

## HAS-A Relationship [Composition]

- It is the process of creating an object of the super class into the subclass. By the reference of “superclass” we can access the attributes of the “superclass” into “subclass”

**Syn:**

**<class> <SuperClassName>:**

**Attributes**

**Fields [static variable | instance fields]**

**Methods [ static methods | class methods | instance methods]**

**<class> <SubClassName>:**

**Attributes**

**Fields**

**Methods**

**<sup\_ref>=<SuperClassname>([list of arguments])**

### Example:

```
class SuperClass:
```

```
    x=111 #static variable
```

```
class SubClass:
```

```
    def method1(self): #instance mtd
```

```
        s=SuperClass()
```

```
        print("SuperClass x : ",s.x)
```

```
#calling
```

```
s=SubClass()
```

```
s.method1()
```

## Example 2:

```
class SuperClass:
    def method1(self):
        print("Super Class Ins Mtd-1")

    @classmethod
    def method2(cls):
        print("Super class class Mtd-2")

    @staticmethod
    def method3():
        print("super static Mtd-3")

class SubClass:
    def method1(self):
        sup=SuperClass()
        sup.method1()
        sup.method2()
        sup.method3()

#calling
sub=SubClass()
sub.method1()
```