Chapter -11
Inheritance & more on OOPs

Inheritance is a way of creating a new class from an existing class.

Syntax:

```
class Employee:          → Base Class
    #codes

class Programmer(Employee):   → Derived or child
    # codes                      class
```

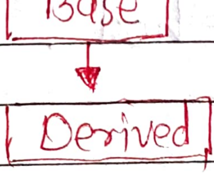
We can use the methods and attributes of employee in Programmer object. Also, we can overwrite or add new attributes and methods in Programmer class.

Types of Inheritance

1> Single inheritance
2> Multiple inheritance
3> Multileved inheritance

Single Inheritance

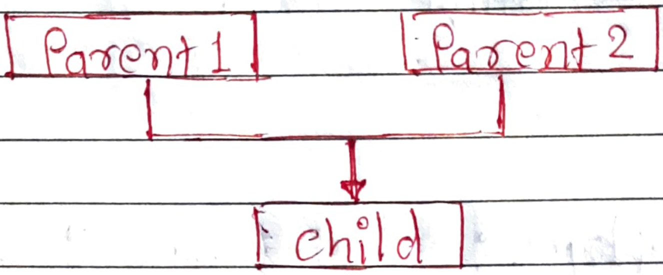
Single inheritance occurs where child class inherits only a single parent class.



DATE :

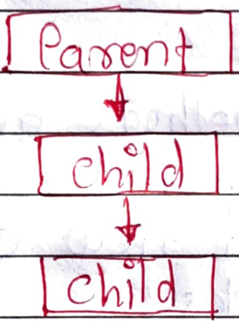
Multiple Inheritance

Multiple inheritance occurs when the child class inherits from more than one parent class.



Multilevel Inheritance

When a child class becomes a parent for another child class.



super() method

super() method is used to access the methods of a super class in the derived class

super().__init__() → calls constructor of the base class

Class Methods

A class method is a method which is bound to the class and not the object of the class. @classmethod decorator is used to create a class method.

Syntax to creat a class method :

```
@classmethod
def (cls, p1 , P2):
        #code
```

@property decorators
Consider the following class

```
class Employee:
        @property
        def name (self):
                return self.name
```

if e = Employee() is an object of class employee. We can print (e.name) to print the ename/ call name() function.

@.getters and @.setters
The method name with @property decorator is called getter method.
We can define a function + @name.setter decorator like below:

```
@name. setter
def name (self, value):
        self. ename = value
```

Operator overloading in Python
Operator in python can be overloaded using dunder methods.

These methods are called when a given operator is used on the objects.

operator in python can be overloaded using the following methods:

$P_1 + P_2$	→	(P_1).__add__(P_2)
$P_1 - P_2$	→	P_1.__sub__(P_2)
$P_1 * P_2$	→	P_1.__mul__(P_2)
P_1 / P_2	→	P_1.__truediv__(P_2)
$P_1 // P_2$	→	P_1.__floordiv__(P_2)

Other dunder/magic methods in Python

__str__() → used to set what gets displayed upon calling str(obj).

__len__() → used for set what gets displayed upon calling __len__() or len(obj).

Chapter - 11
Practice Set

Que 1 Create a class 2-D Vector and use it to create another class representing a 3-D vector.

Que 2. Create a class pets from a class Animals and further Create class Dog from Pets. Add a method bark to class Dog.

Que. 3. Create a class employee and add salary and increment properties to it.
Write a method salaryAfterIncrement method with a @property decorator with a setter which changes the value of increment based on the salary.

Que 4 Write a class Complex to represent Complex numbers, along with overloaded operators + and * which adds and multiplies them.

Que 5. Write a class vector representing a vector of n dimension. Overload the + and * operator which calculates the sum and the dot product of them.

Que 6. Write __str__() method to print the vector as follows:

$$7\hat{i} + 8\hat{j} + 10\hat{k}$$

assume vector of dimension 3 for this problem.

Ques 7. Override the __len__() method on vector of problem 5 to display the dimension of the vector.