Name: - L Prathyusha

Python object oriented programming-OOPs (combination of data and code)

In [2]:

```
class Student:
    '''this is student class with required data'''
print(Student.__doc__)
help(Student)
```

```
this is student class with required data
Help on class Student in module __main__:

class Student(builtins.object)
 |  this is student class with required data
 |
 |  Data descriptors defined here:
 |
 |  __dict__
 |      dictionary for instance variables (if defined)
 |
 |  __weakref__
 |      list of weak references to the object (if defined)
```

In [3]:

```
class Student:
    '''developed by durga for python demo'''

    def __init__(self,name,age,marks,rollno):
        self.name=name
        self.age=age
        self.marks=marks
        self.rollno=rollno

    def talk(self):
        print("hello i am :",self.name)
        print("my age is:",self.age)
        print("my marks are:",self.marks)
        print("my roll no. is:",self.rollno)

s1=Student("Prathyusha",20,90,19111344)
s1.talk()

s2=Student("lala",20,88,19111112)
s2.talk()
```

```
hello i am : Prathyusha
my age is: 20
my marks are: 90
my roll no. is: 19111344
hello i am : lala
my age is: 20
my marks are: 88
my roll no. is: 19111112
```

In [4]:

```
class employee:

    def __init__(self):
        self.eno=100
        self.enamme='prathyusha'
        self.esal=10000

e=employee()
print(e.__dict__)
```

```
{'eno': 100, 'enamme': 'prathyusha', 'esal': 10000}
```

In [5]:

```
class test:
    def __init__(self):
        self.a=10
        self.b=20

    def m1(self):
        self.c=30

t=test()
t.m1()
t.d=40
print(t.__dict__)
```

```
{'a': 10, 'b': 20, 'c': 30, 'd': 40}
```

In [8]:

```
import sys
class Customer:
    '''customer class with bank operations'''
    bankname="Durgabank"

    def __init__(self,name):
        self.name=name
        print(self.name)
        self.balance=0

    def deposit(self,amt):
        print(amt)
        self.balance=self.balance + amt
        print('After deposit the balance:',self.balance)

    def withdraw(self,amt):
        print(self.balance)
        if amt>self.balance:
            print('Insufficent funds..cannot perform this operation')
            sys.exit()
        self.balance=self.balance-amt
        print('After withdraw the balance:',self.balance)

print('Welcome to', Customer.bankname)
name=input('Enter your Name:')

c=Customer(name)


while True:
    print('d-Deposit\nw-Withdraw\ne-exit')
    option=input('choose your option:')
    if option =='d'or option =='D':
        amt=float(input('Enter the amount to deposit:'))
        c.deposit(amt)

    elif option =='w'or option =='W':
        amt=float(input('Enter Amount to withdraw:'))
        c.withdraw(amt)

    elif option =='e'or option =='E':
        print('Thanks for Banking')
        sys.exit()
    else:
        print('Invalid option..plz choose valid option')
```

```
Welcome to Durgabank
Enter your Name:P
P
d-Deposit
w-Withdraw
e-exit
choose your option:d
Enter the amount to deposit:100
100.0
After deposit the balance: 100.0
d-Deposit
```

```
w-Withdraw
e-exit
choose your option:e
Thanks for Banking

An exception has occurred, use %tb to see the full traceback.

SystemExit



C:\Users\Prathyu Lachireddy\anaconda3\lib\site-packages\IPython\core\interac
tiveshell.py:3426: UserWarning: To exit: use 'exit', 'quit', or Ctrl-D.
  warn("To exit: use 'exit', 'quit', or Ctrl-D.", stacklevel=1)
```

In [7]:

```python
class test:
    x=10
    def __init__(self):
        self.y=20

t1=test()
t2=test()
print('t1',t1.x,t1.y)
print('t2',t1.x,t1.y)
test.x=888
t1.y=999
print('t1',t1.x,t1.y)
print('t2',t1.x,t1.y)
```

```
t1 10 20
t2 10 20
t1 888 999
t2 888 999
```

In [ ]:

```

```