Q1. What is the relationship between classes and modules?

Ans. Classes are related to modules. When we create a class and save it as .py file, then we have actually created a module. So we can call it by using simple import keyword following the file name where class was created. Then we can use the class.

Q2. How do you make instances and classes?

Ans . instance\_name = class\_name()

Q3. Where and how should be class attributes created?

And. Class attributes are created inside class and it isn't copied in each instance.

def class\_name():

att1 = 0

att2 = 'ABC'

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. A self act as a pointer to itself. Meaning it pointing to the instance/object, of a class created, binding with the variable which has been initialize and are sending to the class by using **init**() function

Q6. How does a Python class handle operator overloading?

Ans. When we use an operator on user defined data types then automatically a special function or magic function associated with that operator is invoked. Changing the behavior of operator is as simple as changing the behavior of method or function When we use + operator, the magic method \_\_add\_\_ is automatically invoked in

which the operation for + operator is defined. There by changing this magic method’s code, we can give extra meaning to the + operator.

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 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. The most popular one the Addition operator (+),  because it performs Addition on numbers whereas it performs Concatenation on strings.

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

Ans. Inheritence: It means inheriting the properties and behaviour of the parent class to the child class and its attributes

Polymorphism: It means many form. Under the class a function can take many form with the same name. It only differ from others by its parameter or the way it is being operate. Function overloading and overriding comes under this concept.