In [ ]:

```
#Agenda of the day:
          1. Polymorphism
                - Complie Time - Method Overloading
                -  Run Time    - Method Overriding
          2. Data Abstraction

          3. Data Encapusaltion & Data Hiding
```

In [ ]:

```
#Polymorphism:
       Its refers implementing samething in many ways
poly-many
morphim-forms changing
```

In [42]:

```
#compile-Time Ploymorphism:(method overloading)
class cploy:
    def sums(self,a,b,c=100):
        print(a+b+c)


obj=cploy()

obj.sums(10,50)
```

```
160
```

In [ ]:

```
# Run Time  Polymorphim  - Method Overriding:
#same method name and also same parameters.
```

In [79]:

```
#Example:
class OldGen:     #parent class
    def UsedMobile(self):                    #parent class method
        print("We are used Basic Nokia Mobile")
class NewGen(OldGen):                     #child class
    def UsedMobile(self):                 #child class method
        print("we are using Latest Iphone")
obj=NewGen()
obj.UsedMobile()
```

```
we are using Latest Iphone
```

In [ ]:

```
#Data Abstraction:
Its refers to showing only essential part but hiding the implementing part.
```

In [57]:

```
#Example:
from abc import ABC,abstractmethod
class AbstractClass(ABC):               #abstract class
    @abstractmethod
    def flipkart(self):
        None
    @abstractmethod
    def Amazon(self):
        None
class PrimeUser(AbstractClass):                         #contrete Class
    def flipkart(self):                  #normal classmethods
        print("Ur elibile for Flipkart puls offers")
    def Amazon(self):
        print("Ur elibile for Amazon Prime Offers")
obj = PrimeUser()
obj.flipkart()
obj.Amazon()
```

```
Ur elibile for Flipkart puls offers
Ur elibile for Amazon Prime Offers
```

In [ ]:

```
#Data Encapuslation & Data Hiding:
Encapusaltion is refers to Wrapping up to class methods and variables

Data Hiding is refers to restrict the access of the class data members
Its done by making class variables and methods as private.
```

In [78]:

```
#Example: DataHiding:
#(we are making variables as private by using __(doubleunderscore))
class Security:
    def __init__(self,__a,b):
        self.__a=__a             #private  variable
        self.b=b                 #public variable
    def __info(self,__a,b):      #private class method.
        print(self.__a)
        print("This is Office Info")
    def show(self,__a,b):
        self.__info(__a,b)
a = int(input("enter a value"))
b = int(input("enter b value"))
obj = Security(a,b)
#print(obj.a)
print(obj.b)
obj.show(a,b)
```

```
enter a value500
enter b value100
100
500
This is Office Info
```

In [ ]: