```
D:\durga_classes>py test.py
Traceback (most recent call last):
  File "test.py", line 7, in <module>
    print(b1+b2)
TypeError: unsupported operand type(s) for +: 'Book' and 'Book'
```

We can overload + operator to work with Book objects also. i.e Python supports Operator Overloading.

For every operator Magic Methods are available. To overload any operator we have to override that Method in our class.
Internally + operator is implemented by using __add__() method.This method is called magic method for + operator. We have to override this method in our class.

Demo program to overload + operator for our Book class objects:

```
1)  class Book:
2)      def __init__(self,pages):
3)          self.pages=pages
4)
5)      def __add__(self,other):
6)          return self.pages+other.pages
7)
8)  b1=Book(100)
9)  b2=Book(200)
10) print('The Total Number of Pages:',b1+b2)
```

Output: The Total Number of Pages: 300

The following is the list of operators and corresponding magic methods.

```
+   ---> object.__add__(self,other)
-   ---> object.__sub__(self,other)
*   ---> object.__mul__(self,other)
/   ---> object.__div__(self,other)
//   ---> object.__floordiv__(self,other)
%   ---> object.__mod__(self,other)
**   ---> object.__pow__(self,other)
+=   ---> object.__iadd__(self,other)
-=   ---> object.__isub__(self,other)
*=   ---> object.__imul__(self,other)
/=   ---> object.__idiv__(self,other)
//=   ---> object.__ifloordiv__(self,other)
%=   ---> object.__imod__(self,other)
**=   ---> object.__ipow__(self,other)
<   ---> object.__lt__(self,other)
<=   ---> object.__le__(self,other)
>   ---> object.__gt__(self,other)
>=   ---> object.__ge__(self,other)
```