

# object oriented programming concepts

```
1 1.objects
2
3 real world entities.
4 has certain state and behaviour
5 eg:student
6 state:
7 name,id,address
8 methods:
9 study,cgpa
10
```

```
1 2.class
2 blueprint of objects
```

```
In [2]: 1 class Student:
2         def __init__(self,name,ids):
3             self.name=name
4             self.id=ids
5         def print_name(self):
6             print(f"name:{self.name}")
7         def print_id(self):
8             print(f"id:{self.id}")
9
10 st1=Student('anu',1)
11 st1.print_name()
```

name:anu

```
In [3]: 1 st2=Student('ravi',4)
2         st2.print_name()
```

name:ravi

```
In [4]: 1 class Student:
2         def __init__(self):
3             self.name=input("enter name:")
4             self.id=input("enter id:")
5         def print_name(self):
6             print(f"name:{self.name}")
7         def print_id(self):
8             print(f"id:{self.id}")
9
10 st1=Student()
11 st1.print_name()
```

enter name:anu  
enter id:1  
name:anu

```
In [5]: 1 class account:
2         def __init__(self,acct_num,acct_blnc):
3             self.acct=acct_num
4             self.blnc=acct_blnc
5         def withdraw(self,amount):
6             if self.blnc-amount>=0:
7                 print("you can withdraw the ampunt")
8                 self.blnc=self.blnc-amount
9             else:
10                print("withdrawal failed")
11         def display(self):
12             print(self.acct,"your balnce is ",self.blnc)
13
14 acct1=account(12334435,10000)
15 acct1.withdraw(5000)
16
```

you can withdraw the ampunt

```
In [7]: 1 acct1.display()
```

12334435 your balnce is 0

```
In [9]: 1 class Animal:
2         def __init__(self):
3             self.name=input("enter name:")
4             self.color=input('color:')
5         def walk(self):
6             print("animal can walk")
7 a1=Animal()
8 print(a1.name)
9 a1.walk()
```

enter name:dog  
color:black  
dog  
animal can walk

```
In [ ]: 1
```