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

The relationship between class and its instances is one to many as a class can have many instances whereas a relationship between a instance and class is one to one as a instance can only belong to one class.

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

Instance can hold the data for the variables in the defined in the parent class

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

Class has knowledge of the program/functions to be performed on the variables

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

Method is used by applying dot operator. It is different from regular function as regular function is only applied using the parenthesis.

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

Yes inheritance is supported in python language and syntax is below:

Class <child\_class\_name>(<parent\_class\_name>):

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

Pyhton supports encapsulation to the extend that the variables that are encapsulated can be only accessed within the class in which they are encapsulated.

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

A class variable usually does not hold any value(s) whereas the instance variables are derived from the class variables are only values to the class variables.

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

Self is included if we want to define any functions/ assigning the values to the variables in the class

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

\_\_add\_\_ adds the myobject with the number but it can only add the number if the myobject is in left place and the number to be added is in right. If we shift their positions, then it will give a error but if instead of using \_\_add\_\_ we use \_\_radd\_\_ then we will get same output as before we were adding through \_\_add\_\_

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

Reflection refers to the ability of code to be able to examine the attributes of the objects that might be passed as parameters to a function. We need reflection in case of recursion.

We don’t need it we are already having the output as string and we are using reflection to get the output as string. This will return a empty string.

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

\_\_iadd\_\_ method is called when we need to add the number with other number and saves the sum to the number.

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

Yes the \_\_init\_\_ method is inherited from the parent class but if we want to customized \_\_init\_\_ method then we can easily do so by defining it in the sub class