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

Ans: Object-oriented programming is a programming paradigm that provides a means of structuring programs so that properties and behaviors are bundled into individual objects

- In Python, object-oriented Programming (OOPs) uses objects and classes in programming.
- It aims to implement real-world entities like inheritance, polymorphisms, encapsulation, etc. in the programming.
- The main concept of OOPs is to bind the data and the functions that work on that together as a single unit so that no other part of the code can access this data.
- It comes up with the following advantages:
- It helps to divide our over all program into different small segments and thus making it solving easy with the use of objects
- Helps in easy maintenance and modification of existing program
- Multiple instances of an object can be made.

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 super classes listed in its class header

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

Ans: The differences between a class object and an instance object are:

1. Class is a template for creating objects whereas object is an instance of class
2. Seperate memory is allocated for each object whenever an object is created. but for a class this doesnot happens.
3. A Class is created once. Many objects are created using a class.
4. 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?

Ans: Python Classes usually have three types of methods which are:

- 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?

Ans: __init__ is a reseved 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: 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.
Example: my_name = my_class("Mano","vishnu") Here my_name is an instance of class my_class with attributes "Mano" and "Vishnu".

Q7. What is the process for creating a class?

Ans: class keyword is used to created a class in python. The syntax to create a class in python is **class <classname>:**

Example: class Car: → this creates a class called Car

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

Ans Superclass/Parent class is given as a arugment to the child class

Example: class Employee(Person): Here child class Employee inherits attributes and methofs from Superclass/Parent Person