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

Everything in Python is an object and every object has a type. These types are declared and defined using classes. Thus, classes can be considered as the heart of OOP. In order to develop robust and well-designed software products with Python, it is essential to obtain a comprehensive understanding of OOP.

Multiple instances of an object can be made.

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

 Python searches for an attribute in an upward tree of attributes. it first searches for the attribute in its instance and then looks in the class it is generated from, to all super classes listed in its class header

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

Class is a template for creating objects whereas object is an instance of class

Seperate memory is allocated for each object whenever an object is created. but for a class this doesnot happens.

A Class is created once. Many objects are created using a class.

As Classes have no allocated memory. they can't be manipulated. but objects can be manipulated.

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

The first argument of every class method, including init, is always a reference to the current instance of the class. By convention, this argument is always named self. In the init method, self refers to the newly created object; in other class methods, it refers to the instance whose method was called.

* Instance Methods (object level methods)
* Class Methods (class level methods)
* Static Methods (general utility methods)
* **self** is the first argument for instance methods. which refers to the object itself
* **cls** is the first argument for class methods which refers to the class itself

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

It serves the role of 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?

To create a class instance, we need to call the class by its name and pass the arguments to the class, which its \_\_**init\_\_** method accepts.

Eg: obj=Course(python)

Obj-🡪 object reference(class instance),course---class & python-🡪 attributes

Q7. What is the process for creating a class?

**class** keyword is used to created a class in python. The syntax to create a class in python is

**class <classname>:**

**class FSDS:** ➞ this creates a class called FSDS

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

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. In the below example, ineuron is the Superclass and its subclasses are FSDS & JAVA

Eg: class ineuron:

Pass

class FSDS(ineuron):

Print(‘Child class 1 of ineuron’)

class Java(ineuron):

Print(‘Child class 2 of ineuron’)