading?

**Ans. To achieve operator overloading, we define a special method in a class definition. The name of the method should begin and end with a double underscore (\_\_). The + operator is overloaded using a special method named \_\_add\_\_(). This method is implemented by both the int and str classes.**

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

**Ans. If an overloaded operator is called for a certain class object in the program, then the method is automatically called that implements the actions of this operator. Thus, when the operator + (addition) is overloaded, any actions can be programmed in the method that implements it.**

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

**Ans. The most common example in Python is the plus sign, a binary (i.e., two operands) operator that not only adds a pair of numbers, but also concatenates a pair of lists or strings.**

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

**Ans. Inheritance and Polymorphism.**