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. in the programming. The concept of OOP in Python focuses on creating reusable code. Object-oriented programming (OOP) is a method of structuring a program by bundling related properties and behaviors into individual objects.

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

Ans. 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.

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

Ans. CLASS:

a. Class is used as a template for declaring and creating the objects.

b. When a class is created, no memory is allocated.

c. The class has to be declared first and only once.

d. A class cannot be manipulated as they are not available in the memory.

e. A class is a logical entity.

f. Class does not contain any values which can be associated with the field.

OBJECT:

a. An object is an instance of a class.

b. Objects are allocated memory space whenever they are created.

c. An object is created many times as per requirement.

d. Objects can be manipulated.

e. An object is a physical entity.

f. Each object has its own values, which are associated with it.

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

Ans. Self represents the instance of the class. By using the “self” we can access the attributes and methods of the class in python. It binds the attributes with the given arguments.

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

Ans. \_\_init\_\_ is a reseved method in python classes. 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.

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

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

Q7. What is the process for creating a class?

Ans. In Python, a class can be created by using the keyword class, followed by the class name. The syntax to create a class is given below. In Python, we must notice that each class is associated with a documentation string which can be accessed by using <class-name>. \_\_doc\_\_.

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.