

**""" Q17 Yourclass question on page 63"""**

class Yourclass:

marks=10 #class variable

name="ABC"

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

self.marks=marks #instance attribute

self.name=name

def display(self):

print marks

print name

obj=Yourclass(10,"ABC") #THIS IS THE MISSING STATEMENT.

#or

obj1=Yourclass

**""" Q18 Yourclass question on page 63"""**

class Yourclass:

marks=10 #class variable

name="ABC"

def \_\_init\_\_(Self,marks,name):

self.marks=marks #instance attribute

self.name=name

def display(self):

print marks

print name

YC=Yourclass(5,"name") #any argument

"""

If command is YC.display()

the output will be:

10

"ABC"

Which means that it will give output as class variables.

If 'print self.marks' and 'print self.name' is in the display(self),

then the output will be:

5

"name"

"""

### ''' Q19 Predict the output'''

```
class Match:
    """Runs and Wickets"""
    runs=281
    wickets=5

    def __init__(self,runs,wickets):
        self.runs=runs
        self.wickets=wickets

    print "Runs scored are :",runs
    print "Wickets taken are :",wickets

print "Test.__do__ :",Match.__doc__
print "Test.__name__ :",Match.__name__
print "Test.__module__ :",Match.__module__
print "Test.__bases__ :",Match.__bases__
print "Test.__dict__ :",Match.__dict__

'''
```

SOLUTIONS : This is the output -

```
Runs scored are : 281
Wickets taken are : 5
Test.__do__ : Runs and Wickets
Test.__name__ : Match
Test.__module__ : __main__
Test.__bases__ : ()
Test.__dict__ : {'__module__': '__main__', 'runs': 281, '__doc__': 'Runs and
Wickets', '__init__': <function __init__ at 0x0398BA70>, 'wickets': 5}

'''
```

"Q20 Create the class SOCIETY with the following information..."

```
class SOCIETY:
    def __init__(self):
        self.soc_name=""
        self.house_no=0.0
        self.members=0.0
        self.flat=""
        self.income=0.0

    def allocate_flat(self):
        if self.income>=25000:
            self.flat='A'
        elif self.income<25000 and self.income>=20000:
            self.flat='B'
        elif self.income<15000:
            self.flat='C'

    def Inputdata(self):
        self.soc_name=raw_input("Enter society name :")
        self.house_no=input("Enter house number :")
        self.members=input("Enter no. of members : ")
        self.income=input("Enter income :")

    def Showdata(self):
        print "Society Name is : ",self.soc_name
        print "House no. us : ",self.house_no
        print "Number of members are : ",self.members
        print "Flat type is : ",self.flat
        print "Income is : ",self.income

obj=SOCIETY()
obj.Inputdata()
obj.allocate_flat()
obj.Showdata()
```

"Q21 Page 65 - Define a class ITEMINFO with following description...."

```
class ITEMINFO:
    def __init__(self):
        self.icode=0.0
        self.item_name=""
        self.price=0.0
        self.qty=0.0
        self.discount=0.0
        self.netprice=0.0

    def FindDisc(self):
        if self.qty>=20:
            self.discount=20
        elif self.qty<20 and self.qty>10:
            self.discount=15
        elif self.qty<=10:
            self.discount=0

    def Buy(self):
        self.icode=input("Enter Item Code: ")
        self.item_name=raw_input("Enter name of that Item: ")
        self.price=input("Enter price of that item : ")
        self.qty=input("Enter quantity: ")

    def ShowAll(self):
        print "Item Code is : ",self.icode
        print "Item name is : ",self.item_name
        print "Price of that item is : ",self.price
        print "Quantity of that item is :",self.qty
        print "Net price is : (price of 1 x quantity) : ",self.price*self.qty
        print "Discount given is : ",self.discount
        print "Final amount to be paid is : ",(self.price*self.qty)-self.discount

obj=ITEMINFO()
obj.Buy()
obj.FindDisc()
obj.ShowAll()
```