In [2]:

```
Inheritance
Inheritance is a way of creating new class for using details of existing class without modi
it. The newly formed class is a derived class (or child class). Similarly, the existing cla


 def __init__(self):
        # call super() function
        super().__init__()
        print("# parent class
class Bird:

    def __init__(self):
        print("Bird is ready")

    def whoisThis(self):
        print("Bird")

    def swim(self):
        print("Swim faster")

# child class
class Penguin(Bird):
Penguin is ready")

    def whoisThis(self):
        print("Penguin")

    def run(self):
        print("Run faster")

peggy = Penguin()
peggy.whoisThis()
peggy.swim()
peggy.run()
```

```
  File "<tokenize>", line 25
    def whoisThis(self):
    ^
IndentationError: unindent does not match any outer indentation level
```

In [3]:

```
Encapsulation
Using OOP in Python, we can restrict access to methods and variables. This prevent data
from direct modification which is called encapsulation. In Python, we denote private attrib
using underscore as prefix i.e single “ _ “ or double “ __“.
class Computer:

    def __init__(self):
        self.__maxprice = 900

    def sell(self):
        print("Selling Price: {}".format(self.__maxprice))

    def setMaxPrice(self, price):
        self.__maxprice = price

c = Computer()
c.sell()

# change the price
c.__maxprice = 1000
c.sell()

# using setter function
c.setMaxPrice(1000)
c.sell()
```

```
  File "<ipython-input-3-07a72a3b9edb>", line 2
    Using OOP in Python, we can restrict access to methods and variables. Th
is prevent data
            ^
SyntaxError: invalid syntax
```

In [4]:

```
Polymorphism

parent aur child me same naam k function hna chahiye but arguments shld be different.
class Parrot:

    def fly(self):
        print("Parrot can fly")

    def swim(self):
        print("Parrot can't swim")

class Penguin:

    def fly(self):
        print("Penguin can't fly")

    def swim(self):
        print("Penguin can swim")

# common interface
def flying_test(bird):
    bird.fly()

#instantiate objects
blu = Parrot()
peggy = Penguin()

# passing the object
flying_test(blu)
flying_test(peggy)
```

```
  File "<ipython-input-4-b81adab40dc8>", line 3
    parent aur child me same naam k function hna chahiye but arguments shld
 be different.
             ^
SyntaxError: invalid syntax
```

In [ ]: