Single Inheritance

Single inheritance can be referred to as the most basic inheritance of all types of inheritance in python. As the name suggests in this we derive a single base class from the single parent class or the existing class. The child class will take or inherit all the attributes of the parent class. It can also add its own methods and attributes or override the methods of the parent class.

Code:

```
# Parent Class
class Animal:
    def __init__(self, name):
        self.name = name

    def sound(self):
        pass

# Child Class
class Dog(Animal):
    def sound(self):
        return "Bark!"

dog = Dog("Buddy")
print(dog.name)
print(dog.sound())
```

Output:

```
Buddy
Bark!
```

2)**Multi-level Inheritance**

Multi-level inheritance is a type of inheritance in which a new class is derived from a parent class, which in turn is derived from another parent class.The new class inherits all the attributes and methods of both parent classes. The child class can access all the attributes and methods of both parent classes.

The syntax for multilevel inheritance is given below:

```
class GrandParentClass:

    # Grandparent class definition



class ParentClass(GrandParentClass):

    # Parent class definition



class ChildClass(ParentClass):

    # Child class definition
```

In the above syntax we have named the main class as GrandParentClass and the derived class from it as ParentClass and the new derived class as ChildClass

Example code:

```
# Grandparent Class

class Animal:

   def __init__(self, name):

      self.name = name



   def show_name(self):
```

```
        return self.name


# Parent Class
class Dog(Animal):
    def sound(self):
        return "Bark!"


# Child Class
class Bulldog(Dog):
    def run(self):
        return "Running!"


bulldog = Bulldog("Buddy")
print(bulldog.show_name())
print(bulldog.sound())
print(bulldog.run())
```

Output:

```
Buddy
```

Bark!

Running!

In [ ]: