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

Answer: OOPS concepts in python (object oriented programming language) using the idea of object to represents data and methods. it is also an approach using the creating neat and reusable code instead of a redundant one . The programming is divided into self contained object or several mini programs. oops is a model that provides different types of concept such as 'inheritance' , 'abstraction' , 'polymorphism' etc. these concept aim to implememt real world entities in programs. They creating working methods and variables to reuse them compromising security. In python we can easily create and use classes and object. An object oriented paradigm is to design the programme using class and object. the object is related to real world entites such as book, pencil, house etc. the oops concept is focus on writing the reusable code.

- class in nothing just a classification
- This is were class come into the picture
- Class is come of component just like a real word :i am classifing a car, human motercycle etc

---

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

Answer: The whole point of a namespace tool like the class statement is to support name inheritance. In Python, inheritance happens when an object is qualified, and involves searching an attribute definition tree (one or more namespaces). Every time you use an expression of the form object.When we search for an attribute in a class that is involved in python multiple inheritance, an order is followed. First, it is searched in the current class. If not found, the search moves to parent classes. Instance attributes are generated by assignments to self attributes in methods.

Class attributes are created by statements (assignments) in class statements.

Superclass links are made by listing classes in parentheses in a class statement header.

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

Answer: when we create a class in python then a class object is created so whenever python finds a class statement in the whole program then it creates a class object and assigns a name to that object i.e. class name. As we know in python, everything is an object so the class itself is an object and is the instance of metaclasses. Instance object inherits the attributes of the class object from which it was created. class object is like a blueprint for intance object but instance object is a concrete item in out code. instance objects are new namespaces, thay start out empty but inherit object attributes that live in class object.

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

Answer: What makes the first argument in a class's method function special in Python? 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?

Answer: "**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. he self in keyword in Python is used to all the instances in a class. By using the self keyword, one can easily access all the instances defined within a class, including its methods and attributes. init. **init** is one of the reserved methods in Python. In object oriented programming, it is known as a constructor.

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

Answer:The phrase "instantiating a class" means the same thing as "creating an object." 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.

Q7. What is the process for creating a class?

Answer: Tap Classroom . Tap Add. ... Enter the class name. (Optional) To enter a short description, grade level, or class time, tap Section and enter the details. (Optional) To enter the location for the class, tap Room and enter the details. (Optional) To add a subject, tap Subject and enter a name. Tap Create.

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. In the above example, Vehicle is the Superclass and its subclasses are Car,Truck and Motercycle

Colab paid products - Cancel contracts here

● ✕