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

Ans: Object-oriented programming is a programming paradigm that provides a means of structuring programs so that properties and behaviours are bundled into individual objects. The oops concept focuses on writing the reusable code.

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

Ans: inheritance search look in its super(parent) class on most super class for an attribute.

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

Ans: Class objects provide default behaviour and serve as factories for instance objects.

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

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

Ans: Self is the first argument to be passed in Constructor and Instance Method. Self must be provided as a First parameter to the Instance. The self is used to represent the instance of the class.

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

Ans. This method is used to initializing the property of class at initial phase.

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

Ans. Class name with (). Example obj = classname()

Q7. What is the process for creating a class?

Ans. Classname with keyword ‘class’ Example: class classname.

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

Ans. Classname with keyword ‘class’.

class Parent:  
  def \_\_init\_\_(self, txt):  
    self.message = txt  
  
  def printmessage(self):  
    print(self.message)  
  
class Child(Parent):  
  def \_\_init\_\_(self, txt):  
    super().\_\_init\_\_(txt)  
  
x = Child("Hello, and welcome!")  
x.printmessage()