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

* Object oriented programming helps us is structure programming so that properties and behaviour are somewhat related to an object.
* OOPs uses classes and objects in programming which helps us in creating a big program into smaller bit
* It helps us in reusing the code for creating various instances of an object
* It has properties such as polymorphism, inheritance and encapsulation

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

Ans  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 superclasses listed in its class header

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

* Classes are used for creating objects whereas Objects are an instance of classes
* Classes are created once but many objects can be created using the same classes
* Separate memory is allocated to an object once it is created whereas that is not the case for a class

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

Ans . The ‘self’ is received as the first argument of the class method function which refers to an instance of the object of creating.

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

Ans **\_\_init\_\_** is a reserved method in python classes. 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?

Ans. First we can assign a name to the instance of our class and call the class by its name and pass the argument to the class eg Any\_name= Name(argument)

Q7. What is the process for creating a class?

Ans. Using the keyword class followed by the name of the class and colin at the end, eg class class\_name:

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

Ans Super or Parent class is given as an argument to the child class

Eg

class Person(object):

Pass

class Employee(Person):

pass

here employee inherits the properties of a Person class which is the Parent class