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

Ans: OOPs aims to implement real-world entities like inheritance, hiding, polymorphism etc. in programming. The main aim of OOPs is to bind the data together into a function, so that no other part to code can able to access it except that function.

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

Ans: 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.attribute where object is an instance or class object, Python searches the namespace tree at and above object, for the first attr it can find. Because lower definitions in the tree override higher ones, inheritance forms the basis of specialization.

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

Ans: when we create a class in python then a class object is created so whenever python finds a class statement in the whole program then it creates a class object and assigns a name to that object i.e. class name.

Ex: class MyClass:

pass

when we call a class, it creates an instance object of that class from which the object has been created.

Ex: Obj1=MyClass()

class object is like a blueprint for instance object but instance object is a concrete item in out code.

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

Ans: 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.

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

Ans: \_\_init\_\_ method is a constructor for a class and lets the class initialize the object's attributes and serves no other purpose. The \_\_init\_\_ function is called every time an object is created from a class.

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

Ans: Calling ClassName() to create a new instance of the class ClassName . To pass parameters to the class instance, the class must have an \_\_init\_\_() method. Pass the parameters in the constructor of the class.

Ex: class ClassName:

pass

object=ClassName()

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

Ans: For creating a class, we have to mention class followed by ClassName. And we can use use \_\_init\_\_method, user-defined and predefined methods inside a class.

Ex: class ClassName:

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

pass

object=ClassName()

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

Ans: 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.