Inheritance

Inheritance is a process in which one object acquires all the properties and behaviors of its parent object automatically. In such way, you can reuse, extend or modify the attributes and behaviors which are defined in other class.

In Python, the class which inherits the members of another class is called derived class and the class whose members are inherited is called base class. The derived class is the specialized class for the base class.

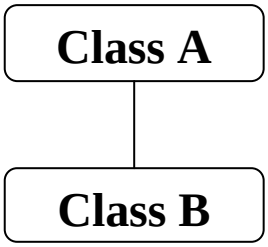
The class whose properties are inherited by other class is called the Parent or Base or Super class. And, the class which inherits properties of other class is called Child or Derived or Sub class.

Inheritance makes the code reusable. When we inherit an existing class, all its methods and fields become available in the new class, hence code is reused.

Types Of Inheritance

- Single inheritance
- Multiple inheritance
- Hierarchical inheritance
- Multilevel inheritance
- Hybrid inheritance

Single Inheritance



In single inheritance, there is only one base class and one derived class. The Derived class gets inherited from its base class. This is the simplest form of inheritance. Above figure is the diagram for single inheritance.

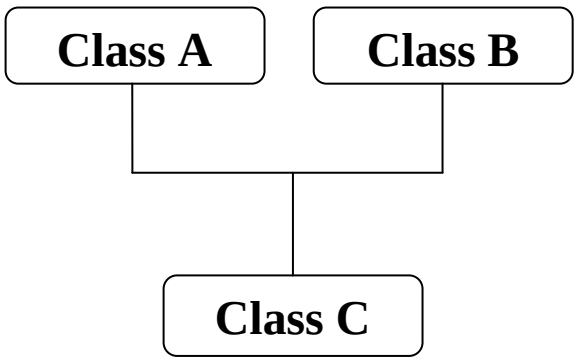
Example	Output

class A:
  def a(self):
  print "Class A"
 	
class B(A):
  def b(self):
  print "Class B"
 	
objb=B()
objb.a()
objb.b()
	Class A
	Class B

Example	Output
class A: x=10	30

class B(A):
  y=20
  def b(self):
  print A.x+B.y
 	
objb=B()
objb.b()	

Multiple Inheritance



In this type of inheritance, a single derived class may inherit from two or more base classes. above figures is the structure of Multiple Inheritance.

Example	Output

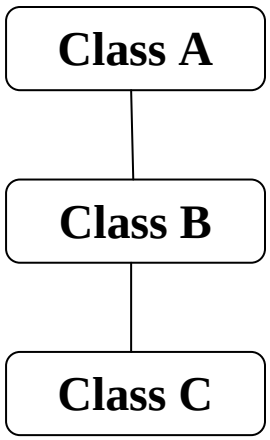
class A:
  def a(self):
  print "Class A"
 	
class B():
  def b(self):
	Class A
	Class B
	Class C

print "Class B"
 	
class C(A,B):
  def c(self):
  print "Class C"
 	
objc=C()
objc.a()
objc.b()
objc.c()	

Example	Output

class A:
  x=10
 	
class B:
  y=20
 	
class C(A,B):
  def c(self):
  print A.x+B.y
 	
objc=C()
objc.c()
	30

Multilevel Inheritance



The classes can also be derived from the classes that are already derived. This type of inheritance is called multilevel inheritance. . above figures is the structure of Multilevel Inheritance.

Example	Output

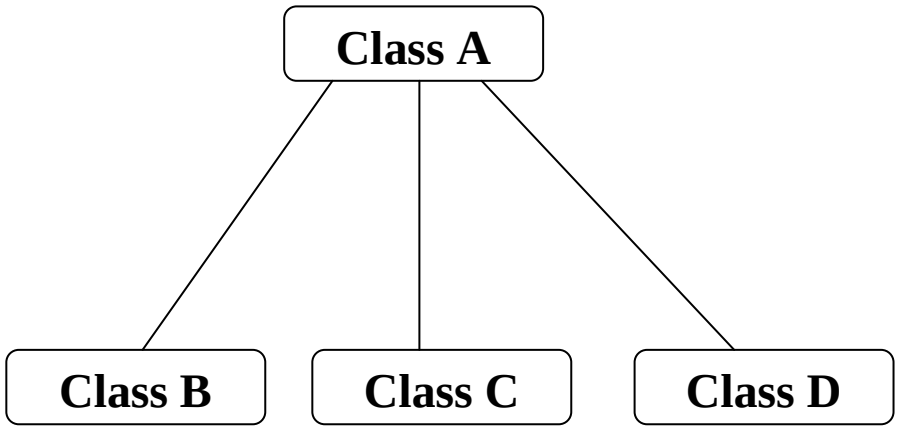
class A:
  def a(self):
  print "Class A"
 	
class B(A):
  def b(self):
  print "Class B"
 	
class C(B):
  def c(self):
  print "Class C"
 	
objb=B()
objb.a()
objb.b()
	Class A
	Class B
	Class A
	Class B
	Class C

objc=C() objc.a() objc.b() objc.c()	

Example	Output

class A:
  x=10
 	
class B(A):
  y=20
  def b(self):
  print A.x+B.y
 	
class C(B):
  def c(self):
  print A.x+B.y
 	
objb=B()
objb.b()
 	
objc=C()
objc.c()
	30
	30

Hierarchical Inheritance



In this type of inheritance, multiple derived classes get inherited from a single base class. Above figures is the structure of Hierarchical Inheritance.

Example	Output

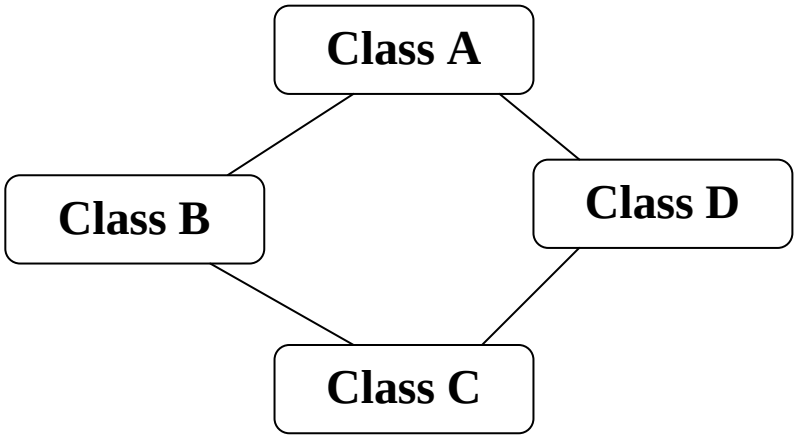
class A:
  def a(self):
  print "Class A"
 	
class B(A):
  def b(self):
  print "Class B"
 	
class C(A):
  def c(self):
  print "Class C"
 	
objb=B()
objb.a()
objb.b()
 	
objc=C()
objc.a()
	Class A
	Class B
	Class A
	Class C

objc.c()	

Example	Output

class A:
  x=10
 	
class B(A):
  y=20
  def b(self):
  print A.x+B.y
 	
class C(A):
  y=30
  def c(self):
  print A.x+C.y
 	
objb=B()
objb.b()
 	
objc=C()
objc.c()
	30
	40

Hybrid Inheritance



This is a Mixture of two or More Inheritance and in this Inheritance, a Code May Contains two or Three types of inheritance in Single Code. Above figure is the diagram for Hybrid inheritance.