Inheritance

- The process of getting property of **one class** into **another class** is called **Inheritance.**
- In other word we can say that the process of deriving a new class from an old class is called **inheritance** in which the new class is called **derived** or **child** or **sub** class and old class is called **Base** or **Parent** or **Super** class.
- When a class inherits the property of a class it means it can access all the data member and meber function of that class except private element.
- In this type of programming mainly **two types** of classes are used.
  - **Parent/Super/Base class**
  - **Child/Sub/Derived class**

Parent/Super/Base class

- The class which is inherited by another class is called Parent or Super or Base class.

Child/Sub/Derived class

- The class which inherits the property of another class is called Child or Sub or Derived class.

How to inherit one class into another

Derived class (Base class)

Example

class Square(Rectangle)

Here Square is a Derived class and Rectangle is a Base class.

```
#creating class
class Rectangle:
    #defining function
    def rec_area(self,height,width):
        area=height*width
        print("Area of Rectangle:",area)
#Inheriting Rectangle into Square
class Square(Rectangle):
    # defining function
    def squ_area(self, side):
        area = side*side
        print("Area of Square:", area)
#creating object derived class
obj=Square()
#calling base class function
obj.rec_area(10,20)
#calling derived class function
obj.squ_area(12)
"""
**Output**
Area of Rectangle: 200
Area of Square: 144
"""
```

Single Inheritance

- In this types of inheritance only two classes are used in which one is inherited by another.

```
#creating class
class Rectangle:
    #defining function
    def rec_area(self,height,width):
        area=height*width
        print("Area of Rectangle:",area)
#Inheriting Rectangle into Square
class Square(Rectangle):
    # defining function
    def squ_area(self, side):
        area = side*side
        print("Area of Square:", area)
#creating object derived class
obj=Square()
#calling base class function
obj.rec_area(10,20)
#calling derived class function
obj.squ_area(12)

"""
**Output**
Area of Rectangle: 200
Area of Square: 144
"""
```

Multiple Inheritance

- When two or more than two classes are inherited by a single class simultaneously called multiple inheritance.
- In other word we can say that in this type of inheritance Base class may be two or more than two but derived class should be one.
- In this type of inheritance atleast three class are compulsory.

```
#creating class
class Rectangle:
    #defining function
    def rec_area(self,height,width):
        area=height*width
        print("Area of Rectangle:",area)
#create class
class Square:
    # defining function
    def squ_area(self, side):
        area = side*side
        print("Area of Square:", area)
# Inheriting Rectangle and Square into Triangle
class Triangle(Rectangle,Square):
    # defining function
    def tri_area(self, length,breadth):
        area = 0.5*length*breadth
        print("Area of Triangle:", area)
#creating object derived class
obj=Triangle()
obj.rec_area(10,20)
obj.squ_area(12)
obj.tri_area(12,25)
"""

**Output**
Area of Rectangle: 200
Area of Square: 144
Area of Triangle: 150.0
"""
```

Multilevel Inheritance

- When first class is inherited by second class,second class is inherited by third class and so on called multlevel inheritance.
- In this type of inheritance each derived class is the base class for the next class.
- In this type of inheritance atleast three class are compulsory.

```
#creating class
class Rectangle:
    #defining function
    def rec_area(self,height,width):
        area=height*width
        print("Area of Rectangle:",area)
#Inheriting Rectangle into Square
class Square(Rectangle):
    # defining function
    def squ_area(self, side):
        area = side*side
        print("Area of Square:", area)
# Inheriting  Square into Triangle
class Triangle(Square):
    # defining function
    def tri_area(self, length,breadth):
        area = 0.5*length*breadth
        print("Area of Triangle:", area)
#creating object derived class
obj=Triangle()
obj.rec_area(10,20)
obj.squ_area(12)
obj.tri_area(12,25)

"""
**Output**
Area of Rectangle: 200
Area of Square: 144
Area of Triangle: 150.0
"""
```

Hierarchical Inheritance

- When a single class is inherited by two or more than two classes simultaneously called hierarchical inheritance.
- In other word we can say that in this type of inheritance derived class may be two or more than two but Base class should be one.
- In this type of inheritance atleast three class are compulsory.

```
#creating class
class Rectangle:
    #defining function
    def rec_area(self,height,width):
        area=height*width
        print("Area of Rectangle:",area)
#Inheriting Rectangle into Square
class Square(Rectangle):
    # defining function
    def squ_area(self, side):
        area = side*side
        print("Area of Square:", area)
# Inheriting  Rectangle into Triangle
class Triangle(Rectangle):
    # defining function
    def tri_area(self, length,breadth):
        area = 0.5*length*breadth
        print("Area of Triangle:", area)
#creating object derived class
obj=Triangle()
obj.rec_area(10,20)
obj.tri_area(12,25)

"""
**Output**
Area of Rectangle: 200
Area of Triangle: 150.0
"""
```

Hybrid Inheritance

- The combination of two or more than two inheritance is called Hybrid inheritance.
- It can be combination of any two or more than two inheritance(single,multiple,multilevel,hierarchical).
- In this type of inheritance atleast three class are compulsory.

```
#creating class
class Rectangle:
    #defining function
    def rec_area(self,height,width):
        area=height*width
        print("Area of Rectangle:",area)
#Inheriting Rectangle into Square
class Square:
    # defining function
    def squ_area(self, side):
        area = side*side
        print("Area of Square:", area)
# Inheriting  Rectangle into Triangle
class Triangle(Rectangle,Square):
    # defining function
    def tri_area(self, length,breadth):
        area = 0.5*length*breadth
        print("Area of Triangle:", area)
#inheriting Triangle into Circle
class Circle(Triangle):
    # defining function
    def cir_area(self, radius):
        area = 3.14*radius*radius
        print("Area of Circle:", area)
#creating object derived class
obj=Circle()
obj.rec_area(10,20)
obj.squ_area(13)
obj.tri_area(12,25)
obj.cir_area(2.3)

"""
**Output**
Area of Rectangle: 200
Area of Square: 169
Area of Triangle: 150.0
Area of Circle: 16.610599999999998
"""
```

Advantage of Inheritance

- **Code Reusability**: It means function inside base class is shared by all the derived class.
- **Time Saving**:Because there is no need to define existing property(same code) of a class in another class.
- **Less Cost**:Because existing code is reused, it leads to less development and maintenance costs.
- It helps to reduce **code redundancy.**