```
#Python magic Method for operator overloading:

#Operator Magic Method:

#1_(+):
class Addition:

    def __init__(self,a):
        self.a = a

    def __add__(self,other):
        return self.a + other.a

a1= Addition(100)
a2= Addition(1000)

print(a1+a2)

#2_(-):
class Subtraction:

    def __init__(self,a):
        self.a = a

    def __sub__(self,other):
        return self.a - other.a

s1= Subtraction(1000)
s2= Subtraction(100)

print(s1-s2)

#3_(*):
class Multiply:

    def __init__(self,a):
        self.a = a

    def __mul__(self,other):
        return self.a * other.a

m1= Multiply(10)
m2= Multiply(9)

print(m1*m2)

#4_(/):
class Division:

    def __init__(self,a):
        self.a = a

    def __truediv__(self,other):
        return self.a / other.a
```

```
d1= Division(500)
d2= Division(50)

print(d1/d2)

#5_(//):
class FloorDivision:

    def __init__(self,a):
        self.a = a

    def __floordiv__(self,other):
        return self.a // other.a

f1= FloorDivision(420)
f2= FloorDivision(40)

print(f1//f2)

#6_(%):
class Mod:

    def __init__(self,a):
        self.a = a

    def __mod__(self,other):
        return self.a % other.a

mo1= Mod(100)
mo2= Mod(2)

print(mo1%mo2)

#7_(**):
class Exponential:

    def __init__(self,a):
        self.a = a

    def __pow__(self,other):
        return self.a ** other.a

e1= Exponential(2)
e2= Exponential(4)

print(e1**e2)

#8_(>>):
#def __rshift__(self,other):

#9_(<<):
#def __lshift__(self,other):
```

```
#10_(&):
#def __and__(self,other):

#11_(|):
#def __or__(self,other):

#12_(^):
#def __xor__(self,other):


#Comparision_Operators:

#1_(<):
class LessThan:

    def __init__(self,a):
        self.a = a

    def __lt__(self,other):
        return self.a < other.a

lt1= LessThan(90)
lt2= LessThan(100)

print(lt1<lt2)

#2_(>):
class GreaterThan:

    def __init__(self,a):
        self.a = a

    def __gt__(self,other):
        return self.a > other.a

gt1= GreaterThan(100)
gt2= GreaterThan(90)

print(gt1 > gt2)

#3_(<=):
class LessThanEqual:

    def __init__(self,a):
        self.a = a

    def __le__(self,other):
        return self.a <= other.a

le1= LessThanEqual(10)
le2= LessThanEqual(16)

print(le1 <= le2)
```

```
#4_(>=):
class GreaterThanEqual:

    def __init__(self,a):
        self.a = a

    def __ge__(self,other):
        return self.a >= other.a

ge1 = GreaterThanEqual(15)
ge2 = GreaterThanEqual(10)

#5_(==):
class EqualEqual:

    def __init__(self,a):
        self.a = a

    def __eq__(self,other):
        return self.a == other.a

ee1= EqualEqual(100)
ee2= EqualEqual(100)

print(ee1==ee2)

#6_(!=):
class NotEqual:

    def __init__(self,a):
        self.a = a

    def __ne__(self,other):
        return self.a != other.a

ne1= NotEqual(50)
ne2= NotEqual(100)

print(ne1!=ne2)
```