Q1. What is the relationship between classes and modules?

Classes are part of oops concept, we can instantiate a class but modules cannot be done, modules can be reused any number of times

Q2. How do you make instances and classes?

Class emp():

Def \_\_init\_\_\_(self,……..)

Instance creation: emp\_inst= emp(pass the values for arguments here)

Q3. Where and how should be class attributes created?

Class attributes are created after the class creation

Class emp():

Def \_\_init\_\_\_(self, emp\_id,salary)

Q4. Where and how are instance attributes created?

Instance attributes are created in the constructor

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

It represents the instance of the class

Q6. How does a Python class handle operator overloading?

different behaviour of a single operator for different types of operands is called Operator Overloading

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

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

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

Inheritance, encapsulation