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

Ans:

Object Oriented Programming is a type of programming that is based on objects rather than just functions and procedures. Individual objects are grouped into classes. In Python, OOPs implements real-world entities like inheritance, polymorphism, encapsulation into programming.

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

Ans:

An inheritance search looks for an attribute first in the class the instance was created from, then in all higher super classes from left to right.

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

Ans:

Everything in Python is an object. Every object has a type of any class. Instance is an object that is created using a class. While a class is just a blueprint and does not take physical space in memory. While any instance will take physical space.

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

Ans:

Every method that we define in a class accepts the instance of the class as its first argument. The “self” variable points to the instance of the class that we are working with. when an instance method is called from an instance object, that instance object is automatically passed as the first argument to the method.

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

Ans:

The \_\_init\_\_ method lets the class initialize the object's attributes. It is called every time an object is created from a class.

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

Ans:

To create instances of a class, we can call the class using class name and pass in whatever arguments in it’s \_\_init\_\_ method.

Q7. What is the process for creating a class?

Ans:

We can create class using class keyword followed by the class name.

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

Ans:

The class from which a class inherits is called the parent or superclass. A class which inherits from a superclass is called a subclass.