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

* 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?**

 An inheritance search looks for an attribute **first in the instance object, then in the class the instance was created from, then in all higher superclasses, progressing from left to right (by default)**. The search stops at the first place the attribute is found.

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

**Object is an instance of a class**. All data members and member functions of the class can be accessed with the help of objects. When a class is defined, no memory is allocated, but memory is allocated when it is instantiated (i.e. an object is created).

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

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?**

"\_\_init\_\_" is a reseved method in python classes. 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?**

 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("anu","pranu")** Here my\_name is an instance of class my\_class with attributes “anu" and "pranu".

**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>:**

**Example:** **class Car:** ➞ this creates a class called Car

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

**The class from which a class inherits** is called the parent or superclass. A class which inherits from a superclass is called a subclass, also called heir class or child class. Superclasses are sometimes called ancestors as well.