Operator Overloading

- Operator Overloading is where operators behave differently in different situations
- Operator Overloading is possible in normal DataType also
- Using operators we can achieve polymorphism
- · Lets see this with an example
- We can add 2 rational numbers using operator overloading

```
class Rational:
    def __init__(self, p=1, q=1):
        self.p=p
        self.q=q

    def __add__(self, other):
        s = Rational()
        s.p = self.p * other.q + self.q * other.p
        s.q = self.q * other.q
        return s

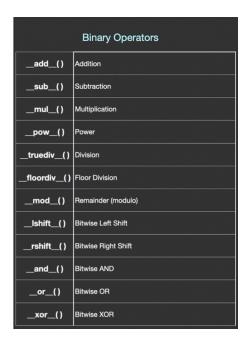
r1 = Rational(2,3)
r2 = Rational(2,5)

sum = r1+r2
print(_sum.p, sum.q)
```

we are writing a constructor and method through this way we add 2 rational numbers

```
16 15
Process finished with exit code 0
```

• You can also overload for subtraction, multiplication, power, division etc.



 Operator Overloading also works well for comparison operators and assignment operators.



