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

Ans. In Python, object-oriented Programming (OOPs) is a programming paradigm that uses objects and classes in programming. It aims to implement real-world entities like inheritance, polymorphisms, encapsulation, etc. in the programming.

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

Ans. 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).

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

Ans. Key Differences between Class and Object

A class is a template for creating objects in a program, whereas the object is an instance of a class.

A class is a logical entity, while an object is a physical entity.

A class does not allocate memory space; on the other hand, an object allocates memory space.

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

Ans. The calling process is automatic while the receiving process is not (its explicit). This is the reason the first parameter of a function in class must be the object itself. Writing this parameter as self is merely a convention.

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

Ans. "\_\_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?

Ans. To create instances of a class, you call the class using class name and pass in whatever arguments its \_\_init\_\_ method accepts.

Q7. What is the process for creating a class?

Ans.

Declaration − A variable declaration with a variable name with an object type. Instantiation − The 'new' keyword is used to create the object. Initialization − The 'new' keyword is followed by a call to a constructor. This call initializes the new object

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

Ans. 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.