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

Ans: Which provides more security with classes and Objects. Code modularity, reusability and readability can achieve. Also can achieve encapsulation, inheritance, and polymorphism.

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

Ans:

The inheritance search for an attribute in Python looks for the attribute in the following order:

1. The instance of the object itself

2. The class of the instance

3. The base classes of the class, in the order determined by the MRO algorithm.

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

Ans:

Class Object:

1. A class object is created when a class is defined.

2. It is used to define the properties and methods of a class.

3. It is essentially a blueprint for creating instances of the class.

4. It is usually defined with the "class" keyword and can have attributes such as variables and methods.

Instance Object:

1. An instance object is created when a class is instantiated.

2. It is an actual object that has its own set of data and can call the methods of the class.

3. It is created using the class name followed by parentheses, which can include arguments for the class constructor.

4. It can have its own unique attributes that are not shared by other instances of the same class.

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

Ans: In Python, the first argument in a class's method function is usually named "self" by convention, and it refers to the instance of the class that the method is called on. This first argument is special because it allows the method to access and modify the attributes of the instance, and it enables the instance to call other methods of the class.

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

Ans: It is a constructor to initialize class variables for that particular instance.

Q6. What is the process for creating a class instance?

Ans:

1. Define the class: First, you need to define the class using the class keyword.

2. Instantiate the class: Once you have defined the class, you can create an instance of the class by calling the class name followed by parentheses. This calls the constructor method, which creates a new instance of the class.

3. Optional: Pass arguments to the constructor: If the class has a constructor that takes arguments, you can pass those arguments to the constructor when you create the instance.

4. Optional: Modify the instance attributes:

Q7. What is the process for creating a class?

Ans:

1. Define the class: First, you need to define the class using the class keyword.

2. Add the class attributes.

3. Add the class methods.

4. Define the constructor, with arguement 'self'.

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

Ans: Need to use the concept of Inheritance.

Ex:

class Parent:

pass

class Child(Parent): #arguement here is parent class name

pass

Using super() method to access the parent constructor and super keyword to access the parent class variables and methods.