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

Object-oriented programming: As the name suggests, Object-Oriented Programming or OOPs refers to languages that uses objects in programming. Object-oriented programming aims to implement real-world entities like inheritance, hiding, polymorphism etc in programming. The main aim of OOP is to bind together the data and the functions that operate on them so that no other part of the code can access this data except that function.

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

The inheritance search is simply a search of the tree from bottom to top looking for the lowest occurrence of an attribute name.

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

Class object : A compound data type that is often used to model a thing or concept in the real world. It bundles together the data and the operations that are relevant for that kind of data. Instance and object are used interchangeably.

Instance Object : An object whose type is of some class.

An instance is a specific representation of an object. An object is a generic thing while an instance is a single object that has been created in memory

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

In object-oriented programming, whenever we define methods for a class, we use self as the first parameter in each case.

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

"init" is a reseved method in python classes. It is called as a constructor in object oriented terminology. This method is called when an object is created from a class and it allows the class to initialize the attributes of the class.

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

When you create an object, you are creating an instance of a class, therefore "instantiating" a class.

class MyClass:

a = 1

ab = MyClass()

print(ab.a)

Q7. What is the process for creating a class?

class definitions begin with a class keyword.

class demo:

'''This is the test class'''

pass

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

The super() function returns a parent class object and can be used to access the attributes or methods of the parent class inside the child class.