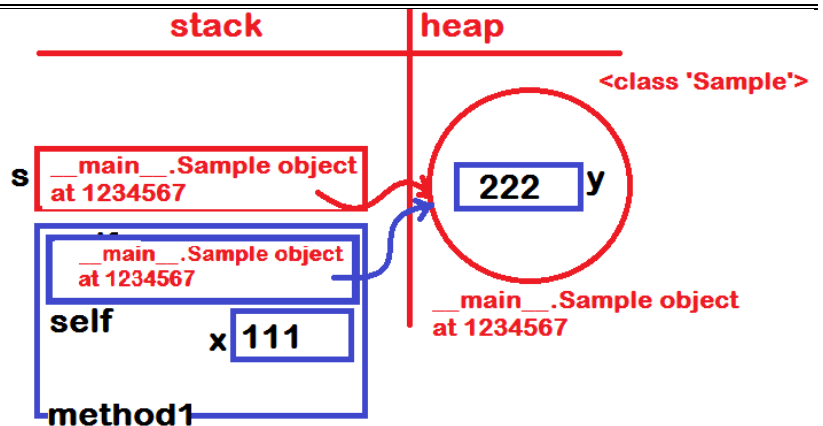# Instance Fields

- The variable which declared in the class by using "self"
- Instance fields can be accessed through out class or outside of the class
- For Every instance field memory is allocated with in the object
- Every instance field must be referred by using "self" with in the class
- Every instance field must be referred by an object reference whenever you want access it from **outside** of the class
- **For Every instance field an individual memory block will be created in every object of the same class**, thus changing the value of an instance field in one object will not reflected to the same instance field of another object of the same class

```
class Sample:
    def method1(self):
        x=111  #local variable
        self.y=222 #instance field
        print("x val is :",x)
        print("y val is : ",self.y)

#calling
s=Sample( )
s.method1( )
print("From outside")
print("y val is : ",s.y)
```

stack | heap

<class 'Sample'>

s | __main__.Sample object at 1234567

222 y

__main__.Sample object at 1234567

self

x 111

__main__.Sample object at 1234567

method1

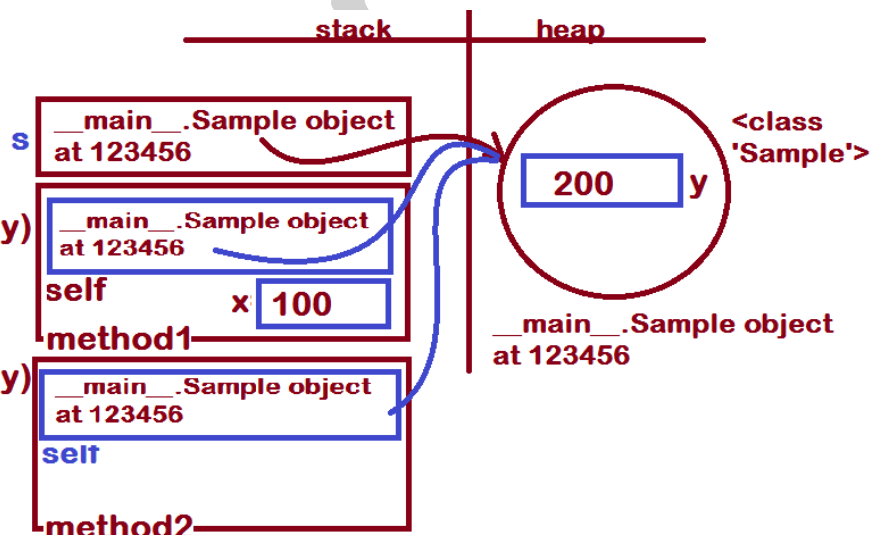## Example 2:

```
class Sample:
    def method1(self):
        x=100
        self.y=200;
        print("x val is : ",x)
        print("y val is : ",self.y)

    def method2(self):
        print("x val is : ",x)
        print("y val is : ",self.y)

#calling
s=Sample()
s.method1()
s.method2()
```

stack | heap

<class 'Sample'>

s | __main__.Sample object at 123456

200 y

__main__.Sample object at 123456

self

x 100

method1

__main__.Sample object at 123456

self

method2

Example 3:

```python
class Sample:
    def method1(self):
        self.x=111
#call
s1=Sample( )
s1.method1( )

s2=Sample()
s2.method1()
print("s1 x  is : ",s1.x) #111
print("s2 x is : ",s2.x) #111
s1.x=999
print("s1 x is : ",s1.x) #999
print("s2 x is : ",s2.x) #111
```
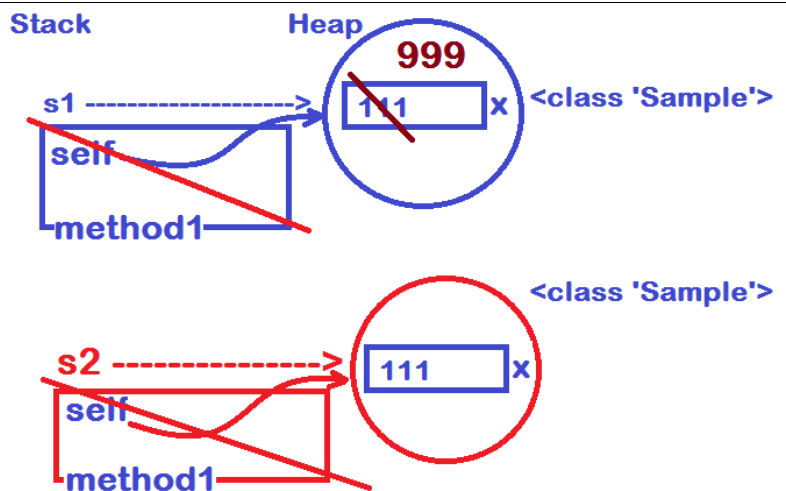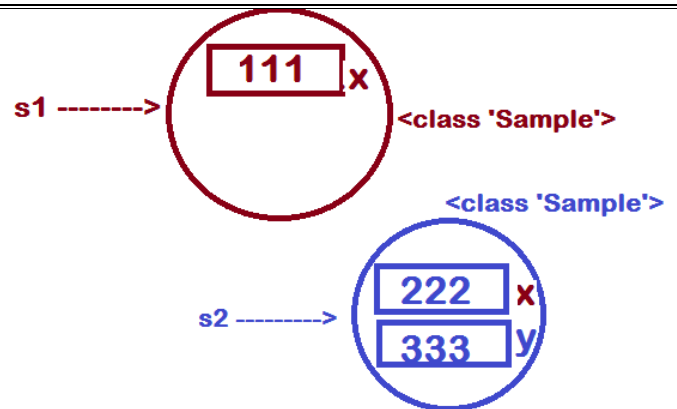

Note : Based on your application requirements we can also add or remove an instance field from the specified object from outside of the class

```
class Sample:
    pass

#calling
s1=Sample()
s1.x=111
print("s1 x is : ",s1.x) #111

s2=Sample( )
print("s2 x is : ",s2.x) #ArributeError
s2.x=222
s2.y=333
print("y val is : ",s2.y) #y val is  : 333
print("x val is : ",s2.x) #x val is : 222
```


Adding and deleting and instance field from an object from outside of the class

```
class Sample:
        pass

#calling
s1=Sample()
#Syn for adding an instance field to an object from outside of the class
#object_ref.fieldname=value
s1.x=100
print("s1 x is : ",s1.x)

#Syn for deleting an instance field from an object from outside of class
#del object_ref.fieldname
del s1.x
print("s1 x is  :",s1.x) #AttributeError
```