UNIT 4 IMP QUESTIONS

(1-4) 2Marks (5-8) 5 Marks and (9-10) 10Marks

1. How class and objects are declared in python

2. Describe Inheritance in Python

3. How we can access the attributes of a class

In [ ]:
```
class rectangle:
    length=0
    breadth=0
r1=rectangle()
print(r1.length)
print(r1.breadth)
```

4. List the different method overloading functions using python

5. What is data Hiding explain with the example

In [ ]:
```
class hide:
    x=5
    def dis(self,x):
        x=10
        print('value of local variable', x)
        print('value of instance variable', x)
ob1=hide()
ob1.dis(50)

#instance variable x=5 initialised
#local variable x=10
#value of instance variable is hidden by local variable
#if programmer does not want to hide the value
#need to use self with name of instance variable
```

6. WAP to implement the multilevel inheritance

In [ ]:
```
class a:
    name=''
    age=0
class b(a):
    height=''
class c(b):
    weight=''
    def read(self):
        print('enter the following values')
        self.name=input('enter name:')
        self.age=eval(input('enter age'))
        self.height=eval(input('enter height'))
        self.weight=eval(input('enter weight'))
    def display(self):
        print('enter values are as follows')
        print('name', self.name)
        print('age', self.age)
        print('height', self.height)
        print('weight', self.weight)
c1=c()
c1.read()
c1.display()
```

7. WAP to implement the concept of function overloading

8. WAP to add and multiply two object using operator overlading

In [ ]:
```
class over:
    def __init__(self,x):
        self.x=x
    def __add__(self,other):
        print('the value of ob1 ',self.x)
        print('the value of ob2 ',other.x)
        print('the addition of two object is: ',end='')
        return (self.x+other.x)
    def __sub__(self,other):
        return (self.x-other.x)
ob1=over(20)
ob2=over(30)
ob3=ob1+ob2
print(ob3)
ob4=ob1-ob2
print(ob4)
```

9. WAP to implement the concept of multiple inhertance

In [ ]:
```
#multiple inheritance

#a-> b <-c  a nd b are base class

class a:
    a1=0
class b:
    b1=0
class c(a,b):
    c1=0
    def read(self):
        print('enter the following values')
        self.a1=eval(input('enter value of a1'))
        self.b1=eval(input('enter value of b1'))
        self.c1=eval(input('enter value of c1'))
    def display(self):
        print('enter values are as follows')
        print('a', self.a1)
        print('b', self.b1)
        print('c', self.c1)

ob1=c()         #instance of child class
ob1.read()
ob1.display()
```

10. WHat is method overriding problem, how inheritance face this problem and how to resolve it

In [ ]:
```
class a(object):
    def dis(self):
        print('i m in a')

class b(a):
    def dis(self):
        print('i m in b')
        a.dis(self)     #call parent class method

class c(a):
    def dis(self):
        print('i m in c')
        a.dis(self)

class d(b,c):
    def dis(self):
        print('i m in d')
        b.dis(self)
        c.dis(self)

ob1=d()
ob1.dis()
```