**Q1. What is the relationship between classes and modules?**

Classes are the codes which can mapped with a real-world object, they have properties and functions same as real word objects do. We create instances of these objects to assign values to the properties specific to that instances and carry out certain operations.

Whereas modules are the python programs which can contain classes and can be imported and used into other program files. These are easy to manage and reuse.

**Q2. How do you make instances and classes?**

We have to define a class first to create instances in which we define attributes and methods of the class. Then we call the class just like we call the function and create instances.

**Q3. Where and how should be class attributes created?**

Class attributes in Python should be created directly within the class definition, outside of any methods. They are defined at the class level and are shared among all instances of the class.

**Q4. Where and how are instance attributes created?**

We define instance attributes in \_\_init\_\_ method and we assign values to those attributes when we instantiate the class. Some times attributes are also defined and initiated inside methods if the attribute name is preceded by ‘self’ keyword and dot then it can also be called as instance attribute.

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

Attributes and methods which are attached to “self” term tells us that they belong to the instance of the class. . The methods defined in the class can be used in another method only if it is attached with self keyword.They will be specific to the instance.

**Q6. How does a Python class handle operator overloading?**

We can use magic methods for operator overloading in python classes which can change the behaviour of the operators for examples \_\_add\_\_ (for ‘ + ‘), \_\_sub\_\_ (for ‘ - ‘), \_\_mul\_\_ (for ‘ \* ‘) etc.

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

When we create custom data types or classes which can be mapped into real world objects they are expected to behave in certain way. We will use operator overloading to define such behaviours. For example in case of vectors, matrices and signals addition has different expected behaviour normal addition will not work in these cases.

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

Arithmetic operator overloading are the most popular form of operator overloading like +, -, / and \*

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

1. Defining simple classes with attributes and methods and how it relates to real world is important concept to grasp in order to comprehend OOP code. This will give a basic idea about objects or instances of the class and how a single template can be reused multiple times.
2. Advance concepts like inheritance and polymorphism