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

In python Object Oriented Programming deals with the concept of class and Object. It is used to structure the python program into simple and reusable form(classes) of which we can create instance/object. It aims to implement to implement inheritance, abstraction, polymorphism, encapsulation.

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

1. In inheritance what ever methods, variables, constructors parent class has by-default available to child class, there is not required to redefine them in child class. In case of code execution inheritance search look for attributes of parent class.
2. We use inheritance for code reusablility and extendability.

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

1. Class object is a real object which is the blue print and built from a class.
2. Instance object is a virtual copy of the object
3. You can distinguish them by the address of the object. For instance object the address will be always same. But in case of class object the address of the object is different always.

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

1. The first argument is a class method should be object itself. Self is the variable which indicates current class object.
2. It is not a keyword, only a naming convention.

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

1. \_\_init\_\_ is the constructor. It is a special type of method which gets executed when an object is created. The main purpose of constructor is to initialize the instance variables.
2. It takes the first argument as self. In python name of the constructor is always \_\_init\_\_.
3. In python constructor is optional, if we are not creating ant constructor then default constructor will be given by PVM.

Eg :

class A:

def \_\_init\_\_(self):

self.B = 10

a = A()

print(a.\_\_dict\_\_) #displays the properties of the class

class Employee:

def \_\_init\_\_(self, eno, esal):

self.eno = eno

self.esal = esal

e = Employee()

print(e.eno)

print(e.esal)

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

In python to create an object we have only one process. First we have to create a class by giving the keyword and class name. Then we need to create and object by giving paranthesis after the name of the class. To access it we need to assign it to a variable.

Eg :

class A:

def function(self, ch):

return ch

a = A()

print(a.function(‘S’))

Q7. What is the process for creating a class?

In python to create a class we need to write class name followed by class keyword and the we need to provide colon. As per OOP concept the class name should start with capital letter and then by small letter.

Eg:

class A:

def function(self, ch):

return ch

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

1. To define a super class of a class we need to write the super class name in parantheses after child class name while creating the child class. By this process the child class will extend the parent class attributes.
2. To access the attributes of parent class from child class, you need to create an object of child class.

Eg:

class Parent:

def p(self):

print('Parent method')

class Child(Parent):

def c(self):

print('Child Method')

c = Child()

print(c.c())

print(c.p())