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

It aims to implement real-world entities like inheritance, polymorphisms, encapsulation, etc. in the programming.

It binds the data and the functions that work together as a single unit so that no other part of the code can access it.

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

The whole point of a namespace tool like the class statement is to support name inheritance. For example, in 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.attr where object is an instance or class object, Python searches the namespace tree at and above object, for the first attr it can find.

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

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.

Instance objects are real objects in your python code process. The instance object has access to attributes of the class from which it is created

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

“self” represents the instance of the class. By using the “self” keyword we can access the attributes and methods of the class in python. It binds the attributes with the given arguments

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

The \_\_init\_\_ method is at the core of OOP and is required to create objects.

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

Call ClassName() to create a new instance of the class ClassName

Q7. What is the process for creating a class?

* To create a class, use the keyword class:

class MyClass:  
  x = 5

* use the class named MyClass to create objects:

p1 = MyClass()  
print(p1.x)

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

The class from which a class inherits is called the parent or superclass. It can be referred to with the use of the super() function.