|  |
| --- |
| Single Level Inheritance When a Child class inherits from only one parent class is called single level inheritance  class Parent: # BaseClass or SuperClass  def d1(self):  print('Parent Function')  class Child(Parent): # Derived1(Base1): or Sub Class  def d2(self):  print('Child Function')  s = Child() s.d1() # Parent Function s.d2() # Child Function  Parent Function  Child Function |

|  |
| --- |
| Multilevel Inheritance Child Class inherits all the aspects of Parent Class and GrandParent Class  class GrandParent: # Base1 or Superclass  def d1(self):  print('Grand Parent Function')  class Parent(GrandParent): # Derived1(Base1): or Subclass  def d2(self):  print('Parent Function')  class Child(Parent): # Derived2(Base2): or Subclass  def d3(self):  print('Child Function') # Derived2(Derived1): or Subclass  s = Child() s.d3() s.d2() s.d1()  Child Function  Parent Function  Grand Parent Function |

|  |
| --- |
| Multiple Inheritance When a class is derived from more than one base class In multiple inheritance, the features of all the base classes are inherited into the derived class  class GrandParent: # Base1 or Superclass  def d1(self):  print('Grand Parent Function')  class Parent(): # Base2 or Superclass  def d2(self):  print('Parent Function')  class Child(Parent, GrandParent): # MultiDerived(Base1, Base2): or Subclass  def d3(self):  print('Child Function')  s = Child() s.d3() s.d2() s.d1()  Child Function  Parent Function  Grand Parent Function |

|  |
| --- |
| Hierarchical Inheritance When more than one derived classes are created from a single base class  class GrandParent: # Base1 Superclass  def d1(self):  print('Grand Parent Function')  class Parent(GrandParent): # Derived1(Base1): subclass  def d2(self):  print('Parent Function')  class Child(GrandParent): # Derived1(Base1): subclass  def d3(self):  print('Child Function')  s = Child() s.d3() s.d1()  p = Parent() p.d2() p.d1()  Child Function  Grand Parent Function  Parent Function  Grand Parent Function |

|  |
| --- |
| Hybrid Inheritance It is a combination of multi-level and hierarchical inheritance  class GrandParent: # Base1 or Superclass  def d1(self):  print('Grand Parent Function')  class Parent(GrandParent): # Derived1(Base1):  def d2(self):  print('Parent Function')  class Child(Parent, GrandParent): # MultiDerived(Base1, Base2):   def d3(self):  print('Child Function')  s = Child() s.d3() s.d2() s.d1()  o = Parent() o.d2() o.d1()  Child Function  Parent Function  Grand Parent Function  Parent Function  Grand Parent Function |

|  |
| --- |
| class Student:   # Instance Method  def d1(self, firstName, lastName, age):  self.firstName = firstName  self.lastName = lastName  self.age = age   # Instance Method  def display(self):  print(self.firstName)  print(self.lastName)  print(self.age)  class College(Student):   # Static Method  @staticmethod  def d3(collegeName):  print(collegeName)  c = College() c.d1('Sai', 'Kiran', 28) c.display()  College.d3("JNTU")  Sai  Kiran  28  JNTU |

|  |
| --- |
| # Invoking Constructor and Instance methods class Parent:   # Constructor  def \_\_init\_\_(self, id, name):  self.id = id  self.name = name  print('Parent Constructor Invoked')  class Child(Parent):   # Instance Method  def d1(self, age):  self.age = age  print(self.age)  c = Child(101, "SaiKiran") print(c.id, c.name) c.d1(28)  Parent Constructor Invoked  101 SaiKiran  28 |