1.write a program to explain the functionality of "issubclass()" method?

Python provides a function issubclass() that directly tells us if a class is subclass of another class.

class Base(object):

pass # Empty Class

class Derived(Base):

pass # Empty Class

# Driver Code

print(issubclass(Derived, Base))

print(issubclass(Base, Derived))

d = Derived()

b = Base()

# b is not an instance of Derived

print(isinstance(b, Derived))

# But d is an instance of Base

print(isinstance(d, Base))

2.write a program to explain inheritance?

#!/usr/bin/pyhton

class A(object):

def fun1(self,x):

self.x=x

print "Class A fun1 method"

def getxval(self):

print self.x

class B(A):

def fun(self):

print "Class B fun1 method"

obj=B()

obj.fun()

obj.fun1(100)

obj.getxval()

3.In how many ways we can access parent class members?explain with example?

#!/usr/bin/python

class parentclass(object):

def parentmember(self):

print "parent member is called"

class childclass(parentclass):

def childmember(self):

print "child member is called"

parent.parentmember(self)//Accessing parent member using base class

def callparentmembers(self):

self.parentmember() //Accessing parent member

def calparentusingsuper(self):

super(childclass,self).parentmember()//Accessing parent member using super classes

obj=childclass()

obj.childmember()

obj.callparentmembers()

obj.parentmember()// Accessing parent member from object reference

4.what is "Object" in below code?

ex :class(Object):

pass

Like Java Object class, in Python (from version 3.x), object is root of all classes.

In Python 3.x, “class Test(object)” and “class Test” are same.  
In Python 2.x, “class Test(object)” creates a class with object as parent (called new style class) and “class Test” creates old style class (without object parent).

This means that class statements that don't have any base classes are always classic classes in Python 2.2.

5.what is multiple inheritance?Does python support it?if yes, explain the procudure with example?

Unlike Java and like C++, Python supports multiple inheritance. We specify all parent classes as comma separated list in bracket.

class Base1(object):

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

self.str1 = "Meena"

print "Base1"

class Base2(object):

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

self.str2 = "paleti"

print "Base2"

class Derived(Base1, Base2):

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

# Calling constructors of Base1

# and Base2 classes

Base1.\_\_init\_\_(self)

Base2.\_\_init\_\_(self)

print "Derived"

def printStrs(self):

print(self.str1, self.str2)

ob = Derived()

ob.printStrs()

6.write a program to print parent class name from child instance?

class Parent(object):

def type(self):

print Parent.\_\_name\_\_

class Child(Parent):

def getclsname(self):

super(Child,self).type()

p = Child()

p.getclsname()