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

**Solution:**

* Python Object Oriented Programming (OOP) is modular coding which is highly reusable.
* It has various concepts such as Class, Objects, Encapsulation, Polymorphism, Inheritance and Data Abstraction.
* Main reason for using Python OOP is make the code reusable by creating functions and calling the same when required any number of times. This also results efficient and fast code understanding and modification becomes easier.

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

**Solution:**

* 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 super-classes, progressing from left to right (by default).

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

**Solution:**

* Class variables are defined inside a class. This can be accessed only inside a class by calling a class object.
* Instance variables are nothing but copy of objects. This can be called using a class name followed by instance object name. These are going to be unique for every instance of our class.

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

**Solution:**

* First argument in a class’s method is always self and it has to be mentioned explicitly always. “self” is not a keyword it can be replaced with anything like your name or my name.
* Self is used to represent an instance of the object of the given class. This is needed so that same class can hold the information for multiple class objects.

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

**Solution:** The \_\_init\_\_ method is similar to constructors in java & C++. It is called whenever an object is instantiated. This method is useful to do any initialization you want to do with your object.

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

**Solution:**

* Class instance is nothing but the object which can be used to call the class method.
* It is created as below (tata is the class instance) :

class Car:

def mileage (self, cost):

self.cost = cost

tata = Car()

Q7. What is the process for creating a class?

**Solution:**

* Class can be created by using a keyword **class**.

Eg: class car:

def x1:

pass

* We can have any number of required classes in python script and define any specific number of functions inside class.

Q8. How would you define the super classes of a class?

**Solution:**

* Super class is nothing but the parent class of a class. Super class is available in inheritance and it inherits all the properties, methods of parent class.
* It is defined as follows (Car is superclass of Mercedes class) :

class Car:

def specifications (self, color, fuel):

pass

class Mercedes(Car):

pass