Date: Page:

CHAPTER-11: INHERITANCE & MORE ON OOPs

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

Syntax:

```
Class Employee:          → Base Class
    # Code
    ...
Class Programmer(Employee): → Derived or child class
    # Code
```

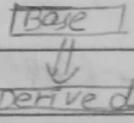
- We can use the methods & attributes of employee in Programmer object.
  Also, we can overwrite or add new attributes and methods in Programer class.

Type of inheritance

1. Single inheritance
2. Multiple inheritance
3. Multilevel inheritance

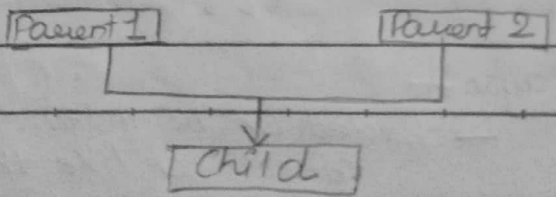
→ Single Inheritance

Single inheritance occurs when child class inherits only a single Parent class



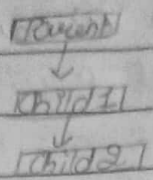
→ Multiple Inheritance

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 method

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 create a class method:

```
@classmethod
def(cls, p1, p2):
    ...
```

@property decorators.

Consider the following class

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

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

Date: _____ Page: _____

→ @getter 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

Operators in python can be overloaded using dunder methods.

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

Operators 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 to set what gets displayed upon calling __len__() or len(obj)