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

Dilated convolutions, also known as atrous convolutions, are a type of convolutional operation that allows for an increased receptive field without increasing the number of parameters or the computational cost. Dilated convolutions introduce gaps or holes between the kernel elements, allowing them to sample input data at a larger stride or spacin

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

The purpose of Python's Object-Oriented Programming (OOP) is to organize and structure code in a way that promotes modularity, reusability, and maintainability. OOP allows you to define classes and objects, which encapsulate data (attributes) and functionality (methods) into coherent units

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

Class Object:

A class object represents the blueprint or definition of a class

It is created when the class itself is defined.

Instance Object:

An instance object represents a specific instance or occurrence of a class.

It is created using the class constructor or by calling the class as a function.

the class object represents the overall class definition and holds the class-level attributes and methods, while the instance objects represent specific instances of the class with their own unique set of attributes and data. Class objects are used to create and access instances, whereas instance objects are used to work with specific instances and their specific attributes or data

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

In Python, the first argument in a class's method function is conventionally named self (though any valid variable name can be used). This first argument is special because it refers to the instance object itself when the method is called. It allows the method to access and manipulate the attributes and behavior specific to that instance.

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

The \_\_init\_\_ method, also known as the constructor method, is a special method in Python classes that is automatically called when an object of the class is created. Its purpose is to initialize the object's attributes and perform any necessary setup or configuration.

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

The process for creating a class instance involves the following steps

Class Definition:

Class Instantiation:

Instance Initialization:

Accessing Attributes and Methods

By following these steps, you can create instances of a class and work with the specific objects that they represent, accessing their attributes and invoking their methods as needed.

Q7. What is the process for creating a class?

Class Definition:

Class Instantiation:

Instance Initialization:

Accessing Attributes and Methods

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

The superclasses of a class are the classes from which the current class directly inherits. In Python, you can define the superclasses of a class by specifying them in parentheses after the class name when defining the class. This is known as class inheritance or subclassing.