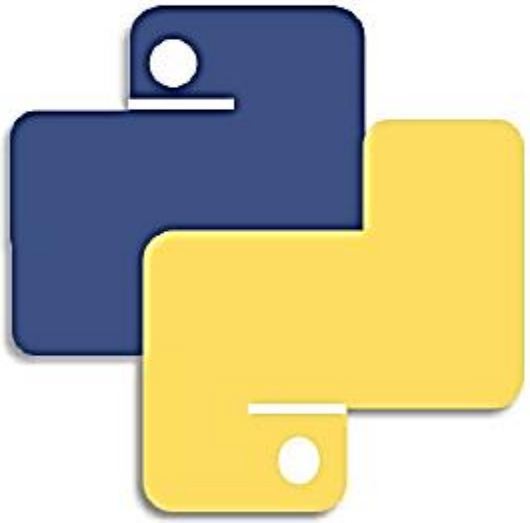
Python

By

Dr. rabih ali sbera

Department of Computer Engineering Techniques-2 nd

2024-2025



Python Programming

Classes &Inheritance

Introduction to Object Oriented Programming in Python

Object Oriented Programming is a way of computer programming using the idea of "objects" to represents data and methods. It is also, an approach used for creating neat and reusable code instead of a redundant one.

What are Classes and Objects?

A class is a collection of objects or you can say it is a blueprint of objects defining the common attributes and behavior. Now the question arises, how do you do that?

Class is defined under a "Class" Keyword.

Example:

```
class class1(): // class 1 is the name of the class
```

Creating an Object and Class in python

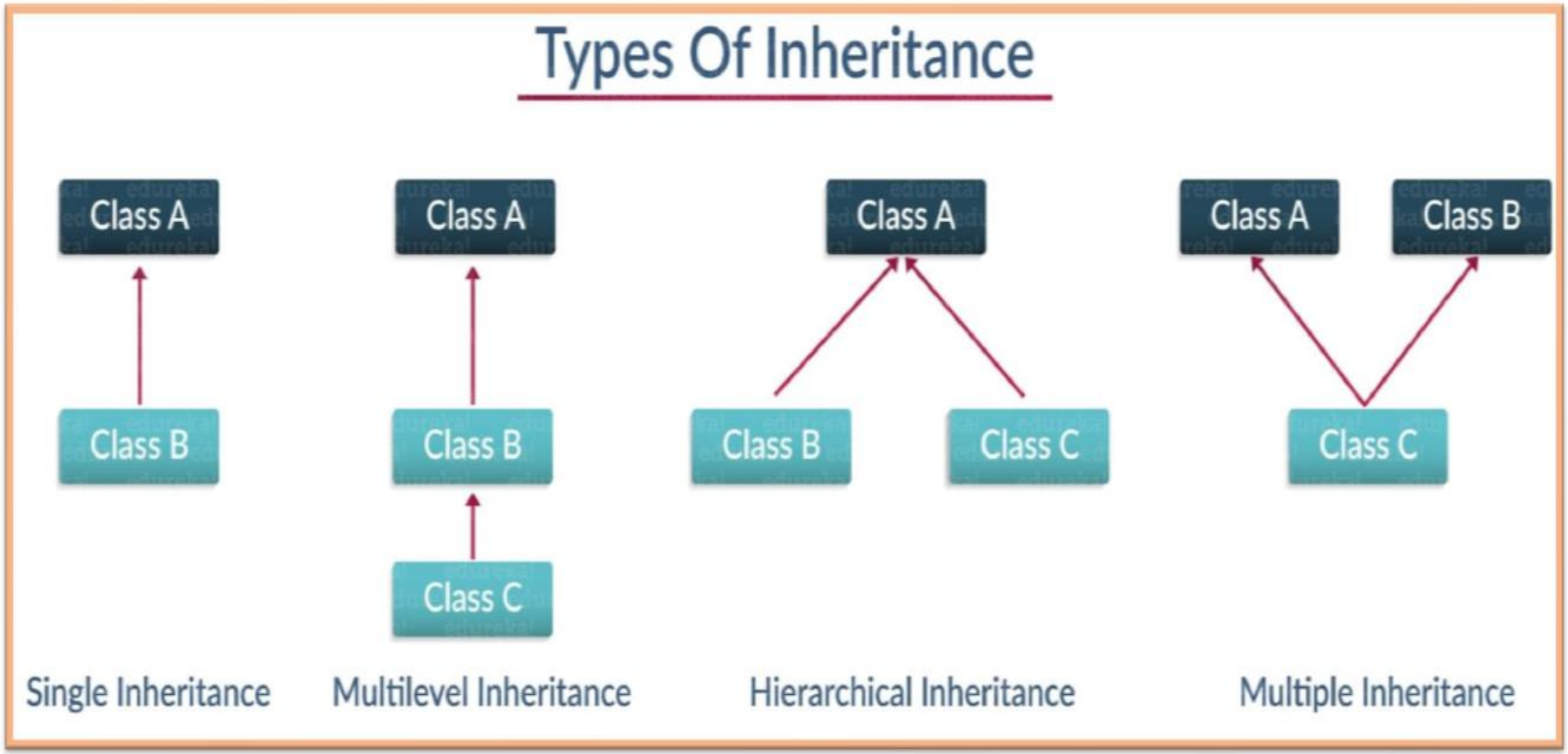
Example:

```
class employee():
    def __init__(self,name,age,id,salary):   //creating a function
        self.name = name // self is an instance of a class
        self.age = age
        self.salary = salary
        self.id = id

emp1 = employee("harshit",22,1000,1234) //creating objects
emp2 = employee("arjun",23,2000,2234)
print(emp1.__dict__)//Prints dictionary
```

Inheritance

Ever heard of this dialogue from relatives “you look exactly like your father/mother” the reason behind this is called ‘inheritance’. From the Programming aspect, It generally means “inheriting or transfer of characteristics from parent to child class without any modification”. The new class is called the derived/child class and the one from which it is derived is called a parent/base class.


Types Of Inheritance
Class A
Class B
Class A
Class B
Class C
Class A
Class B
Class C
Class A
Class B
Class C
Single Inheritance
Multilevel Inheritance
Hierarchical Inheritance
Multiple Inheritance

Single Inheritance

Single level inheritance enables a derived class to inherit characteristics from a single parent class.

Example

```
class employee1()://This is a parent class
    def __init__(self, name, age, salary):
        self.name = name
        self.age = age
        self.salary = salary

class childemployee(employee1)://This is a child class
    def __init__(self, name, age, salary,id):
        self.name = name
        self.age = age
        self.salary = salary
        self.id = id
emp1 = employee1('harshit',22,1000)
print(emp1.age)
```

Output: 22

Multilevel Inheritance

Multi-level inheritance enables a derived class to inherit properties from an immediate parent class which in turn inherits properties from his parent class.

Example:

```
class employee()://Super class
    def __init__(self,name,age,salary):
        self.name = name
        self.age = age
        self.salary = salary
class childemployee1(employee)://First child class
    def __init__(self,name,age,salary):
        self.name = name
        self.age = age
        self.salary = salary
```

Multilevel Inheritance

```
class childemployee2(childemployee1)://Second child class
    def __init__(self, name, age, salary):
        self.name = name
        self.age = age
        self.salary = salary
emp1 = employee('harshit',22,1000)
emp2 = childemployee1('arjun',23,2000)

print(emp1.age)
print(emp2.age)
```

Output: 22,23

Hierarchical Inheritance

Hierarchical level inheritance enables more than one derived class to inherit properties from a parent class.

Example:

```
class employee():
    def __init__(self, name, age, salary):     //Hierarchical Inheritance
        self.name = name
        self.age = age
        self.salary = salary
```

Hierarchical Inheritance

```
class childemployee1(employee):
    def __init__(self,name,age,salary):
        self.name = name
        self.age = age
        self.salary = salary

class childemployee2(employee):
    def __init__(self, name, age, salary):
        self.name = name
        self.age = age
        self.salary = salary
emp1 = employee('harshit',22,1000)
emp2 = employee('arjun',23,2000)
```

Multiple Inheritance

Multiple level inheritance enables one derived class to inherit properties from more than one base class.

Example:

```
class employee1(): //Parent class
    def __init__(self, name, age, salary):
        self.name = name
        self.age = age
        self.salary = salary
```

Multiple Inheritance

```
class employee2(): //Parent class
    def __init__(self,name,age,salary,id):
     self.name = name
     self.age = age
     self.salary = salary
     self.id = id

class childemployee(employee1,employee2):
    def __init__(self, name, age, salary,id):
     self.name = name
     self.age = age
     self.salary = salary
     self.id = id

emp1 = employee1('harshit',22,1000)
emp2 = employee2('arjun',23,2000,1234)
```