………………………………………………………………………………………………

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Chapter-16\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*VARIABLES\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*:-

………………………………………………………………………………………………

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Types of Variables:\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*:-

…………………………………………………………………………………………….

Inside Python class 3 types of variables are allowed.

1. Instance Variables (Object Level Variables)

2. Static Variables (Class Level Variables)

3. Local variables (Method Level Variables)

……………………………………………………………………..

1. Instance Variables: -

If the value of a variable is varied from object to object, then such type of variables are called instance variables.

For every object a separate copy of instance variables will be created.

………………………………………………………………..

Where we can declare Instance variables:

1. Inside Constructor by using self variable

2. Inside Instance Method by using self variable

3. Outside of the class by using object reference variable

…………………………………………………………………………….

1. Inside Constructor by using self variable:-

………………………………………………………………………………….. We can declare instance variables inside a constructor by using self keyword. Once we creates object, automatically these variables will be added to the object.

………………………………………………………………………

Example:

1) class Employee:

2)

3) def \_\_init\_\_(self):

4) self.eno=100

5) self.ename=' prasanna '

6) self.esal=10000

7)

8) e=Employee()

9) print(e.\_\_dict\_\_)

Output: {'eno': 100, 'ename': 'prasanna', 'esal': 10000}

………………………………………………………………………………

1. Inside Instance Method by using self variable: -

……………………………………………………………………….

We can also declare instance variables inside instance method by using self variable. If any instance variable declared inside instance method, that instance variable will be added once we call taht method.

………………………………………………………………

Example:

1) class Test:

2)

3) def \_\_init\_\_(self):

4) self.a=10

5) self.b=20

6)

7) def m1(self):

8) self.c=30

9)

10) t=Test()

11) t.m1()

12) print(t.\_\_dict\_\_)

………………………………………………………………

Output {'a': 10, 'b': 20, 'c': 30}

………………………………………………………………………………..

3. Outside of the class by using object reference variable:

……………………………………………………………………………………

We can also add instance variables outside of a class to a particular object.

1) class Test:

2)

3) def \_\_init\_\_(self):

4) self.a=10

5) self.b=20

6)

7) def m1(self):

8) self.c=30

9)

10) t=Test()

11) t.m1()

12) t.d=40

13) print(t.\_\_dict\_\_)

…………………………..

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

………………………………………………………………………….

\*\*\*\*\*\*\*How to access Instance variables:\*\*\*\*\*\*\*\*:-

………………………………………………………………………………….

We can access instance variables with in the class by using self variable and outside of the class by using object reference.

1) class Test:

2)

3) def \_\_init\_\_(self):

4) self.a=10

5) self.b=20

6)

7) def display(self):

8) print(self.a)

9) print(self.b)

10)

11) t=Test()

12) t.display()

13) print(t.a,t.b)

……………………………..

Output 10 20 10 20

…………………………………………………………………………………

How to delete instance variable from the object: -

…………………………………………………………………………

1. Within a class we can delete instance variable as follows

del self.variableName

2. From outside of class we can delete instance variables as follows

del objectreference.variableName

…………………………………………………………….

Example:

1) class Test:

2) def \_\_init\_\_(self):

3) self.a=10

4) self.b=20

5) self.c=30

6) self.d=40

7) def m1(self):

8) del self.d

9)

10) t=Test()

11) print(t.\_\_dict\_\_)

12) t.m1()

13) print(t.\_\_dict\_\_)

14) del t.c

15) print(t.\_\_dict\_\_)

…………………………………………………………………..

Output

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

{'a': 10, 'b': 20, 'c': 30}

{'a': 10, 'b': 20}

…………………………………………………………………………

Note: The instance variables which are deleted from one object,will not be deleted from other objects.

……………………………………………………

Example:

1) class Test:

2) def \_\_init\_\_(self):

3) self.a=10

4) self.b=20

5) self.c=30

6) self.d=40

7)

8)

9) t1=Test()

10) t2=Test()

11) del t1.a

12) print(t1.\_\_dict\_\_)

13) print(t2.\_\_dict\_\_)

……………………………………………………………

Output

{'b': 20, 'c': 30, 'd': 40}

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

…………………………………………………………….

If we change the values of instance variables of one object then those changes won't be reflected to the remaining objects, because for every object we are separate copy of instance variables are available.

…………………………………………………………………………….

Example:

1) class Test:

2) def \_\_init\_\_(self):

3) self.a=10

4) self.b=20

5)

6) t1=Test()

7) t1.a=888

8) t1.b=999

9) t2=Test()

10) print('t1:',t1.a,t1.b) 11) print('t2:',t2.a,t2.b)

…………………………..

Output

t1: 888 999

t2: 10 20

…………………………………………………………..

1. Static variables:

………………………………….

If the value of a variable is not varied from object to object, such type of variables we have to declare with in the class directly but outside of methods. Such type of variables are called Static variables.

For total class only one copy of static variable will be created and shared by all objects of that class.

We can access static variables either by class name or by object reference. But recommended to use class name.

……………………………………………………………………….

Instance Variable vs Static Variable: -

…………………………………………………………………

Note: In the case of instance variables for every object a seperate copy will be created,but in the case of static variables for total class only one copy will be created and shared by every object of that class.

1) class Test:

2) x=10

3) def \_\_init\_\_(self):

4) self.y=20

5)

6) t1=Test()

7) t2=Test()

8) print('t1:',t1.x,t1.y)

9) print('t2:',t2.x,t2.y)

10) Test.x=888

11) t1.y=999

12) print('t1:',t1.x,t1.y)

13) print('t2:',t2.x,t2.y)

/...............................................

Output t1: 10 20 t2: 10 20 t1: 888 999 t2: 888 20

……………………………………………………………………………………

\*\*\*\*\*\*\*Various places to declare static variables:\*\*\*

……………………………………………………………………………

1. In general we can declare within the class directly but from out side of any method

2. Inside constructor by using class name

3. Inside instance method by using class name

4. Inside classmethod by using either class name or cls variable

5. Inside static method by using class name

……………………………………………………………………………………………

1) class Test:

2) a=10

3) def \_\_init\_\_(self):

4) Test.b=20

5) def m1(self):

6) Test.c=30

7) @classmethod

8) def m2(cls):

9) cls.d1=40

10) Test.d2=400

11) @staticmethod

12) def m3():

13) Test.e=50

14) print(Test.\_\_dict\_\_)

15) t=Test()

16) print(Test.\_\_dict\_\_)

17) t.m1()

18) print(Test.\_\_dict\_\_)

19) Test.m2()

20) print(Test.\_\_dict\_\_)

21) Test.m3()

22) print(Test.\_\_dict\_\_)

23) Test.f=60

24) print(Test.\_\_dict\_\_)

………………………………………………………………………………….

\*\*\*\*\*\*\*\*\*\*\*\*How to access static variables:\*\*\*\*\*\*\*\*\*\*\*\*

………………………………………………………………………………………….

1. inside constructor: by using either self or classname

2. inside instance method: by using either self or classname 3. inside class method: by using either cls variable or classname

4. inside static method: by using classname

5. From outside of class: by using either object reference or classnmae

………………………………………………………………………………………………

1) class Test:

2) a=10

3) def \_\_init\_\_(self):

4) print(self.a)

5) print(Test.a)

6) def m1(self):

7) print(self.a)

8) print(Test.a)

9) @classmethod

10) def m2(cls):

11) print(cls.a)

12) print(Test.a)

13) @staticmethod

14) def m3():

15) print(Test.a)

16) t=Test()

17) print(Test.a)

18) print(t.a)

19) t.m1()

20) t.m2()

21) t.m3()

……………………………………………………………………………………………

\*\*\*\*Where we can modify the value of static variable:\*\*\*

…………………………………………………………………………………………….

Anywhere either with in the class or outside of class we can modify by using classname. But inside class method, by using cls variable.

………………………………………………………………………………….

Example:

1) class Test:

2) a=777

3) @classmethod

4) def m1(cls):

5) cls.a=888

6) @staticmethod

7) def m2():

8) Test.a=999

9) print(Test.a)

10) Test.m1()

11) print(Test.a)

12) Test.m2()

13) print(Test.a)

…………………………………………………………………………………………..

Output

777

888

999

………………………………………………………………………………………………\*\*\*\*\* If we change the value of static variable by using either self or object reference variable:\*\*\*\*\*\*\*\*\*\*\*

………………………………………………………………………………………………

If we change the value of static variable by using either self or object reference variable, then the value of static variable won't be changed,just a new instance variable with that name will be added to that particular object.

……………………………………………………………………………………….

Example 1:

1) class Test:

2) a=10

3) def m1(self):

4) self.a=888

5) t1=Test()

6) t1.m1()

7) print(Test.a)

8) print(t1.a)

…………………………………………………….

Output

10

888

……………………………………………………………………….

Example:

1) class Test:

2) x=10

3) def \_\_init\_\_(self):

4) self.y=20

5)

6) t1=Test()

7) t2=Test()

8) print('t1:',t1.x,t1.y)

9) print('t2:',t2.x,t2.y)

10) t1.x=888

11) t1.y=999

12) print('t1:',t1.x,t1.y)

13) print('t2:',t2.x,t2.y)

……………………………………………………………………

Output

t1: 10 20

t2: 10 20

t1: 888 999

t2: 10 20

…………………………………………………………………………………………

Example:

1) class Test:

2) a=10

3) def \_\_init\_\_(self):

4) self.b=20

5) t1=Test()

6) t2=Test()

7) Test.a=888

8) t1.b=999

9) print(t1.a,t1.b)

10) print(t2.a,t2.b)

…………………………………………………………………………

Output

888 999

888 20

--------------------------------------------------------

1) class Test:

2) a=10

3) def \_\_init\_\_(self):

4) self.b=20

5) def m1(self):

6) self.a=888

7) self.b=999

8)

9) t1=Test()

10) t2=Test()

11) t1.m1()

12) print(t1.a,t1.b)

13) print(t2.a,t2.b)

………………………………………………………………….

Output

888 999

10 20

………………………………………………………….

Example:

1) class Test:

2) a=10

3) def \_\_init\_\_(self):

4) self.b=20

5) @classmethod

6) def m1(cls):

7) cls.a=888

8) cls.b=999

9)

10) t1=Test()

11) t2=Test()

12) t1.m1()

13) print(t1.a,t1.b)

14) print(t2.a,t2.b)

15) print(Test.a,Test.b)

…………………………………………..

Output:-

888 20

888 20

888 999

……………………………………………………………………………………….

\*\*\*\*\*\*\*\*\*How to delete static variables of a class:\*\*\*\*\*\*

…………………………………………………………………………………………

We can delete static variables from anywhere by using the following syntax

del classname.variablename

But inside classmethod we can also use cls variable

del cls.variablename

1) class Test:

2) a=10

3) @classmethod

4) def m1(cls):

5) del cls.a

6) Test.m1()

7) print(Test.\_\_dict\_\_)

………………………………………………………….

Example:

1) class Test:

2) a=10

3) def \_\_init\_\_(self):

4) Test.b=20

5) del Test.a

6) def m1(self):

7) Test.c=30

8) del Test.b

9) @classmethod

10) def m2(cls):

11) cls.d=40

12) del Test.c

13) @staticmethod

14) def m3():

15) Test.e=50

16) del Test.d

17) print(Test.\_\_dict\_\_)

18) t=Test()

19) print(Test.\_\_dict\_\_)

20) t.m1()

21) print(Test.\_\_dict\_\_)

22) Test.m2()

23) print(Test.\_\_dict\_\_)

24) Test.m3()

25) print(Test.\_\_dict\_\_)

26) Test.f=60

27) print(Test.\_\_dict\_\_)

28) del Test.e

29) print(Test.\_\_dict\_\_)

………………………………………………………………………..

\*\*\*\* Note: By using object reference variable/self we can read static variables, but we cannot modify or delete.

………………………………………………………………………….

If we are trying to modify, then a new instance variable will be added to that particular object. t1.a = 70

…………………………………………………

If we are trying to delete then we will get error.

…………………………………………………

Example:

1) class Test:

2) a=10

3)

4) t1=Test()

5) del t1.a ===>AttributeError: a

……………………………………………………………….

We can modify or delete static variables only by using classname or cls variable.

…………………………………………………………………………..

1) import sys

2) class Customer:

3) ''''' Customer class with bank operations.. '''

4) bankname='PRASANNABANK'

5) def \_\_init\_\_(self,name,balance=0.0):

6) self.name=name

7) self.balance=balance

8) def deposit(self,amt):

9) self.balance=self.balance+amt

10) print('Balance after deposit:',self.balance)

11) def withdraw(self,amt):

12) if amt>self.balance:

13) print('Insufficient Funds..cannot perform this operation')

14) sys.exit()

15) self.balance=self.balance-amt

16) print('Balance after withdraw:',self.balance)

17)

18) print('Welcome to',Customer.bankname)

19) name=input('Enter Your Name:')

20) c=Customer(name)

21) while True:

22) print('d-Deposit \nw-Withdraw \ne-exit')

23) option=input('Choose your option:')

24) if option=='d' or option=='D':

25) amt=float(input('Enter amount:'))

26) c.deposit(amt)

27) elif option=='w' or option=='W':

28) amt=float(input('Enter amount:'))

29) c.withdraw(amt)

30) elif option=='e' or option=='E':

31) print('Thanks for Banking')

32) sys.exit()

33) else:

34) print('Invalid option..Plz choose valid option')

……………………………………………

output:

D:\ prasanna \_classes>py test.py

Welcome to PRASANNABANK

Enter Your Name:Prasanna

d-Deposit

w-Withdraw

e-exit

Choose your option:d

Enter amount:10000

Balance after deposit: 10000.0

d-Deposit

w-Withdraw

e-exit

Choose your option:d

Enter amount:20000

Balance after deposit: 30000.0

d-Deposit

w-Withdraw

e-exit

Choose your option:w

Enter amount:2000

Balance after withdraw: 28000.0

d-Deposit

w-Withdraw

e-exit

Choose your option:r

Invalid option..Plz choose valid option

d-Deposit

w-Withdraw

e-exit Choose your option:e

Thanks for Banking

……………………………………………………………………………………………..\*\*\*\*\*\*\*\*\*\*\*\*\*\*Local variables:\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*:-

……………………………………………………………………………………………. Sometimes to meet temporary requirements of programmer,we can declare variables inside a method directly,such type of variables are called local variable or temporary variables.

Local variables will be created at the time of method execution and destroyed once method completes.

Local variables of a method cannot be accessed from outside of method.

…………………………………………………………………………

Example:

1) class Test:

2) def m1(self):

3) a=1000

4) print(a)

5) def m2(self):

6) b=2000

7) print(b)

8) t=Test()

9) t.m1()

10) t.m2()

……………………………………………………………..

Output

1000

2000

……………………………………………………………………………..

Example 2:

1) class Test:

2) def m1(self):

3) a=1000

4) print(a)

5) def m2(self):

6) b=2000

7) print(a) #NameError: name 'a' is not defined

8) print(b)

9) t=Test()

10) t.m1()

11) t.m2()

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

………………………………………………………………………………………………