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

Ans🡺

1. Provides a clear program structure, which makes it easy to map real world problems and their solutions.
2. Facilitates easy maintenance and modification of existing code.
3. Enhances program modularity because each object exists independently and new features can be added easily without disturbing the existing ones.
4. Presents a good framework for code libraries where supplied components can be easily adapted and modified by the programmer.
5. Imparts code reusability

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

Ans🡺

Python used the inheritance attribute lookup to find \_\_init\_\_ in animal class.

When we created the child class, first it will look the \_\_init\_\_ method in the dog class, then it didn’t find it then looked into parent class and found there and called that there.

So as our class design became complex we may wish to initialize a instance firstly processing it through parent class constructor and then through child class constructor.

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

A blueprint for a house design is like a class description. All the houses built from that blueprint are objects of that class. A given house is an instance.

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

Ans🡺A self keyword

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

Ans🡺  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?

class MyClass(object):

pass

# Create first instance of MyClass

this\_obj = MyClass()

print(this\_obj)

Q7. What is the process for creating a class?

Ans🡺

The class has a documentation string, which can be accessed via ClassName.\_\_doc\_\_.

The class\_suite consists of all the component statements defining class members, data attributes and functions.

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

Ans🡺

class Child(Parent):

//sample code