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

A class is a set of entities which are better known as instances. It is a one-to-one partnership

# `a` and `b` are distinct instances of the list class/type

# even though they contain the same sequence of integers

>>> a = list((1, 2, 3))

>>> b = list((1, 2, 3))

>>> a is b

False

>>> isinstance(a, list)

True

>>> isinstance(b, list)

True

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

Classes and variables are only held in an instance.

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

Class stores the data/values of objects.

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

Both method and regular functions are called by name; Method is a piece of code and associated with an object, whereas regular function is not.

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

Yes, its the ability of one class to replicate the properties from another class

\_\_init\_\_

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

Encapsulation works on class variables by wrapping them and declaring them private.

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

Class variable has the same value for all instances of the class and instance variable has different values for all instances of a class.

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

When there is need to access the class, then class method helps to create instances of such class.

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

When two objects have same method \_\_add\_\_ method is used and if not then \_\_radd\_\_ is used to check if values can be added

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

It is used to support concatenation and slicing for strings, lists and sequences by writing a recursive reverse function.

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

The \_ \_iadd\_ \_ method is called Emulating numeric type

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

Yes, to customize instance variable add double underscores before its name.