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

Python is a multi-paradigm programming language. It supports different programming approaches. The concept of OOP in python focuses on creating reusable code. This concept is also known as DRY(Don’t Repeat Yourself)

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

All of these objects are namespaces (packages of variables), and the inheritance search is simply a search of the tree from bottom to top looking for the lowest occurrence of an attribute name.

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

A class is a blueprint which you use to create objects. An object is an instance of a class-it’s a concrete ‘thing’ that you made using a specific class. So, ‘object’ and ‘instance’ are the same thing, but the word ‘instance’ indicates the relationship of an object to the class.

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

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. It is not a keyword and has no special meaning in python.

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

“\_\_init\_\_” is a reserved method in. python classes. It is called as 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.

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

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. The name of the constructor provides the name of the class to instantiate. The constructor initializes the new object.

Q7. What is the process for creating a class?

A class is a user-defined blueprint or prototype form which objects are created. Classes provide a means of bundling data and functionality together. Creating a new class creates a new type of object, allowing new instances of that type to be made. Each class instances can have attributes attached to it for maintaining its state.

There are three steps when creating an object from a class. **Declaration:** A variable declaration with a variable name with an object type. **Instantiation**: The “new” keyword is used to create the object. **Initialization:** The “new” keyword is followed by a call to a constructor

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

The class form which a class inherits is called the parent or superclass. A class which inherits from a superclass is called a subclass also called their class or child class. Super classes are sometimes called ancestors as well.

The super () built-in returns as proxy object(temporary object of the superclass) that allows us to access methods of the base class. In python, super () have two major use cases: allows us to avoid using the basic class name explicitly. Working with multiple inheritances.