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

**Answer 1:** The purpose of Python's OOP (Object-Oriented Programming) is to organize code into reusable and structured components called objects, facilitating code modularity, encapsulation, and abstraction.

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

**Answer 2:** An inheritance search looks for an attribute first in the instance's class, then in its parent classes (superclasses) following the order.

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

**Answer 3:** A class object is a blueprint for creating instances, while an instance object is a specific occurrence of that class, representing a unique entity.

For example:

Class Dog:

def \_\_init\_\_(self,name):

self.name = name

def bark(self):

print(f”{self.name} is barking!”)

dog1 = Dog(“Tommy”)

print(dog1)

In the above example, class object is Dog and instance object is dog1 to call the class object.

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

**Answer 4:** The first argument in a class's method function, conventionally named 'self,' refers to the instance calling the method, allowing access to instance-specific attributes and methods.

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

**Answer 5:** The \_init\_ method is a special method in Python classes used for initializing instance attributes when an object is created.

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

**Answer 6:** To create a class instance, you instantiate the class by calling its constructor, typically using the class name followed by parentheses, e.g., obj = ClassName().

1. Class instance holds unique data specific to that object.
2. Class instance can invoke methods defining in the class, performing actions specific to that instance.
3. Class instances allow you to reuse code from the class blueprint.

**Q7. What is the process for creating a class?**

**Answer 7:** To create a class, you define it using the class keyword, specifying attributes and methods within the class body. Instances are then created based on this class.

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

**Answer 8:** The superclasses of a class are the classes from which it inherits. These are the classes mentioned in the parentheses following the class name during its definition.