Q1. Define the relationship between a class and its instances. Is it a one-to-one or a one-to-many partnership, for example?

Ans=>class is a template where we define some methods related to that specific class and instance is an object of that class where we can just call all the methods inside that class.

Q2. What kind of data is held only in an instance?

Ans=>methods and variables stored in an instance of class

Q3. What kind of knowledge is stored in a class?

Ans=> class contains all the methods its values and variables etc.

Q4. What exactly is a method, and how is it different from a regular function?

Ans => method is same as a regular function but it only belongs to some class and it has its self attribute which represents its properties.

Q5. Is inheritance supported in Python, and if so, what is the syntax?

Ans=>yes inheritance supported in python

class Derived(Base1, Base2)

where Base1 and Base2 are different classes.

Q6. How much encapsulation (making instance or class variables private) does Python support?

Ans=>yes python supports encapsulation

Q7. How do you distinguish between a class variable and an instance variable?

Ans =>variable which defined inside class is class variable and instance variable is attached when instance is called.

Q8. When, if ever, can self be included in a class's method definitions?

Ans=>yes its always included in class init method but not in class def method.

Q9. What is the difference between the \_ \_add\_ \_ and the \_ \_radd\_ \_ methods?

Ans=>\_\_radd\_\_ is only called if the left object does not have an \_\_add\_\_ method, or that method does not know how to add the two objects (which it flags by returning NotImplemented). Both classes have an \_\_add\_\_ method, which do not return NotImplemented. Therefore the \_\_radd\_\_ method would never be called

we can control the result of a sum of two objects by modifying or defying the \_\_add\_\_ method. We can define the \_\_add\_\_ method to return a Day instance with the total number of visits and contacts

Q10. When is it necessary to use a reflection method? When do you not need it, even though you support the operation in question?

Ans=>Reflection refers to the ability for code to be able to examine attributes about objects that might be passed as parameters to a function

Q11. What is the \_ \_iadd\_ \_ method called?

Ans=>This function is used to assign and add the current value

Q12. Is the \_ \_init\_ \_ method inherited by subclasses? What do you do if you need to customize its behavior within a subclass?

Ans=>yes it is, we can redefine method in our class to change its behaviour