Q1. What is the relationship between classes and modules?

Classes and modules both help in reusing the code with the help of the methods defined in them.

Q2. How do you make instances and classes?

We make classes by use of keyword “**class**” and we make instance by use of any name / word except the ones that are reserved keywords.

For defining a class:

class <name\_of\_class>(#parameters):

// class methods

For defining of instance of class:

<name\_of\_variable>=<class\_name>(#parameters)

Q3. Where and how should be class attributes created?

Class attributes should be created inside the class using functions.

Q4. Where and how are instance attributes created?

Instance of classes are created outside of the class by simply using any variable name.

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

“Self” is instance of the class. With the help of “self” we can access methods and attributes of the class

Q6. How does a Python class handle operator overloading?

Python provides some special function that is automatically invoked when it is associated with that operator to handle the operator overloading.

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

We generally consider when we need to have multiple tasks from the same operator name.

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

The most popular form in python is Addition (+)

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

The two major concepts are: Inheritance and Polymorphism.

Inheritance is creating the child classes

Polymorphism means same function for performing different operations.