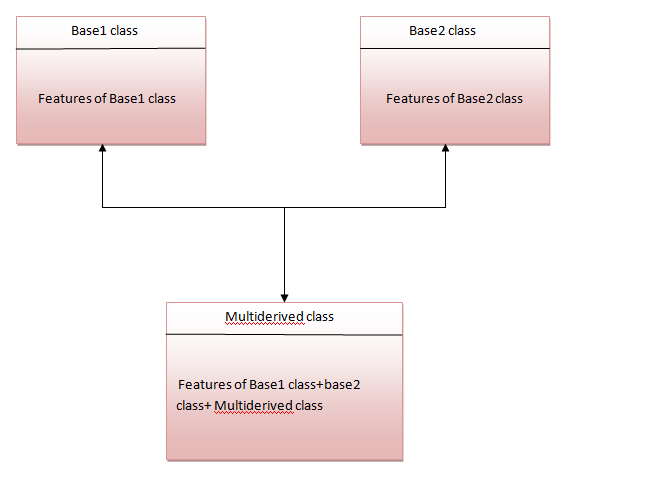
# Multiple Inheritance in Python

Python supports multiple inheritance also. You can derive a child class from more than one base (parent) class.

**Image representation:**



The multiderived class inherits the properties of both class base1 and base2.

Let's see the syntax of multiple inheritance in Python.

**Syntax:**

1. class DerivedClassName(Base1, Base2, Base3):
2. <statement-1>
3. .
4. .
5. .
6. <statement-N>

**Or**

1. class Base1:
2. pass
4. class Base2:
5. pass
7. class MultiDerived(Base1, Base2):
8. pass

**Example:**

1. class First(object):
2. def \_\_init\_\_(self):
3. super(First, self).\_\_init\_\_()
4. print("first")
6. class Second(object):
7. def \_\_init\_\_(self):
8. super(Second, self).\_\_init\_\_()
9. print("second")
11. class Third(Second, First):
12. def \_\_init\_\_(self):
13. super(Third, self).\_\_init\_\_()
14. print("third")
16. Third();

**Output:**

1. first
2. second
3. third

## Why super () keyword

The most commonly super() is used with \_\_init\_\_ function in base classes. This is usually the only place where you need to do some things in a child then complete the initialization in the parent.

**See this example:**

1. class Child(Parent):
2. def \_\_init\_\_(self, stuff):
3. self.stuff = stuff
4. super(Child, self).\_\_init\_\_()

## Composition in Python

Composition is used to do the same thing which can be done by inheritance.