Q1. What is the relationship between classes and modules?

Classes will generate instances(objects) and module cannot be instantiated.

Modules are files present in the package and class is used to encapsulate data and functions together inside the same unit.

Q2. How do you make instances and classes?

We can use class keyword to create class

class test:

we can create instance of the class as below:

testobj = test()

Q3. Where and how should be class attributes created?

Class attributes are attributes which are owned by the class itself. They will be shared by all the instances of the class.

Class attributes will be defined outside all the methods usually at the top, right below the class header  
for the class test we can create class attribute ‘a’ as below:

class test:

a=’this is class variable’

classobj=test()

classobj.a

Q4. Where and how are instance attributes created?

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

class test:

def \_\_init\_\_(self,a):

self.a=a #instance attribute

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

Self means the current instance of the class

Q6. How does a Python class handle operator overloading?

Python provides some special functions or magic functions that is automatically invoked when it is associated with that operator to handle operator overloading

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

When we want to allow the same operator to have different meaning according to the context

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

Binary "+" Operator is most popular form of operator overloading

we use the "+" operator for adding two integers as well as joining two strings or merging two lists. We can achieve this as the "+" operator is overloaded by the "int" class and "str" class.

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

Inheritance and polymorphism which helps to create code that can be extended and easily maintainable.