**Que.1-Single,Multilevel,Multiple,Hierarchical and Hybrid Inheritance in Python.**

**Ans.**

**->single inheritance: single inheritance allows a class to inherit from one superclass.This means a single derived class inheritsfeatures from one base class.**

**Ex. class parent:**

**Def parent\_method():**

**Print(“this is the parent method”)**

**class child(parent):**

**Def child\_method():**

**Print(“this is thechild method”)**

**child\_object=child()**

**child\_object.parent\_method()**

**child\_object.child\_method()**

**->Multilevel inheritance: multilevel inheritance is a feature where a class is derived from another class,which is also derived from another class,forming a “chain” of inheritance.**

**Ex. Class grandparent:**

**Def grandparent\_method(self):**

**Print(“this is the grandparent method”)**

**Class parent(grandparent):**

**Def parent\_method(self):**

**Print(“this is the parent method”)**

**Class child(parent):**

**Def child\_method(self):**

**Print(“this is the child method”)**

**Child\_object=child()**

**Child\_object.greandparent\_method()**

**Child\_object.parent\_method()**

**Child\_object.child method()**

**->Multiple Inheritance:Multiple inheritance allows aclass to inherit from more than one superclass.This canlead to the “diamond problem” but python’s method resolution order handles it.**

**Ex. class Mother:**

**Def mother\_method():**

**Print(“this is the mother method”)**

**Class father:**

**Def father\_method():**

**Print(“this is the father method”)**

**Class child(mother,father):**

**Def child\_method():**

**Print(“this is the child method”)**

**Child\_object=child()**

**Child\_object.mother\_method()**

**Child\_object.father\_method()**

**Child\_object.child\_method()**

**->Hierarchical Inheritance: In hierarchical inheritance multiple derived classes inherit from a single base class.**

**Ex.**

**Class parent:**

**Def parent\_method():**

**Print(“this is parent method”)**

**Class childone(parent):**

**Def child\_one\_method():**

**Print(“this is the first child method”)**

**Class childtwo(parent):**

**Def child\_two\_method():**

**Print(“this is the second child method”)**

**Child\_one\_object=childone()**

**Child\_two \_objec=childtwo()**

**Child\_one\_object.parent\_method()**

**Child\_one\_object.child\_one\_method()**

**Child\_two\_object.parent\_method()**

**Child\_two\_object.child\_two\_method()**

**->Hybrid inheritance: hybrid inheritance is a combination of two or more types of inheritance.it can lead to complex inheritance hierarchies.**

**Ex. class grandparent:**

**Def grandparent\_method():**

**Print(“this is the grandparent method”)**

**class parentone(grandparent):**

**Def parent\_one\_method():**

**Print(“this is the first parent method”)**

**Class parenttwo(grandparent):**

**Def parent\_two\_method():**

**Print(“this is the second parent method”)**

**Class child(parentone,parenttwo):**

**Def child\_method():**

**Print(“this is the child method”)**

**Child\_object=child()**

**Child\_object.grandparent\_method()**

**Child\_object.parent\_one\_method()**

**Child\_object.parent\_two\_method()**

**Child\_object.child\_method()**

**Que-2 : using the super() function to acess properties of the parent class.**

**Ans.**

**Super() function in python is a handy tool for accessing methods and properties from a parent class.i allows you to call a method from a superclass from within a subclass,ensuring that you can extend or modify inherited behaviours without losing the original functionality.**

**Ex.,**

**Class parent:**

**Def show():**

**Print(“this is the parent method”)**

**Class child(parent):**

**Def show():**

**Super().show()**

**Print(“this is the child method”)**

**Child\_object=child()**

**Child\_object.show()**