Name: Tausif Khan

Rollno: 12

Branch: SE/COMPS Subject - OSTL

Div: B

Experiment No: 6b.

**AIM:** A Python program to demonstrate multiple inheritance and display the order of execution of methods in several base classes according to MRO.

**TOOLS USED:** Python 3.4.3, Terminal

# THEORY:

* 1. **Explain Inheritance in Python with syntax**

 Creating new classes from existing classes, so that new classes will acquire all the Features of the existing classes is called inheritance.

 Lets take class A with some members. If we feel another class B wants almost same

members, then we can derive or create class B from A as:

class B(A)

 The original class A is called the base class and class B is called the derived class or sub class.

 3 Advantages:

* + - We can create more useful classes needed by the application.
    - The process of creating the new classes is very easy, since they are built upon already existing classes.
    - Managing the code becomes easy.

 Syntax:- class BaseClass:

Body of base class

class DerivedClass(BaseClass):

Body of derived class

# Explain what is constructor overriding and method overriding and how to overcome

constructor overriding and method overriding

 Constructor Overriding is never possible in Java. This is because, Constructor looks like a method but name should be as class name and no return value. Overriding means what we have declared in Super class, that exactly we have to declare in Sub class it is called Overriding.

 Declaring a method in sub class which is already present in parent class is known as method overriding. Overriding is done so that a child class can give its own implementation to a method which is already provided by the parent class. In this case the method in parent class is called overridden method and the method in child class is called overriding method.

# Explain the types of inheritance with syntax

 The types of inheritance are:-

 Single Inheritance:- A derived class with only one base class is called single inheritance

Syntax:- class base

{

int i,j; public:

void set(int a,int b); void show();

}

class derived:public base

{

int Ki public: derived(int x)

.

.

.

}

 Multiple Inheritance:-A derived with several base class is called Multiple Inheritance.

Syntax:-

Class D:visibility B,visibility A

{

Body of D

…

…

};

# What is Method Resolution Order (MRO)

 In multiple inheritance, the method is first in the current class.

 Left to right search, if not found

 Method Resolution Order (MRO) in Python. Method Resolution Order (MRO) is the order in which Python looks for a method in a hierarchy of classes. Especially it plays vital role in the context of multiple inheritance as single method may be found in multiple super classes.

# Explain polymorphism with examples

 Polymorphism represents the ability to assume several different forms.

 In programming, if an object or method is exhibiting different behavior in different contexts , it is called polymorphic nature.

 Polymorphism provides flexibility in writing programs in such a way that the programmer uses same method call to perform different operations depending on the requirement.

 Example: A Function that exhibits polymorphism def add(a,b)

print(a+b)

# call add() and pass two integers add(5,10)

# call add() and pass two integers add(“hello”,”ALL”)

**CODE & OUTPUT:**

**Code:**

class A:

def init\_(self):

self.a='a'

print(self.a)

super().\_init\_(self)

class B:

def init\_(self):

self.b='b'

print(self.b)

super().\_init\_(self)

class C:

def init\_(self):

self.c='c'

print(self.c)

super().\_init\_(self)

class X(A,B):

def method(self):

print(" X class method")

super().method()

class Y(B,C):

def method(self):

print(" Y class method")

super().method()

class P(X,Y,C):

def method(self):

print(" P class method")

super().method()

print(P.mro())

**Output:**

[<class '\_\_main\_\_.P'>, <class '\_\_main\_\_.X'>, <class '\_\_main\_\_.A'>, <class '\_\_main\_\_.Y'>, <class '\_\_main\_\_.B'>, <class '\_\_main\_\_.C'>, <class 'object'>]

**CONCLUSION:** Thus we have successfully studied and implemented the multiple Inheritance method using MRO in python.