Q1. What is the meaning of multiple inheritance?

A: 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.

class Class1:

def m(self):

print("In Class1")

class Class2(Class1):

def m(self):

print("In Class2")

Class1.m(self)

class Class3(Class1):

def m(self):

print("In Class3")

Class1.m(self)

class Class4(Class2, Class3):

def m(self):

print("In Class4")

Class2.m(self)

Class3.m(self)

obj = Class4()

obj.m()

Q2. What is the concept of delegation?

A:

Delegation is an object oriented technique (also called a design pattern). Let's say you have an object x and want to change the behaviour of just one of its methods. You can create a new class that provides a new implementation of the method you're interested in changing and delegates all other methods to the corresponding method of x.

Q3. What is the concept of composition?

A:

Here by using the class name or by creating the object we can access the members of one class inside another class. It enables creating complex types by combining objects of different classes. It means that a class Composite can contain an object of another class Component. This type of relationship is known as **Has-A Relation**.

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

A:

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.

class A:

def func(self, arg):

self.arg = arg

print("Value of arg = ", arg)

# Creating an instance

obj = A()

# bound method

obj.func

Q5. What is the purpose of pseudoprivate attributes?

A:

Pseudo private 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.