Q1. What is the meaning of multiple inheritance?

Ans. When a class is derived from more than one base class it is called multiple Inheritance. The derived class inherits all the features of the base case.

Example:

class Parent\_one:

pass

class Parent\_two:

pass

class child(Parent\_one,Parent\_two):

pass

Q2. What is the concept of delegation?

Ans. In object-oriented programming, delegation refers to evaluating a member (property or method) of one object (the receiver) in the context of another original object (the sender). The delegation pattern is an object-oriented design pattern that allows object composition to achieve the same code reuse as inheritance.

Q3. What is the concept of composition?

Ans. Composition is one of the important concepts of Object-oriented programming (OOPs). Composition basically enables us for creating complex types objects by combining other types of objects in the program.

Q4. What are bound methods and how do we use them?

Ans. A bound method is the one which is dependent on the instance of the class as the first argument. It passes the instance as the first argument which is used to access the variables and functions. A bound method is one that has ' self ' as its first argument. Since these are dependent on the instance of classes, these are also known as instance methods.

Q5. What is the purpose of pseudoprivate attributes?

Ans. Within a class method in Python, whenever a method assigns to a self attribute (e.g., self.attr=value), it changes or creates an attribute in the instance (inheritance search only happens on reference, not assignment). Because this is true even if multiple classes in a hierarchy assign to the same attribute, collisions are possible. Hence pseudoprivate attributes are used.

Pseudoprivate attributes are also useful in larger frameworks or tools, both to avoid introducing new method names that might accidentally hide definitions elsewhere in the class tree and to reduce the chance of internal methods being replaced by names defined lower in the tree. If a method is intended for use only within a class that may be mixed into other classes, the double underscore prefix ensures that the method won't interfere with other names in the tree, especially in multiple-inheritance scenarios.