# Day 10– Inheritance

# Inheritance provides code reusability to the program. The child class acquires the properties and can access all the data members and functions defined in the parent class.  Child class can also provide its specific implementation to the functions of the parent class. Derived class can inherit base class by just mentioning the base in the bracket after the derived class name. Consider the following syntax to inherit a base class into the derived class

**Single Inheritance:**

class Parent:

def func1(self):

print("this is example one")

class Child(Parent):

def func2(self):

print(" this is example 2 ")

ob = Child()

ob.func1()

ob.func2()

## **Multiple Inheritance**

When a child class inherits from more than one parent class

class Parent:

def func1(self):

print("this is example 1")

class Parent2:

def func2(self):

print("this is example 2")

class Child(Parent , Parent2):

def func3(self):

print("this is example 3")

ob = Child()

ob.func1()

ob.func2()

ob.func3()

## **Multilevel Inheritance**

When a child class becomes a parent class for another child class.

class Parent:

def func1(self):

print("this is example 1")

class Child(Parent):

def func2(self):

print("this is example 2")

class Child2(Child):

def func3("this is example 3")

ob = Child2()

ob.func1()

ob.func2()

ob.func3()

## **Hierarchical Inheritance**

Hierarchical inheritance involves multiple inheritance from the same base or parent class.

class Parent:

def func1(self):

print("this is example 1")

class Child(Parent):

def func2(self):

print("this is example 2")

class Child2(Parent):

def func3(self):

print("this is example 3")

ob = Child()

ob1 = Child2()

ob.func1()

ob.func2()

## **Hybrid Inheritance**

Hybrid inheritance involves multiple inheritance taking place in a single program.

class Parent:

def func1(self):

print("this is example 1")

class Child(Parent):

def func2(self):

print("this is example 2")

class Child1(Parent):

def func3(self):

print(" this is example 3"):

class Child3(Parent , Child1):

def func4(self):

print(" this is example 4")

ob = Child3()

ob.func1()

## **Python Super() Function**

Super function allows us to call a method from the parent class.

class Parent:

def func1(self):

print("this is example 1")

class Child(Parent):

def func2(self):

Super().func1()

print("this is example 2")

ob = Child()

ob.func2()

**Method Overriding**

class Parent:

def func1(self):

print("this is parent function")

class Child(Parent):

def func1(self):

print("this is child function")

ob = Child()

ob.func1()

# Task:

# Create a real time scenario for inheritance example Banking concept, ecommerce concept.