OOP's

① → Class: A class is considered as a blueprint of objects. We can think of the class as a sketch (prototype) of a house.

Object: An object is called an instance of a class.

Example:

```
class Bike:              # create class
    name = " "
    gear = 0

Bike1 = Bike()           # create objects of class
```

② Four pillars of OOP's
  ① Abstraction
  ② Encapsulation
  ③ Inheritance
  ④ Polymorphism

③ Use the __init__ function to assign values to object prob properties, or other operations that are necessary to do when the object is being created.

Example,

```
class person:
    def __init__(self, name, age):    #__init__
        self.name = name               function used
        self.age = age

p1 = person("John", 57)

p1.name
```

John: ← Output

④→ "Self" is used to access other attributes or methods of the object from inside the method.
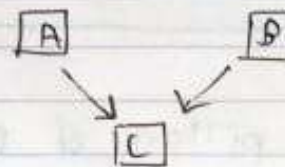
⑤→ Inheritance:
It is defined as the mechanism of inheriting the properties of the base class to the child class.
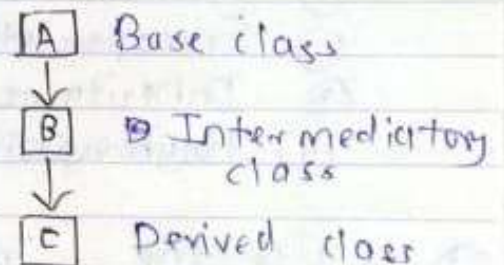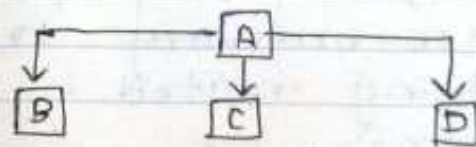
Types of Python Inheritance

① Single Inheritance

```
[A]
 |
 v
[B]
```

② Multiple Inheritance

```
[A]        [B]
   \      /
    v    v
     [C]
```

③ Multilevel Inheritance

```
[A]  Base class
 |
 v
[B]  → Intermediatory
        class
 |
 v
[C]  Derived class
```

④ Hierarchical Inheritance

```
       [A]
      / | \
     v  v  v
   [B] [C] [D]
```

⑤ Hybrid Inheritance

```
         [A]        [B]
        /   \        |
       v     v       v
     [D]    [C]
    /   \
   v     v
 [E]    [F]
```