Q1. What is the relationship between classes and modules?

Classes are used to define a blueprint for an object and can have instance, they can inherit or some other classes can inherit from them, while modules are a group of constants and functions which store in a simple file, we cannot define any instance for them and they are called through import when we need to reuse those functions and constants.

Q2. How do you make instances and classes?

We can call a class with its name and its input variables and save this object into a variable which will be instance of that class.

Q3. Where and how should be class attributes created?

Class attributes can define directly inside the class and will share across all objects, we can call them with class name of object name and if we change its value it will reflect to all object of that class

Q4. Where and how are instance attributes created?

Instance attributes are defined inside the constructer or \_\_init\_\_ method with the help of ‘self’ and can be used by object name. if we change their value, it will only change for related object not all the objects.

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

“self” is a reference to the current instance object and when a instance call an attribute or method of the class, its name will send to the class through ‘self’

Q6. How does a Python class handle operator overloading?

There is some magic function in python associated to each operator, if we redefine those operators through related magic function (for example for + we have \_\_add\_\_ function or – has \_\_sub\_\_ function) we can add some new ability to that specific operator, this approach call operator overloading.

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

Operator overloading can give simplicity to your code and makes it user-friendly, because when user uses known operator, it will work without error same as when we use them for normal variables. It can be use in any class and for this we have to redefine all the operator through their magic functions based on the needs of the class.

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

The most popular operator overloading is for + operator and after that \* , <= >= and == are the most operators which overload inside the classes.

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

Inheritance and polymorphism are two most important concepts in OOP programming. They are useful in case of extend, modification and troubleshoot of the code.