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

Difference Between Class And Object:

Class Object

Class is used as a template for declaring and creating the objects. An object is an instance of a class.

When a class is created, no memory is allocated. Objects are allocated memory space whenever they are created.

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

Ans = An instance variable is a variable which is declared in a class but outside of constructors, methods, or blocks. Instance variables are created when an object is instantiated, and are accessible to all the constructors, methods, or blocks in the class.

What kind of knowledge is stored in a class?

Q3. Explicit knowledge.

Implicit knowledge.

Tacit knowledge.

Procedural knowledge.

Declarative knowledge.

A Posteriori knowledge.

A Priori knowledge

Q4. A function doesn't need any object and is independent, while the method is a function, which is linked with any object. We can directly call the function with its name, while the method is called by the object's name. Function is used to pass or return the data, while the method operates the data in a class.

Q5. Is inherPython even supports Hybrid inheritance in which we implement more than one type of inheritance in one code. In the above code, we have implemented more than one type of inheritance. Classes a, b, and c implement hierarchical inheritance. On the other hand, classes a, c, and d implement multi-level inheritance.itance supported in Python, and if so, what is the syntax?

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

Ans = Encapsulation is one of the fundamental concepts in object-oriented programming (OOP). It describes the idea of wrapping data and the methods that work on data within one unit. This puts restrictions on accessing variables and methods directly and can prevent the accidental modification of data. To prevent accidental change, an object’s variable can only be changed by an object’s method. Those types of variables are known as private variables.

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

It is a variable whose value is instance-specific and now shared among instances. It is a variable that defines a specific attribute or property for a class.

These variables cannot be shared between classes . Instead, they only belong to one specific class. These variables can be shared between class and its subclasses.

It usually reserves memory for data that the class needs. It usually 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?

self represents the instance of the class. By using the “self” we can access the attributes and methods of the class in python. It binds the attributes with the given arguments.

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

self represents the instance of the class. By using the “self” we can access the attributes and methods of the class in python. It binds the attributes with the given arguments.

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

You can use reflection to dynamically create an instance of a type, bind the type to an existing object, or get the type from an existing object and invoke its methods or access its fields and properties. If you are using attributes in your code, reflection enables you to access them.

Q11e Python \_\_iadd\_\_() magic method implements in-place addition x += y that adds together the operands and assigns the result to the left operand. This operation is also called augmented arithmetic assignment. The method simply returns the new value to be assigned to the first operand.

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

The \_\_init\_\_ method is the Python equivalent of the C++ constructor in an object-oriented approach. The \_\_init\_\_ function is called every time an object is created from a class. The \_\_init\_\_ method lets the class initialize the object's attributes and serves no other purpose. It is only used within classes.