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: A class is a set of entities, which are called the instances of the class. An entity can be an instance of many classes, which are called its types, and a class can be a type of many classes.

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

Ans: Instance variables are created when an object is instantiated, and are accessible to all the constructors, methods, or blocks in the class. ... It is a type of class attribute (or class property, field, or data member).

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

Ans: Reasoning with the stored knowledge, in the sense of classical logic, is the process of inferring conclusions from the given premises. Thus, the efficiency of the selected reasoning method depends strongly on the method used for knowledge representation.

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

Ans: A function is a piece of code that is called by name. It can be passed data to operate on (i.e. the parameters) and can optionally return data (the return value). All data that is passed to a function is explicitly passed. A method is a piece of code that is called by a name that is associated with an object.

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

Yes inheritance supported in python

Class A:

Pass

Class B(A):

Pass

Class C(B):

pass

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

Ans: Encapsulation. Using OOP in Python, we can restrict access to methods and variables. This prevents data from direct modification which is called encapsulation. In Python, we denote private attributes using underscore as the prefix i.e single \_ or double \_\_ .

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

Ans: Instance Variable usually reserves memory for data that the class needs. Class Variable maintains a single shared value for all instances of class even if no instance object of the class exists.

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

Ans: Self is pointer variable which is used to point the class global variable in python

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

Ans: There is no difference.

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: The name reflection is used to describe code which is able to inspect other code in the same system (or itself).

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

Ans: This function is used to assign and add the current value. This operation does “a+=b” operation. Assigning is not performed in case of immutable containers, such as strings, numbers and tuples.

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

Ans: init is constructor in python. We can’t inherited by subclass