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

Ans: the purpose of Python’s OOP are -  
 1. make the programming modular for easier troubleshooting  
 2. reuse the code

3. make the code more flexible

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

Ans: first in child class if not found then in parent class.

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

Ans: (i)class object is like a blueprint for intance object but instance object is a concrete item in out code.

(ii) instance objects are new namespaces, thay start out empty but inherit object attributes that live in class object.

(iii) The first argumetn of class functions(self) reference the instance object and assignments to attributes of self change data in the instance.

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.

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

Ans: The \_\_init\_\_ method is the Python equivalent of the C++ constructor in an object-oriented approach. 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: 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 new operator returns a reference to the object it created.

Q7. What is the process for creating a class?

Ans: A Class is like an object constructor, or a "blueprint" for creating objects. To create a class, we use the keyword class:

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

Ans: A class that is derived from another class is called a subclass (also a derived class, extended class, or child class). The class from which the subclass is derived is called a superclass (also a base class or a parent class).