**Inheritance in Python**

Inheritance is the capability of one class to derive or inherit the properties from another class. The benefits of inheritance are: 

1. It represents real-world relationships well.
2. It provides **reusability** of a code. We don’t have to write the same code again and again. Also, it allows us to add more features to a class without modifying it.
3. It is transitive in nature, which means that if class B inherits from another class A, then all the subclasses of B would automatically inherit from class A.

**# 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 an employee

def isEmployee(self):

return False

# Inherited or Subclass (Note Person in bracket)

class Employee(Person):

# Here we return true

def isEmployee(self):

return True

# Driver code

emp = Person("Karma") # An Object of Person

print(emp.getName(), emp.isEmployee())

emp = Employee("Dharma") # An Object of Employee

print(emp.getName(), emp.isEmployee())