## **Polymorphism**

```
1 Polymorphism is a key principle in OOP, and it promotes code reusability, flexibility and modularity.
          2 It simplifies the writing of code that can work with a variety of objects without needing
          3 to know their specific types, making it a powerful tool for designing and implementation complex
            software systems.
            Polymorphism --- 1.Compile Time 2.Run Time
           1.Compile Time --- 1.Function Overloading 2.Operator Overloading
          7 2.Run Time
                           --- 1.Function Overriding
In [1]:
            class polyDemo:
                def add(self,a=0,b=0,c=0,d=0):
          3
                    print("Addition of Given Numbers: ",a+b+c+d)
In [2]:
          1 a1 = polyDemo()
In [3]:
          1 a1.add(10,20)
        Addition of Given Numbers: 30
In [4]:
          1 a1.add(10,20,30)
        Addition of Given Numbers: 60
          1 a1.add(10,20,30,40)
In [5]:
        Addition of Given Numbers: 100
```

```
In [6]:
           1 a1.add(1,2,3,4,5)
         TypeError
                                                    Traceback (most recent call last)
         Cell In[6], line 1
         ---> 1 a1.add(1,2,3,4,5)
         TypeError: polyDemo.add() takes from 1 to 5 positional arguments but 6 were given
             ##Program for compile Time polymorphism ---Function Overloading
In [15]:
              class SearchBar:
           5
                  def search(self,bname=None,pname=None,price=None):
           6
           7
                      if bname is None and pname is None:
                          print("You are looking for product price of", price ,"Rs.")
           8
           9
                      elif pname is None and price is None:
          10
                          print(bname, "brand is available")
          11
          12
          13
                      elif bname is None and price is None:
          14
                          print("Are you looking for",pname)
          15
          16
                      elif bname is not None and price is not None:
          17
                          print(bname, "with", price, " Rs. is available")
          18
          19
                      elif pname is not None and price is not None:
                          print(pname, "with", price, "Rs. is available")
          20
          21
          22
```

```
In [16]: 1 p1 = SearchBar()
2     p1.search(price=45000)
4     p1.search(bname="apple")
5     p1.search(bname="Xiaomi",price=18000)
6     p1.search(pname="Vivo",price=8900)

You are looking for product price of 45000 Rs.
apple brand is available
Xiaomi with 18000 Rs. is available
Vivo with 8900 Rs. is available
In []: 1
```