**Q1. What is the meaning of multiple inheritance?**

* **Multiple Inheritance**:
  + A class can inherit from more than one superclass.
  + Combines attributes and methods from multiple classes.
  + Example:

class SuperClass1:

pass

class SuperClass2:

pass

class SubClass(SuperClass1, SuperClass2): # Multiple inheritance

pass

* **Use Case**:
  + Useful for creating classes that need to combine features from multiple sources.

**Q2. What is the concept of delegation?**

* **Delegation**:
  + An object handles a request by passing it to a second object (the delegate).
  + Promotes code reuse and separation of concerns.
  + Example:

class Delegate:

def process(self):

print("Processing in Delegate")

class MainClass:

def \_\_init\_\_(self):

self.delegate = Delegate()

def process(self):

self.delegate.process() # Delegate the task

**Q3. What is the concept of composition?**

* **Composition**:
  + A class is composed of one or more objects of other classes.
  + Represents a "has-a" relationship.
  + Example:

class Engine:

def start(self):

print("Engine started")

class Car:

def \_\_init\_\_(self):

self.engine = Engine() # Composition

def start(self):

self.engine.start()

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

* **Bound Methods**:
  + Methods that are associated with an instance of a class.
  + Automatically pass the instance (self) as the first argument.
  + Example:

class MyClass:

def my\_method(self):

print("Hello from my\_method")

instance = MyClass()

bound\_method = instance.my\_method # Bound method

bound\_method() # Automatically passes 'instance' as self

**Q5. What is the purpose of pseudoprivate attributes?**

* **Pseudoprivate Attributes**:
  + Attributes prefixed with \_ClassName (e.g., \_MyClass\_\_attr).
  + Not truly private but name-mangled to avoid accidental overriding in subclasses.
  + Example:

class MyClass:

def \_\_init\_\_(self):

self.\_\_attr = 42 # Pseudoprivate attribute

instance = MyClass()

print(instance.\_MyClass\_\_attr) # Accessing pseudoprivate attribute

* **Purpose**:
  + Prevent name clashes in subclasses.
  + Indicate that an attribute is intended for internal use.