Q1. What is the purpose of Python's OOP?

In Python, object-oriented Programming (OOPs) is a programming paradigm that uses objects and classes in programming. It aims to implement real-world entities like inheritance, polymorphisms, encapsulation, etc. in the programming.

Q2. Where does an inheritance search look for an attribute?

The whole point of a namespace tool like the class statement is to support name inheritance. In Python, inheritance happens when an object is qualified, and involves searching an attribute definition tree (one or more namespaces). Every time you use an expression of the form object.

Q3. How do you distinguish between a class object and an instance object?

A class is a template for creating objects in a program, whereas the object is an instance of a class. A class is a logical entity, while an object is a physical entity. A class does not allocate memory space; on the other hand, an object allocates memory space.

Q4. What makes the first argument in a class’s method function special?

The calling process is automatic while the receiving process is not (its explicit). This is the reason the first parameter of a function in class must be the object itself. Writing this parameter as self is merely a convention. It is not a keyword and has no special meaning in Python.

Q5. What is the purpose of the \_\_init\_\_ method?

The \_\_init\_\_ method lets the class initialize the object's attributes and serves no other purpose. It is only used within classes.

Q6. What is the process for creating a class instance?

The new operator requires a single, postfix argument: a call to a constructor. The name of the constructor provides the name of the class to instantiate. The constructor initializes the new object. The new operator returns a reference to the object it created.

Q7. What is the process for creating a class?

A Class is like an object constructor, or a "blueprint" for creating objects.

Create a Class. To create a class, use the keyword class : ...

Create Object. Now we can use the class named MyClass to create objects: ...

The self Parameter. ...

Modify Object Properties. ...

Delete Object Properties. ...

Delete Objects.

Q8. How would you define the superclasses of a class?

A superclass is the class from which many subclasses can be created. The subclasses inherit the characteristics of a superclass. The superclass is also known as the parent class or base class. In the above example, Vehicle is the Superclass and its subclasses are Car, Truck and Motorcycle