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

Answer>

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. 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.

Main Concepts of Object-Oriented Programming (OOPs)

Class

Objects

Polymorphism

Encapsulation

Inheritance

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

Answer> inheritance search look for an attribute in Namespace.

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

Answer> Class : A class is a blue print.

. Object : It is the copy of the class. Objects have states and behaviors.

Instance : Its a variable which is used to hold memory address of the object.

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

Answer> the special thing about methods is that the instance object is passed as the first argument of the function. 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

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

Answer> The \_\_init\_\_ method is the Python equivalent of the C++ constructor in an object-oriented approach. The \_\_init\_\_ function is called every time an object is created from a class. The \_\_init\_\_ method lets the class initialize the object's attributes and serves no other purpose. It is only used within classes. The first method \_\_init\_\_() is a special method, which is called class constructor or initialization method that Python calls when you create a new instance of this class.

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

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

emp1 = Employee("Zara", 2000) #"This would create first object of Employee class"

Q7. What is the process for creating a class?

Answer> The class statement creates a new class definition. The name of the class immediately follows the keyword class followed by a colon as follows −

class ClassName:

'Optional class documentation string'

class\_suite

The class has a documentation string, which can be accessed via ClassName.\_\_doc\_\_.

The class\_suite consists of all the component statements defining class members, data attributes and functions.

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

Answer> A superclass is the class from which many subclasses can be created. The subclasses inherit the characteristics of a superclass. The superclass is also known as the parent class or base class.