

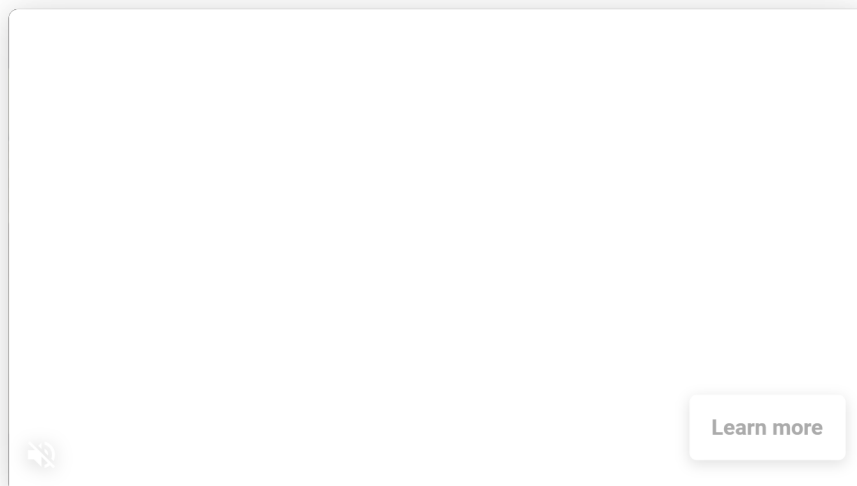
# OOP in Python | Set 3 (Inheritance, examples of object, issubclass and super)

We have discussed following topics on Object Oriented Programming in Python

- [Object Oriented Programming in Python | Set 1](#)
- [Object Oriented Programming in Python | Set 2 \(Data Hiding and Object Printing\)](#)

In this article, Inheritance is introduced.

One of the major advantages of Object Oriented Programming is re-use. Inheritance is one of the mechanisms to achieve the same. In inheritance, a class (usually called superclass) is inherited by another class (usually called subclass). The subclass adds some attributes to superclass.

[i](#) [x](#)

Below is a sample Python program to show how inheritance is implemented in Python.

```
# A Python program to demonstrate inheritance

# Base or Super class. Note object in bracket.
# (Generally, object is made ancestor of all classes)
# In Python 3.x "class Person" is
# equivalent to "class Person(object)"
class Person(object):

    # Constructor
    def __init__(self, name):
        self.name = name

    # To get name
    def getName(self):
        return self.name

    # To check if this person is employee
    def isEmployee(self):
        return False
```



```
# Inherited or Sub class (Note Person in bracket)
class Employee(Person):

    # Here we return true
    def isEmployee(self):
        return True

# Driver code
emp = Person("Geek1") # An Object of Person
print(emp.getName(), emp.isEmployee())

emp = Employee("Geek2") # An Object of Employee
print(emp.getName(), emp.isEmployee())
```

**Output:**

```
('Geek1', False)
('Geek2', True)
```

**How to check if a class is subclass of another?**

Python provides a function `issubclass()` that directly tells us if a class is subclass of another class.

```
# Python example to check if a class is
# subclass of another
```

```
class Base(object):
    pass # Empty Class

class Derived(Base):
    pass # Empty Class

# Driver Code
print(issubclass(Derived, Base))
print(issubclass(Base, Derived))

d = Derived()
b = Base()

# b is not an instance of Derived
print(isinstance(b, Derived))

# But d is an instance of Base
print(isinstance(d, Base))
```

**Output:**

```
True
False
False
True
```



## What is object class?

Like Java Object class, in Python (from version 3.x), object is root of all classes.

In Python 3.x, "class Test(object)" and "class Test" are same.

In Python 2.x, "class Test(object)" creates a class with object as parent (called new style class) and "class Test" creates old style class (without object parent). Refer [this](#) for more details.

## Does Python support Multiple Inheritance?

Unlike Java and like C++, Python supports multiple inheritance. We specify all parent classes as comma separated list in bracket.

## Travel Smart with InterMiles

Ad Earn & Redeem InterMiles across 8000+ destinations

InterMiles

Sign Up

```
# Python example to show working of multiple
# inheritance
```

```
class Base1(object):
    def __init__(self):
        self.str1 = "Geek1"
        print "Base1"
```

```
class Base2(object):
    def __init__(self):
        self.str2 = "Geek2"
        print "Base2"
```

```
---> class Derived(Base1, Base2):
        def __init__(self):

            # Calling constructors of Base1
            # and Base2 classes
            Base1.__init__(self)
            Base2.__init__(self)
            print "Derived"

        def printStrs(self):
            print(self.str1, self.str2)
```

```
ob = Derived()
ob.printStrs()
```

## Output:

```
Base1
Base2
Derived
('Geek1', 'Geek2')
```



## How to access parent members in a subclass?

### 1. Using Parent class name

```
# Python example to show that base
# class members can be accessed in
# derived class using base class name
class Base(object):

    # Constructor
    def __init__(self, x):
        self.x = x

class Derived(Base):

    # Constructor
    def __init__(self, x, y):
        Base.x = x
        self.y = y

    def printXY(self):

        # print(self.x, self.y) will also work
        print(Base.x, self.y)

# Driver Code
d = Derived(10, 20)
d.printXY()
```

#### Output:

(10, 20)

### 2. Using super()

We can also access parent class members using super.

```
# Python example to show that base
# class members can be accessed in
# derived class using super()
class Base(object):

    # Constructor
    def __init__(self, x):
        self.x = x

class Derived(Base):

    # Constructor
    def __init__(self, x, y):

        ''' In Python 3.x, "super().__init__(name)"
            also works'''
        super(Derived, self).__init__(x)
        self.y = y

    def printXY(self):

        # Note that Base.x won't work here
```



```
# because super() is used in constructor
print(self.x, self.y)

# Driver Code
d = Derived(10, 20)
d.printXY()
```

**Output:**

(10, 20)

Note that the above two methods are not exactly the same. In the next article on inheritance, we will covering following topics.

- 1) How super works? How accessing a member through super and parent class name are different?
- 2) How Diamond problem is handled in Python?

**Exercise:**

Predict the output of following Python programs

1.

```
class X(object):
    def __init__(self, a):
        self.num = a
    def doubleup(self):
        self.num *= 2

class Y(X):
    def __init__(self, a):
        X.__init__(self, a)
    def tripleup(self):
        self.num *= 3
```

```
obj = Y(4)
print(obj.num)
```

```
obj.doubleup()
print(obj.num)
```

```
obj.tripleup()
print(obj.num)
```

**Output:**

4  
8  
24

2. # Base or Super class

```
class Person(object):
    def __init__(self, name):
        self.name = name

    def getName(self):
```



```
        return self.name

    def isEmployee(self):
        return False

# Inherited or Subclass (Note Person in bracket)
class Employee(Person):
    def __init__(self, name, eid):

        ''' In Python 3.0+, "super().__init__(name)"
            also works'''
        super(Employee, self).__init__(name)
        self.empID = eid

    def isEmployee(self):
        return True

    def getID(self):
        return self.empID

# Driver code
emp = Employee("Geek1", "E101")
print(emp.getName(), emp.isEmployee(), emp.getID())
```

**Output:**

('Geek1', True, 'E101')

### Python Programming Tutorial | Object Oriente...



This article is contributed by **Shwetanshu Rohatgi** and **Mayank Rawat**. If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to [contribute@geeksforgeeks.org](mailto:contribute@geeksforgeeks.org). See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above



## Recommended Posts:

[Python | super\(\) in single inheritance](#)

[Python | super\(\) function with multilevel inheritance](#)

[Python issubclass\(\)](#)

[Object Oriented Programming in Python | Set 2 \(Data Hiding and Object Printing\)](#)

[Object Oriented Programming in Python | Set 1 \(Class, Object and Members\)](#)

[Inheritance in Python](#)

[Inheritance in Python | Set 2](#)

[Types of inheritance Python](#)

[Multiple inheritance in Python](#)

[Inheritance in Python Inner Class](#)

[Data Classes in Python | Set 4 \(Inheritance\)](#)

[Method resolution order in Python Inheritance](#)

[Super Keyword in Java](#)

[Python | os.sysconf\\_names object](#)

[Python | object\(\) method](#)

→ **Improved By :** [Ani\\_1107](#), [fridayda13](#)

**Article Tags :** [Python](#) [School Programming](#)



8

3.1

☐ To-do ☐ Done

Based on 17 vote(s)

[Feedback/ Suggest Improvement](#)

[Add Notes](#)

[Improve Article](#)

Please write to us at [contribute@geeksforgeeks.org](mailto:contribute@geeksforgeeks.org) to report any issue with the above content.



Writing code in comment? Please use [ide.geeksforgeeks.org](https://ide.geeksforgeeks.org), generate link and share the link here.

Load Comments

A computer science portal for geeks

5th Floor, A-118,  
Sector-136, Noida, Uttar Pradesh - 201305  
[feedback@geeksforgeeks.org](mailto:feedback@geeksforgeeks.org)

#### COMPANY

About Us  
Careers  
Privacy Policy  
Contact Us

#### PRACTICE

Courses  
Company-wise  
Topic-wise  
How to begin?

#### LEARN

Algorithms  
Data Structures  
Languages  
CS Subjects  
Video Tutorials

#### CONTRIBUTE

Write an Article  
Write Interview Experience  
Internships  
Videos

@geeksforgeeks, Some rights reserved

