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**. The main aim of OOP is **to bind together the data and the functions that operate on them so that no other part of the code can access this** data except that function.

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

Ans:- All of these objects are namespaces (packages of variables), and the inheritance search is simply a **search of the tree from bottom to top looking for the lowest occurrence of an attribute name**.

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

Ans:- In simple words, Instance refers to the copy of the object at a particular time whereas **object refers to the memory address of the class**.22-May-2010

The basic concept of OOP is this: **Class >> Object >> Instance.** **The class** = the blue print. The Object is an actual thing that is built based on the 'blue print' (like the house). An instance is a virtual copy (but not a real copy) of the object.

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. It is not a keyword and has no special meaning in Python.

The first argument of every class method, including init, is **always a reference to the current instance of the class**. By convention, this argument is always named self. In the init method, self refers to the newly created object; in other class methods, it refers to the instance whose method was called.

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

Ans:- The \_\_init\_\_ method is similar to constructors in C++ and Java . Constructors are **used to initialize the object's state**. The task of constructors is to initialize(assign values) to the data members of the class when an object of class is created.

“\_\_init\_\_" is a reseved method in python classes. It is known as a constructor in object oriented concepts. This method called **when an object is created from the class and it allow the class to initialize the attributes of a class.**

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

Ans:- When you create an object, you are creating an instance of a class, therefore "instantiating" a class. The new operator requires a single, postfix argument: a **call to a constructor**. The name of the constructor provides the name of the class to instantiate. The constructor initializes the new object.

**Creating** a new **class** creates a new type of **object**, allowing new **instances** of that type to be made. Each **class instance** can have attributes ...

Q7. What is the process for creating a class?

Ans:- **Python Classes and Objects**

1. Create a Class. To create a class, use the keyword class : ...
2. Create Object. Now we can use the class named MyClass to create objects: ...
3. The self Parameter. ...
4. Modify Object Properties. ...

To create a class, use the keyword class:

class MyClass:  
  x = 5

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

Ans:- 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.