**Class**

**Collection of COmmon attributes and Methods**

**Def : Logical Structure how Object looks**

**Class will not occupy,**

**once object is created, memory allocated to data variable, but not functions**

**Functions are used to process to Data**

**Functions are called using OBjects**

**OOP Features**

**Encapsulation, Adv : security (Only Class Members can access data)**

**Polymorphism, Adv: Easy to Process (No Need to Remember method names)**

**Abstraction Adv: Complexity Decreases ( Hiding How method is implemented)**

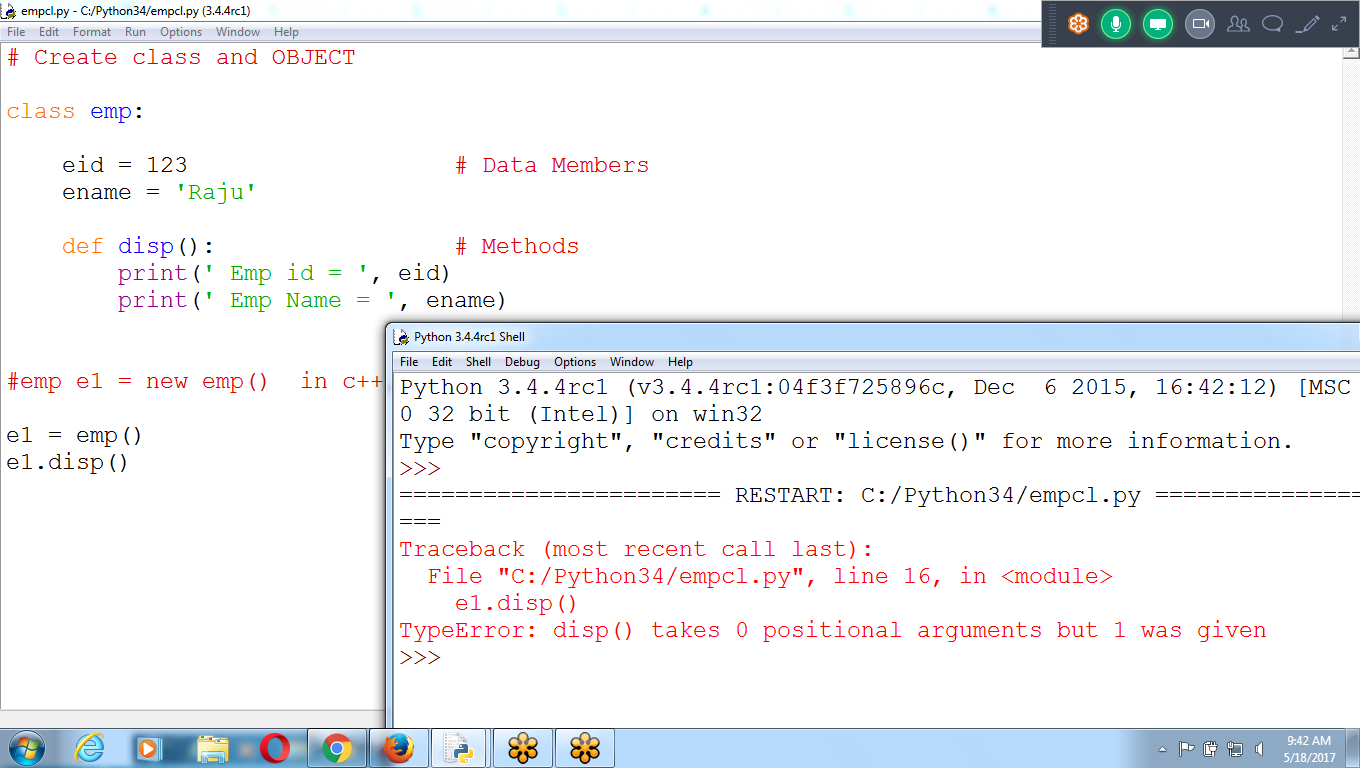
**Inheritance Adv: Reusability ( Extend or Modify existing class features)**

**Create a class and call method using object**

**IN a Method displaying Eid and Ename**

**Error: which eid and ename**

**Creating emp class with two Data members and one disp function**

****

**# Create class and OBJECT**

**class emp:**

**eid = 123 # Data Members**

**ename = 'Raju'**

**def disp(): # Methods**

**print(' Emp id = ', eid)**

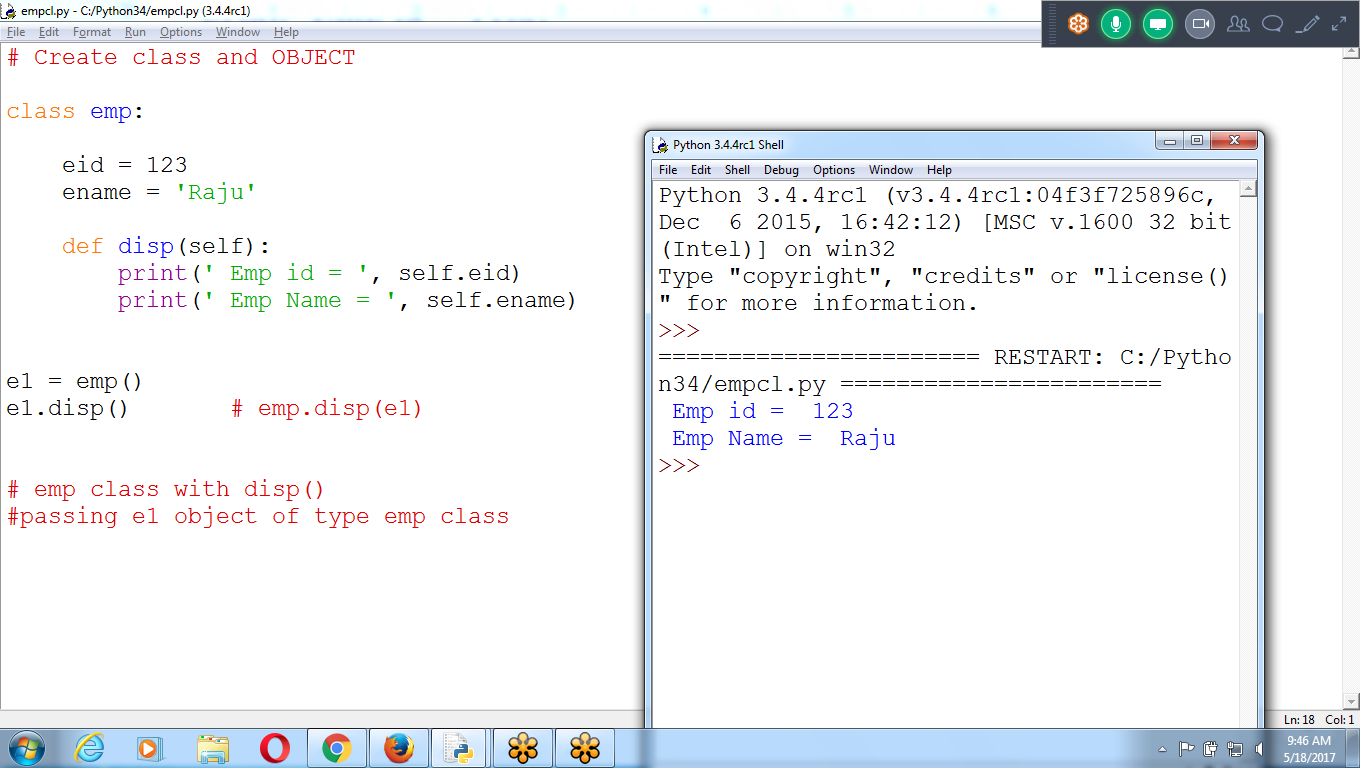
**print(' Emp Name = ', ename)**

**#emp e1 = new emp() in c++, java**

**e1 = emp()**

**e1.disp()**

**Program 3: Declare self or any object name in function disp()**



# Create class and OBJECT

class emp:

eid = 123

ename = 'Raju'

def disp(self):

print(' Emp id = ', self.eid)

print(' Emp Name = ', self.ename)

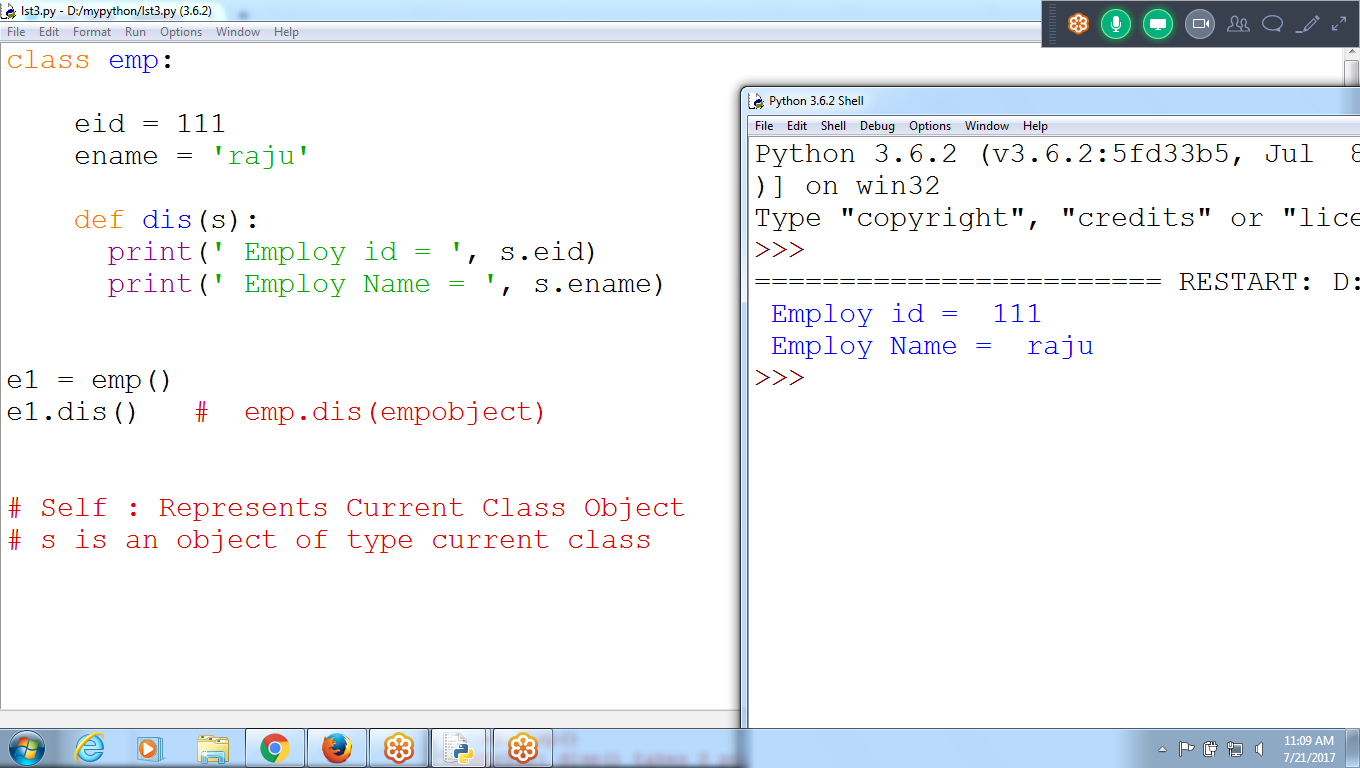
e1 = emp()

e1.disp() **# emp.disp(e1)**

# emp class with disp()

#passing e1 object of type emp class

E1 object passed to object s in dis()



class emp:

eid = 111

ename = 'raju'

**def dis(s):**

print(' Employ id = ', s.eid)

print(' Employ Name = ', s.ename)

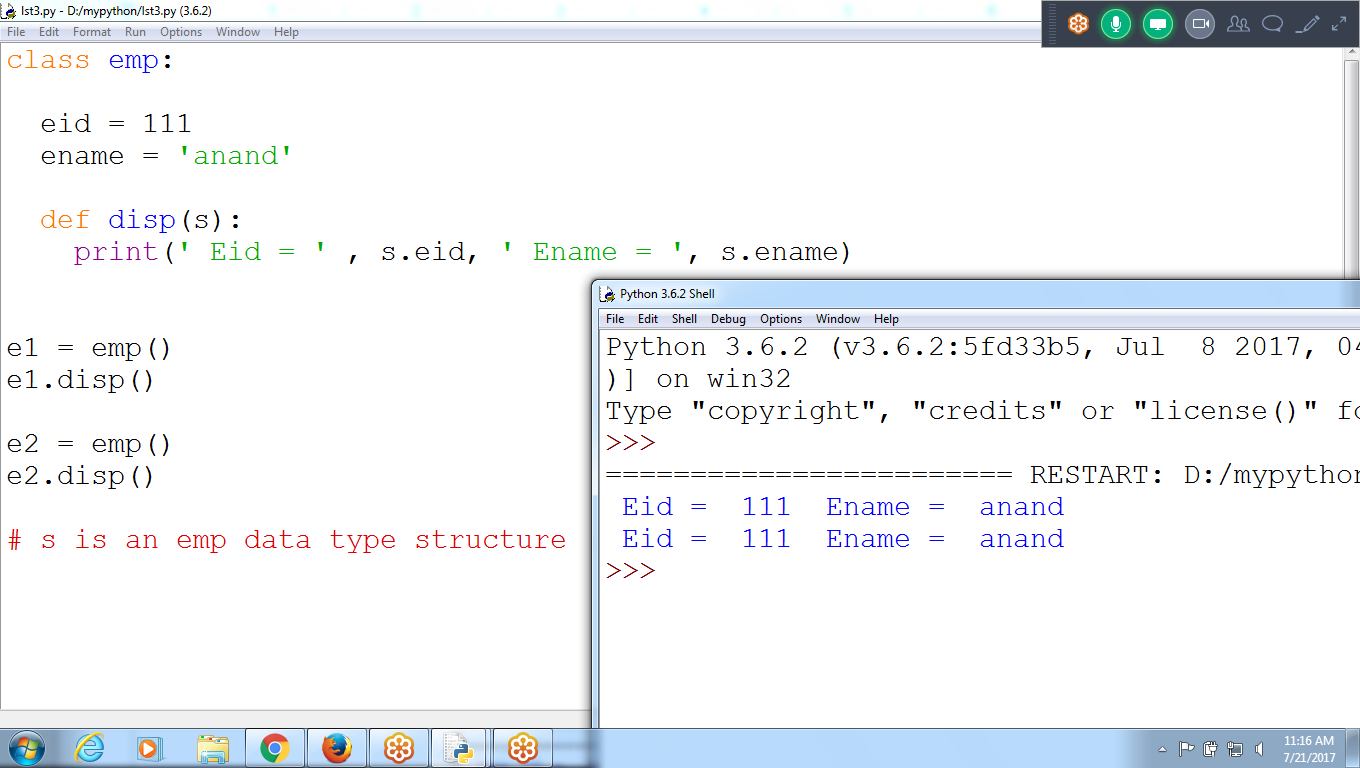
e1 = emp()

e1.dis() # emp.dis(empobject)

# Self : Represents Current Class Object

# s is an object of type current class

**3: Creating Multiple Objects**



**class emp:**

**eid = 111**

**ename = 'anand'**

**def disp(s):**

**print(' Eid = ' , s.eid, ' Ename = ', s.ename)**

**e1 = emp()**

**e1.disp()**

**e2 = emp()**

**e2.disp()**

**# s is an emp data type structure**

**While Creating MULTIPLE OBjects**

**When Object is Created, Memory is allocated to Data Members**

e1 = emp()

E1 contains starting address of the object of two values eid and ename::

Does not contains method related

Allocates memory to Data Members

Eid ename

111 anand

6000(address)

E1 object contains Starting address of Memory block for eid and ename values

E1

**6000**

4000 to 4001

Eid ename

111 anand

**6000**

a= 24 # a contains address of the value 24 existing

St = ‘python program ‘ # contains starting address where value ‘Python program” is existing

>>> a =24

>>> type(a)

**<class 'int'>**

>>> st = 'python program '

>>> type(st)

**<class 'str'>**

>>>

e2 = emp()

E2 referring separate memory block for e2 object

Eid ename

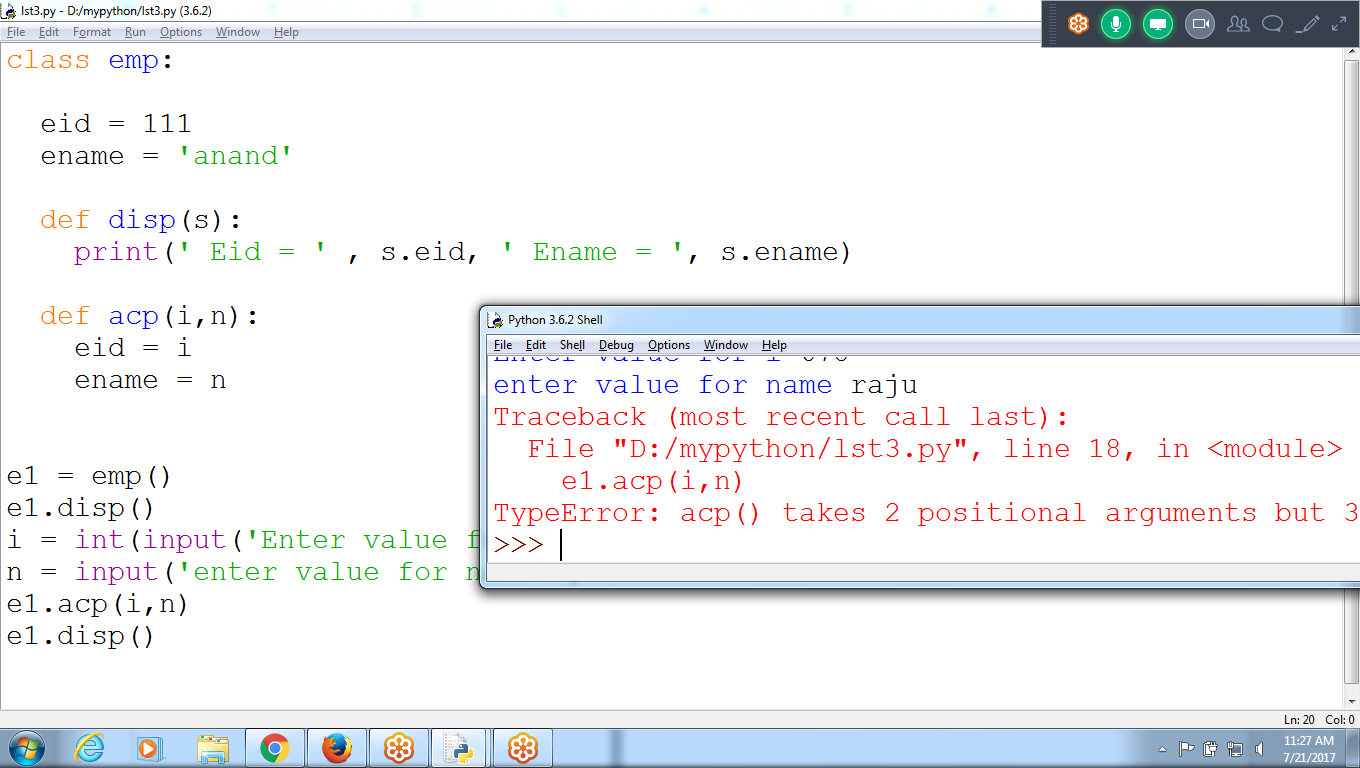
111 anand

e1.disp() :: Moves to e1 object data members

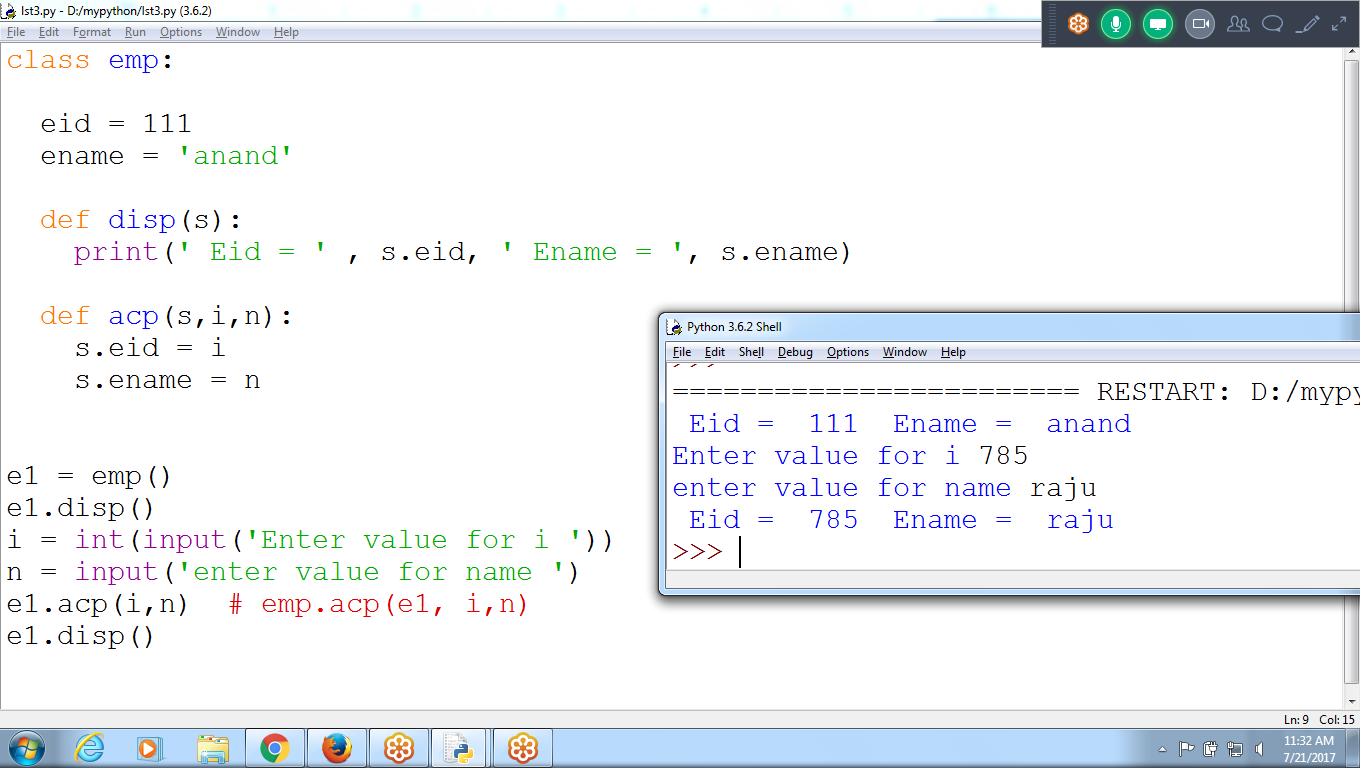
e2.disp()

Assigning value to data members using acp()

acp() requires 3 parameters



Passing Parameter to a Method



class emp:

eid = 111

ename = 'anand'

def disp(s):

print(' Eid = ' , s.eid, ' Ename = ', s.ename)

def acp(s,i,n):

s.eid = i

s.ename = n

e1 = emp()

e1.disp()

i = int(input('Enter value for i '))

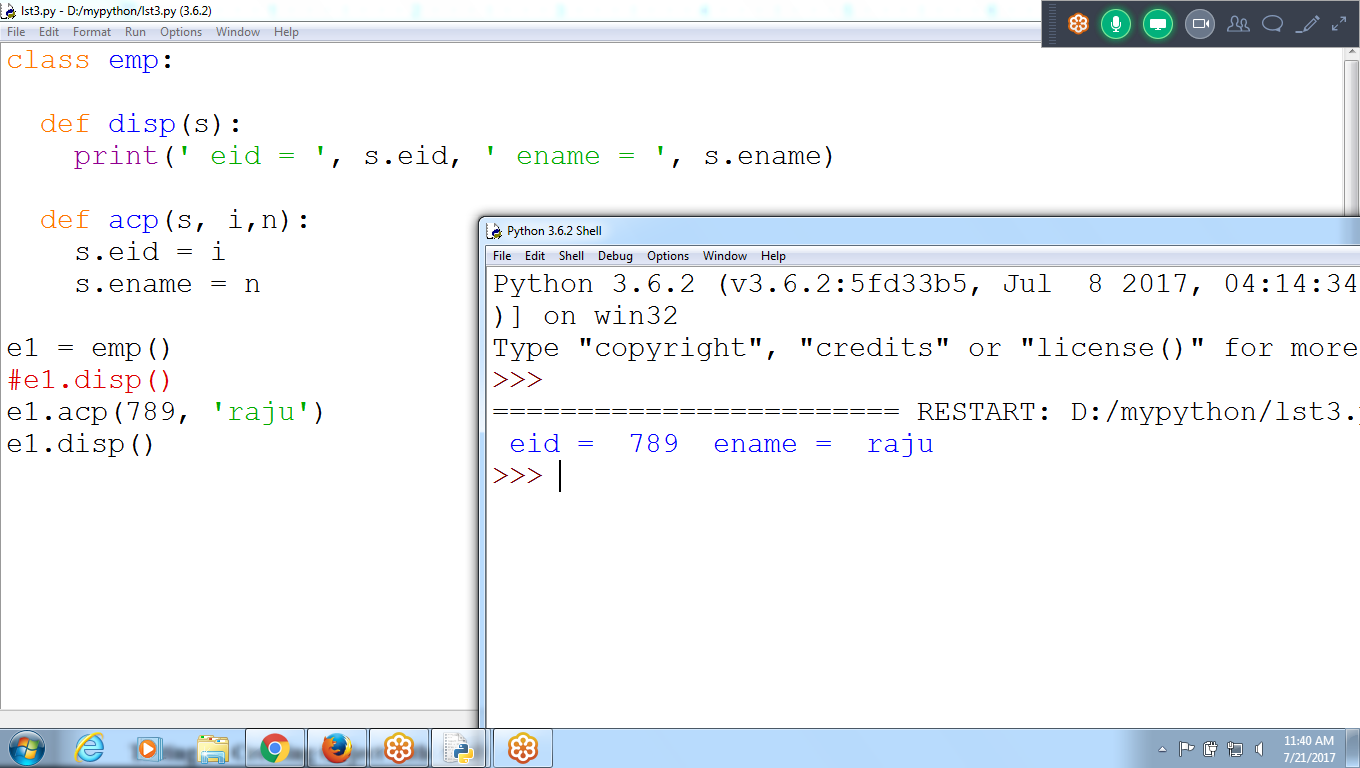
n = input('enter value for name ')

e1.acp(i,n) # emp.acp(e1, i,n)

e1.disp()

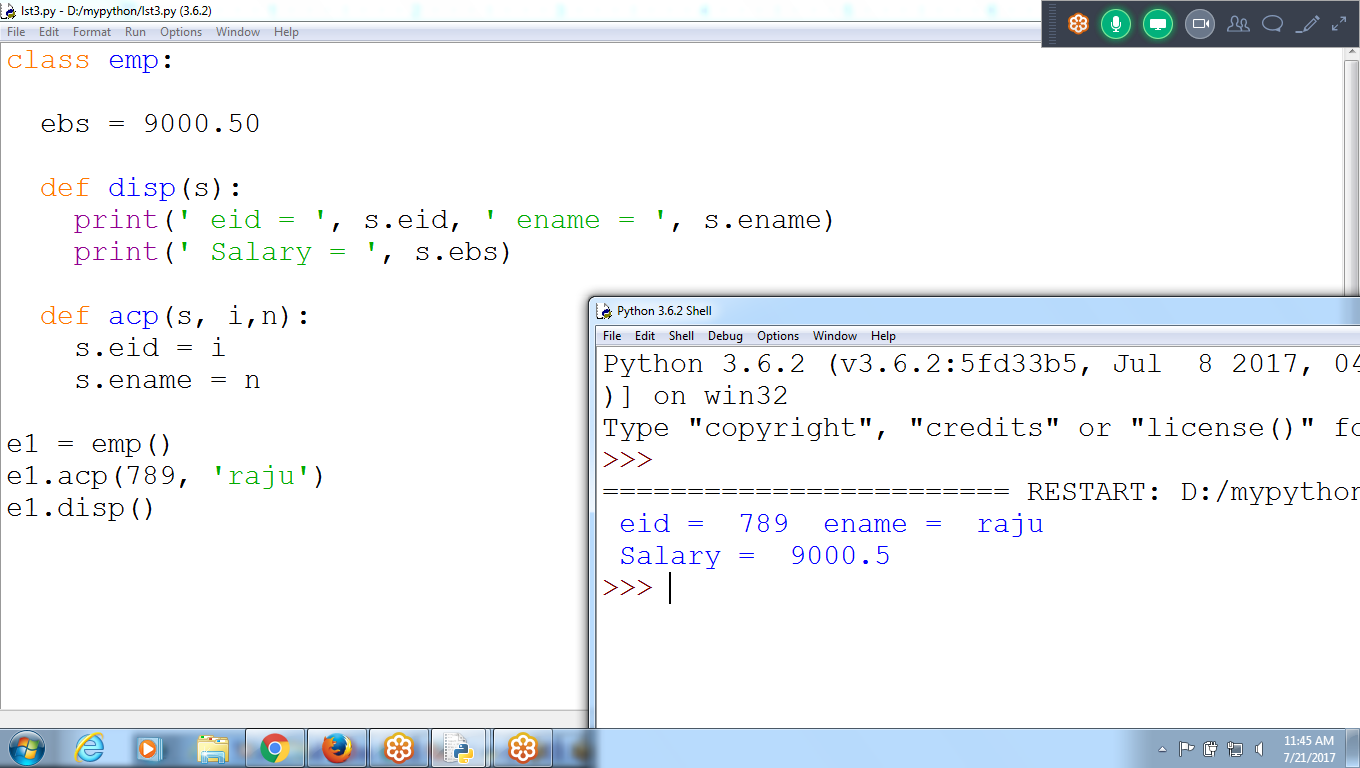
Data members or variable not Declared, ie. eid and ename

At the time calling acp(), eid and ename variables are created



Object e1 contains ebs

While passing values to acp(), eid and ename are created to object e1

****

E1 = emp() # object created allocated memory for data members

E1 occupies two bytes memory, contains starting address of object data members

E1

**6000**

4000 to 4001

Eid ename

111 anand

**6000**

e1.disp()

Result: Eid =111 Ename = anand

e1.acp(999, ‘chaitu’)

Replace default values with 999, ‘chaitu’

Eid ename

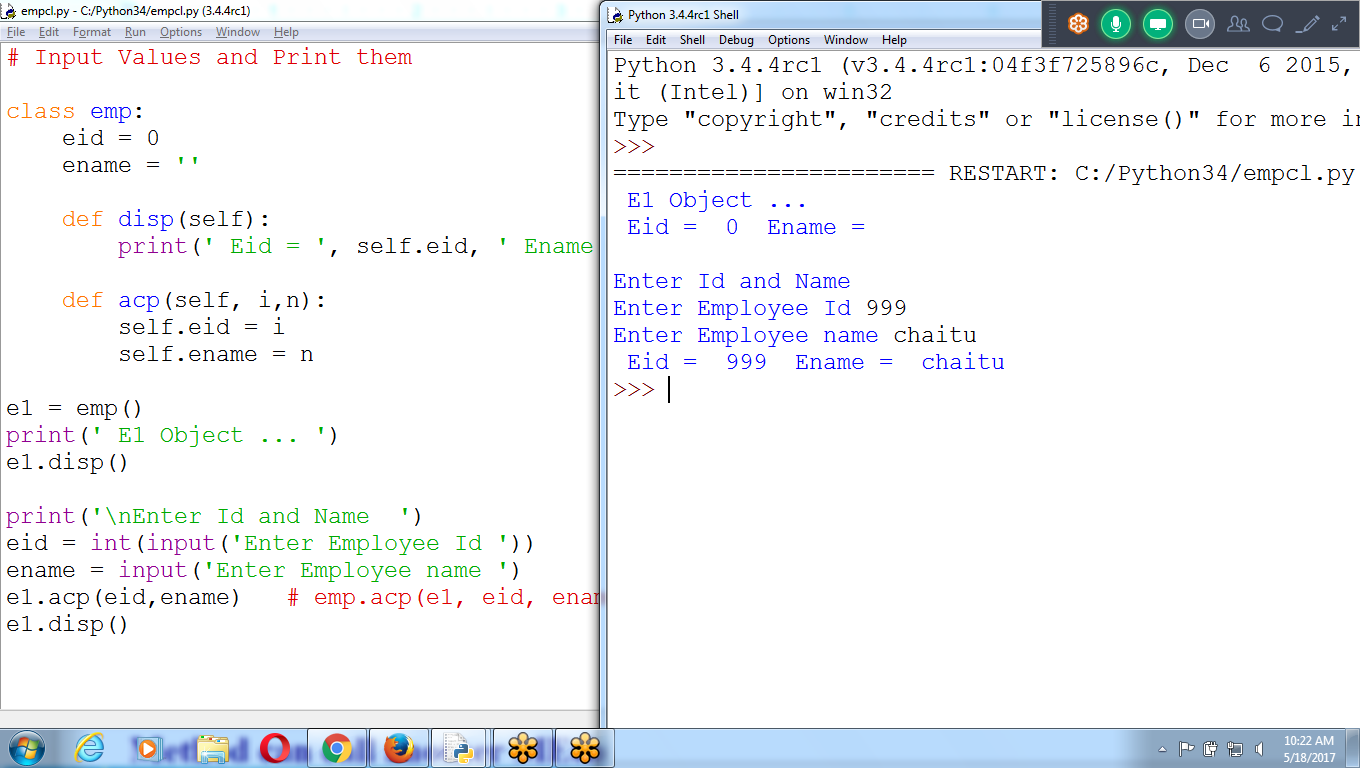
999 chaitu

**6000**

e1.disp()

Displays Current object value

**Accept values and Print them**

****

**# Input Values and Print them**

**class emp:**

**eid = 0**

**ename = ''**

**def disp(self):**

**print(' Eid = ', self.eid, ' Ename = ', self.ename)**

**def acp(self, i,n):**

**self.eid = i**

**self.ename = n**

**e1 = emp()**

**print(' E1 Object ... ')**

**e1.disp()**

**print('\nEnter Id and Name ')**

**eid = int(input('Enter Employee Id '))**

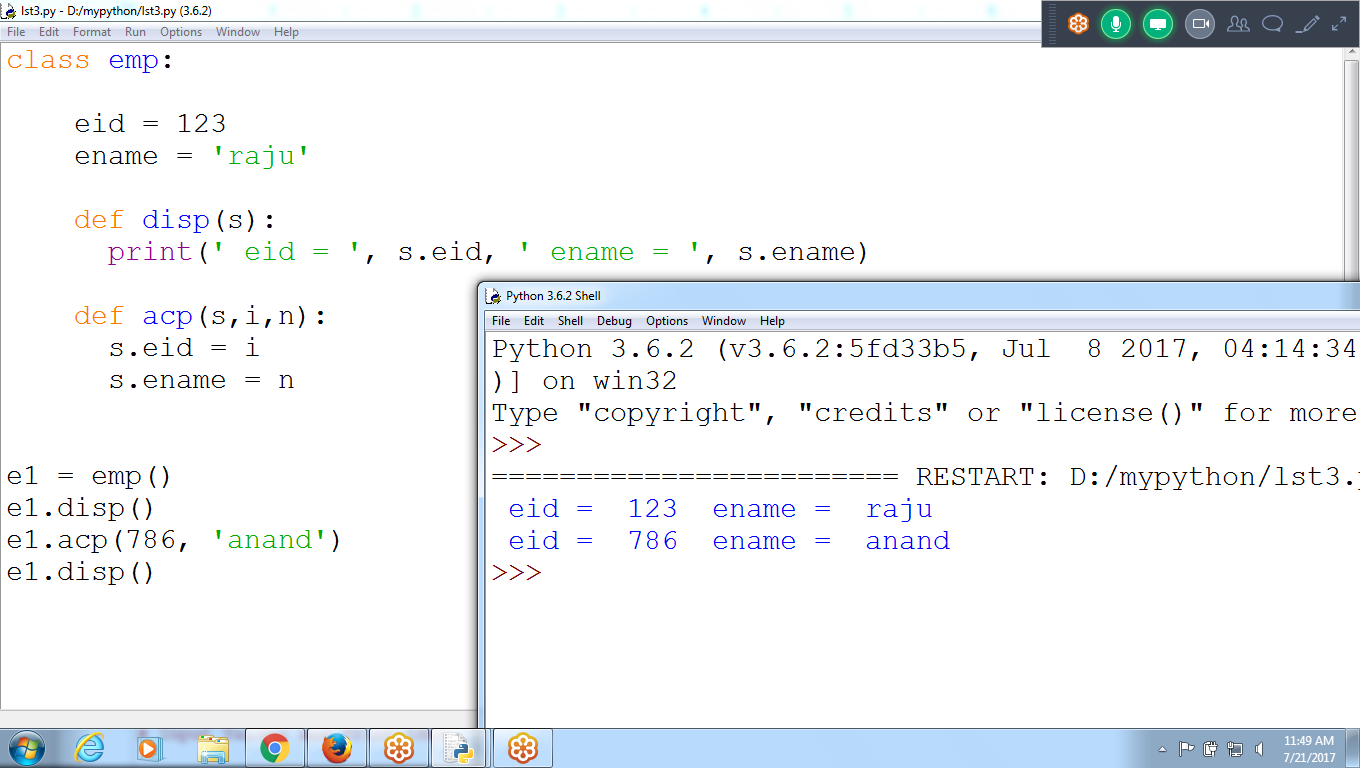
**ename = input('Enter Employee name ')**

**e1.acp(eid,ename) # emp.acp(e1, eid, ename)**

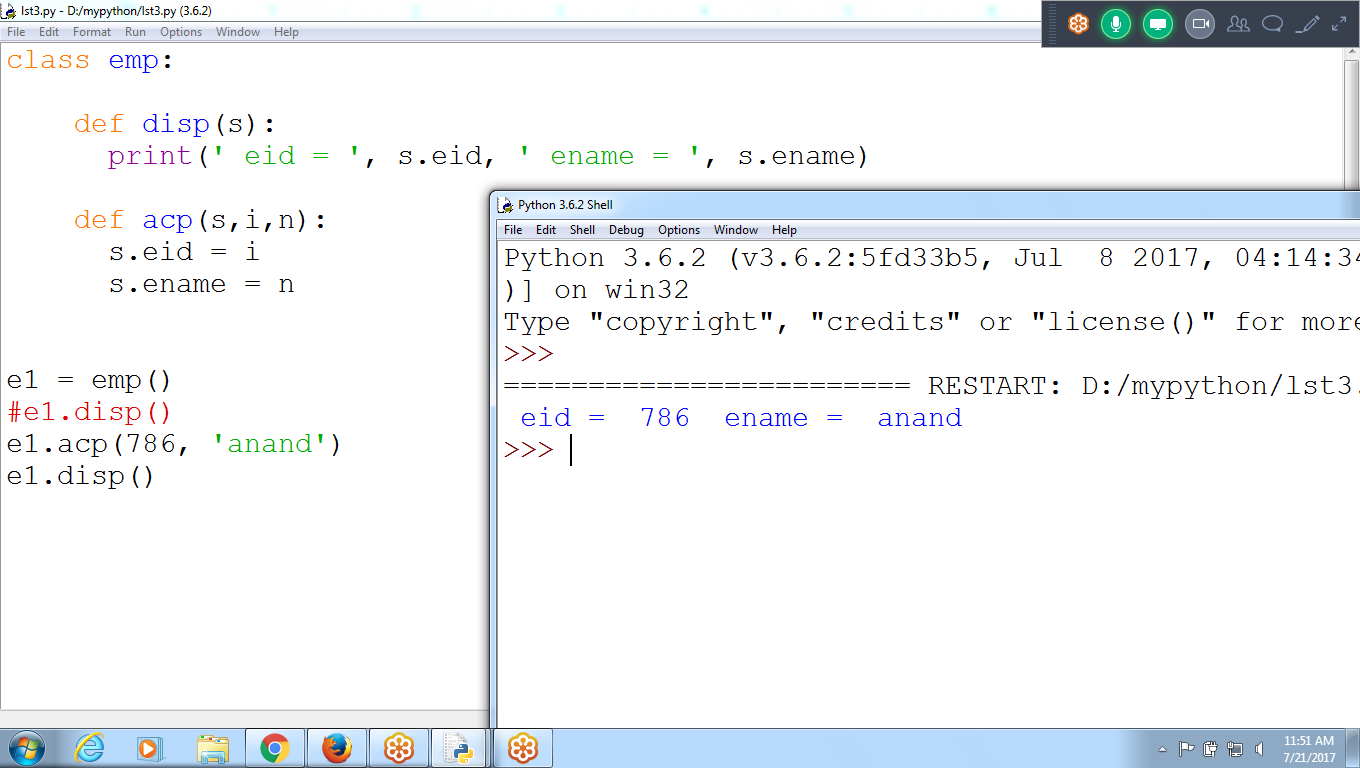
**e1.disp()**

**Example 1:**

**Creating Object, displaying and Accepting value**

****

**Example 2: Without data members**

****

**class emp:**

**def disp(s):**

**print(' eid = ', s.eid, ' ename = ', s.ename)**

**def acp(s,i,n):**

**s.eid = i**

**s.ename = n**

**e1 = emp()**

**#e1.disp()**

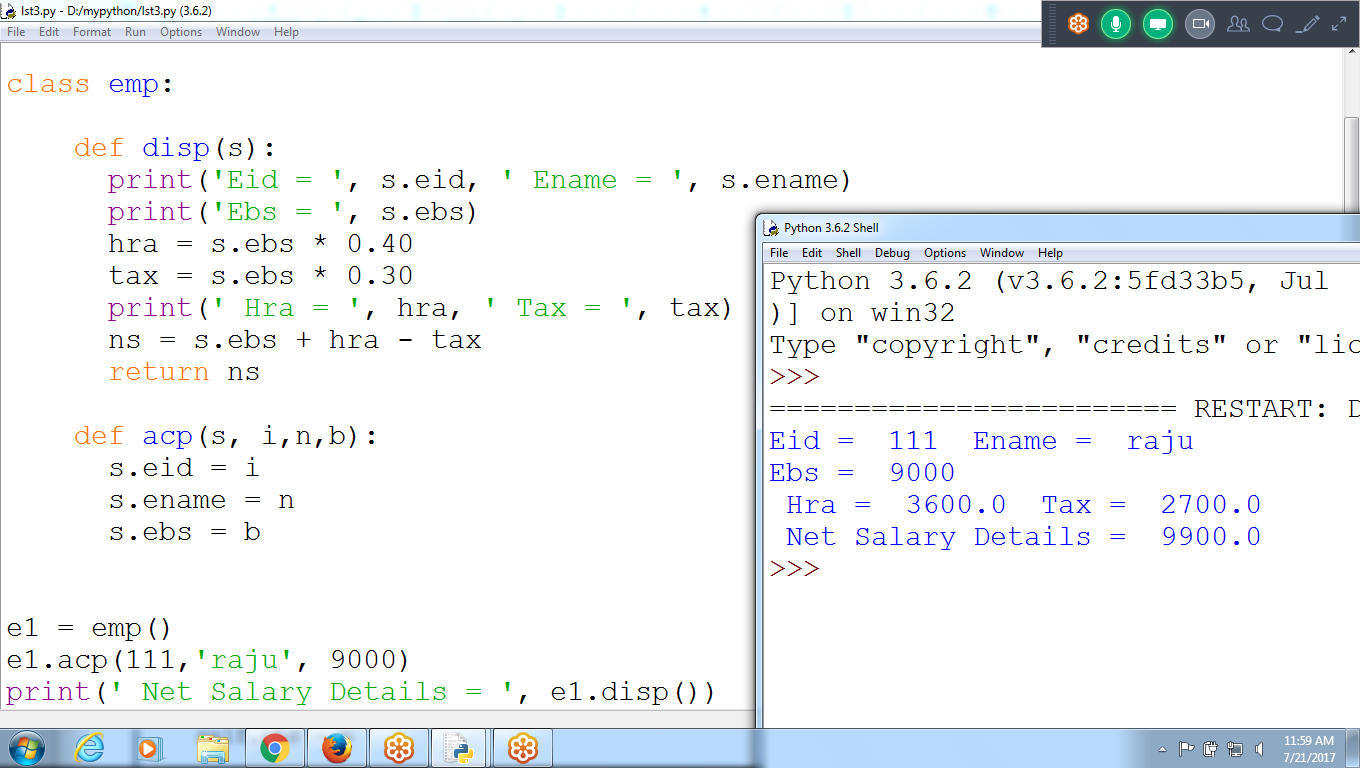
**e1.acp(786, 'anand')**

**e1.disp()**

**# Create class, object, methods and return Net Salary**

**Eid,ename,ebs are object variables**

**Hra, tax, ns are local to disp()**

****

**# Program to accept eid, ename, ebs**

**# PRint HRA, Tax and Net salary**

**class emp:**

**def disp(s):**

**print('Eid = ', s.eid, ' Ename = ', s.ename)**

**print('Ebs = ', s.ebs)**

**hra = s.ebs \* 0.40**

**tax = s.ebs \* 0.30**

**print(' Hra = ', hra, ' Tax = ', tax)**

**ns = s.ebs + hra - tax**

**return ns**

**def acp(s, i,n,b):**

**s.eid = i**

**s.ename = n**

**s.ebs = b**

**e1 = emp()**

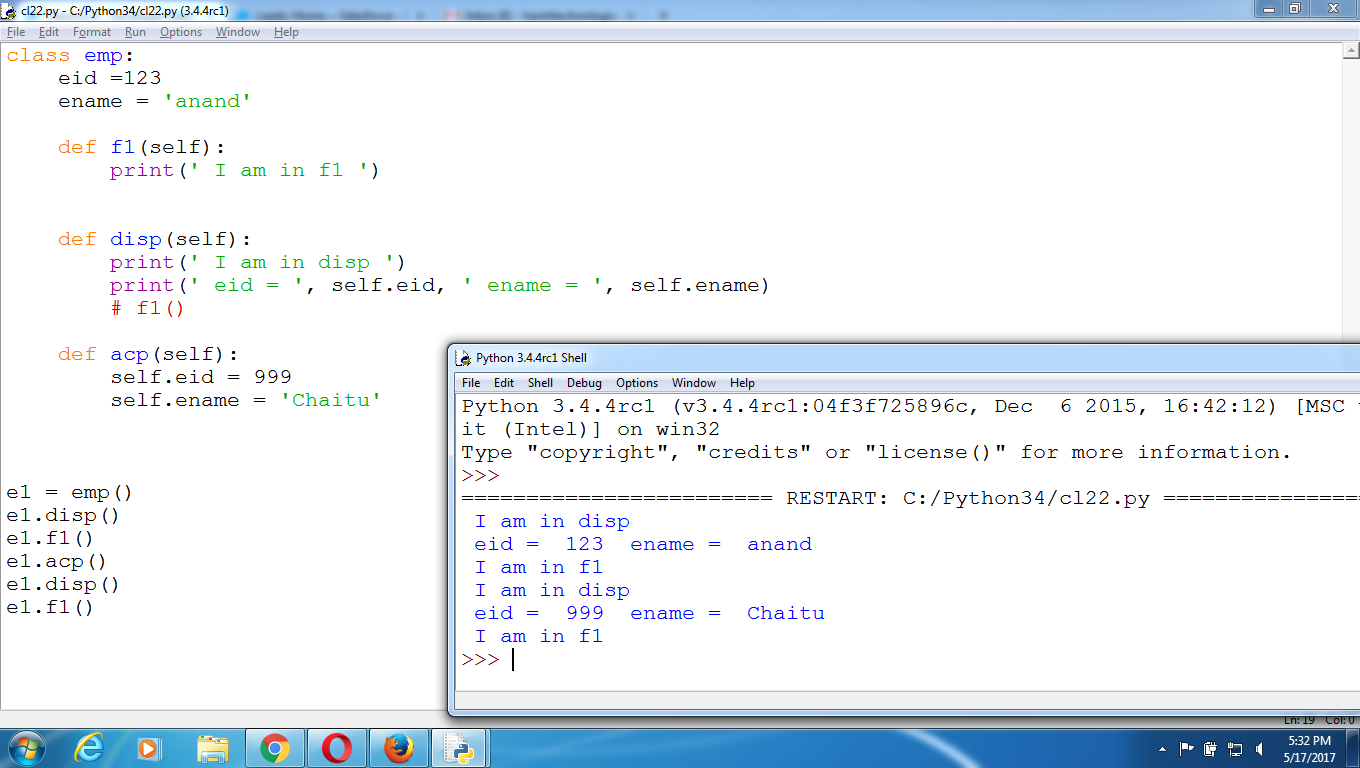
**e1.acp(111,'raju', 9000)**

**print(' Net Salary Details = ', e1.disp())**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Method can call another MEthod**



class emp:

eid =123

ename = 'anand'

def f1(self):

print(' I am in f1 ')

def disp(self):

print(' I am in disp ')

print(' eid = ', self.eid, ' ename = ', self.ename)

# f1()

def acp(self):

self.eid = 999

self.ename = 'Chaitu'

e1 = emp()

e1.disp()

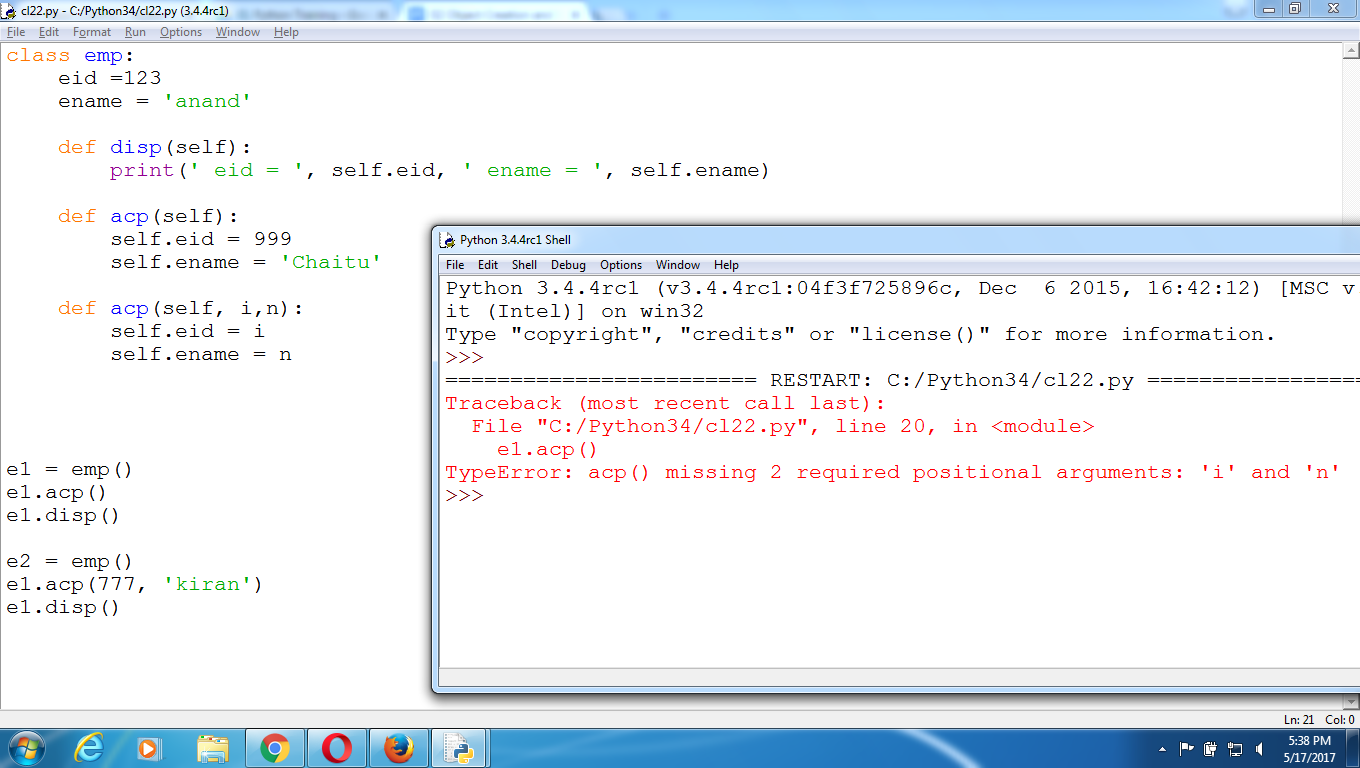
e1.f1()

e1.acp()

e1.disp()

e1.f1()

**Multiple Objects and Multiple Method with Parameters**

****

class emp:

eid =123

ename = 'anand'

def disp(self):

print(' eid = ', self.eid, ' ename = ', self.ename)

def acp(self):

self.eid = 999

self.ename = 'Chaitu'

def acp(self, i,n):

self.eid = i

self.ename = n

e1 = emp()

e1.acp()

e1.disp()

e2 = emp()

e1.acp(777, 'kiran')

e1.disp()

## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## 

## 

## 