1.# accessing the class variable using self

class student:

num123 = 40

def dis(self, name, num123):

print(name)

print(num123)

print(" ")

print(self.num123)

self.num123 = self.num123 + 2

print(self.num123)

st1 = student()

st1.dis("game", 60)

st2 = student()

st2.dis("game", 60)

# accessing the class variable using class name

class student:

num222 = 40

def dis(self, name, num222):

print(name)

print(num222)

print(" ")

print(student.num222)

student.num222 = student.num222 + 2

print(student.num222)

st1 = student()

st1.dis("game", 60)

st2 = student()

st2.dis("game", 60)

2.class User:

name = ""

def \_\_init\_\_(self, name):

self.name = name

def sayHello(self):

print("Hello, my name is " + self.name)

# create virtual objects

james = User("James")

david = User("David")

eric = User("Eric")

# call methods owned by virtual objects

james.sayHello()

david.sayHello()

# class variables

class CoffeeMachine:

name = ""

beans = 0

water = 0

def \_\_init\_\_(self, name, beans, water):

self.name = name

self.beans = beans

self.water = water

def addBean(self):

self.beans = self.beans + 1

def removeBean(self):

self.beans = self.beans - 1

def addWater(self):

self.water = self.water + 1

def removeWater(self):

self.water = self.water - 1

def printState(self):

print("Name = " + self.name)

print("Beans = " + str(self.beans))

print("Water = " + str(self.water))

pythonBean = CoffeeMachine("Python Bean", 83, 20)

pythonBean.printState()

print

""

pythonBean.addBean()

pythonBean.printState()

3. >>> class ComplexNumber:

... def \_\_init\_\_(self,r=0,i=0):

... self.real=r

... self.imag=i

... def getData(self):

... print("{0} + {1}j".format(self.real,self.imag))

...

>>> c1 = ComplexNumber(5,6)

>>> c1.getData()

>>> class ComplexNumber:

... def \_\_init\_\_(self,r=0,i=0):

... self.real=r

... self.imag=i

... def getData(self):

... print("Real:",self.real,"Imag:",self.imag)

...

>>> c1 = ComplexNumber(5,6)

>>> c1.getData()

4.class Customer:

... cust\_count=0

... def \_\_init\_\_(self,no,name):

... self.custno=no

... self.custname=name

... def setattr(self,age):

... self.age=age

... Customer.cust\_count +=1

... def display\_count(self):

... print("total Customer %d",Customer.cust\_count)

... def display\_Customer(self):

... print ("Name :",self.custname,"No:",self.custno,"Age:",self.age)

...

>>> cust1=Customer(10,"Monisha")

>>> cust1.setattr(30)

>>> cust1.display\_count()

total Customer %d 1

>>> cust1.display\_Customer()

5.getattr

========

>>> class Customer:

... cust\_count=0

... def \_\_init\_\_(self,no,name):

... self.custno=no

... self.custname=name

... def setattr(self,age):

... self.age=age

... def getattr(self):

... print("the age is ",self.age)

... Customer.cust\_count +=1

... def display\_count(self):

... print("total Customer %d",Customer.cust\_count)

... def display\_Customer(self):

... print ("Name :",self.custname,"No:",self.custno,"Age:",self.age)

...

>>> cust1=Customer(10,"Lara")

>>> cust1.setattr(30)

>>> cust1.display\_count()

total Customer %d 0

>>> cust1.getattr()

the age is 30

>>> cust1.display\_Customer()