**FSDS MAY BATCH 2022(Python Assignment -5)**

**Submitted by: Shubham Tiwari**

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.The basic syntax of it is shown below:

class BaseClass1:

*# Body of base class 1 (parent class 1)*

class BaseClass2:

*# Body of base class 2 (parent class 2)*

class DerivedClass(BaseClass1, BaseClass2):

*# Body of derived class (child class)*

*# Inherited properties of baseclass1 and baseclass2*

Here, we have defined two classes which would be the parent classes of our new derived class. The derived class will have the properties of BaseClass1 as well as BaseClass2 as specified in the code above, i.e. the child has properties of both parents.

Q2: What is the concept of delegation?

Ans: Delegation simply means if we want an object to pass attribute access requests to another object it contains, rather than handling them itself or inheriting the attributes from a parent class, we can use delegation. This can be useful for implementing a proxy or for an alternative to inheritance.In other words it simply means the delegation pattern is an object-oriented design pattern that allows object composition to achieve the same code reuse as inheritance.

For example:

class Animal:

def \_\_init\_\_(self, name, num\_of\_legs):

self.name = name

self.num\_of\_legs = num\_of\_legs

def get\_number\_of\_legs(self):

print(f"I have {self.num\_of\_legs} legs")

class Dog(Animal):

def \_\_init\_\_(self, name, num\_of\_legs):

super().\_\_init\_\_(name, num\_of\_legs)

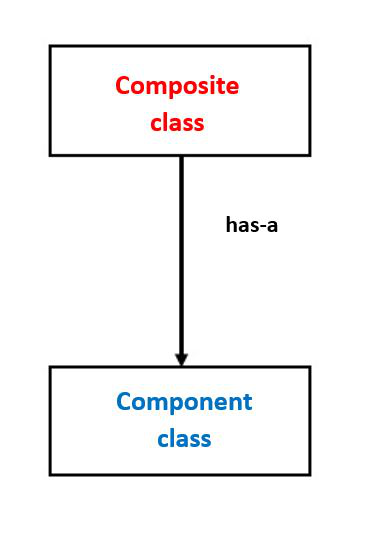
dog = Dog('Fido', 4) dog.get\_number\_of\_legs()

# Output:

"I have 4 legs".

Q3: What is the concept of composition?

Ans: **Composition** is a concept that means models a **has a** relationship. 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 called as **Has-A Relation**.



The basic syntax can be written as :

class A :

# variables of class A

# methods of class A

...

...

class B :

# by using "obj" we can access member's of class A.

obj = A()

# variables of class B

# methods of class B

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**](https://www.geeksforgeeks.org/self-in-python-class/)‘ as its first argument. Since these are dependent on the instance of classes, these are also known as instance methods.The methods inside the classes would take at least one argument. To make them zero-argument methods, ‘[**decorators**](https://www.geeksforgeeks.org/decorators-in-python/)**‘** has to be used. Different instances of a class have different values associated with them.

Q5: What is the purpose of pseudoprivate attributes?

Ans: Pseudoprivate attributes are 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. The problem that the pseudo-private attribute feature is basically to reduce with the way instance attributes are stored. In Python, all instance attributes wrap up in the single instance object at the bottom of the class tree.