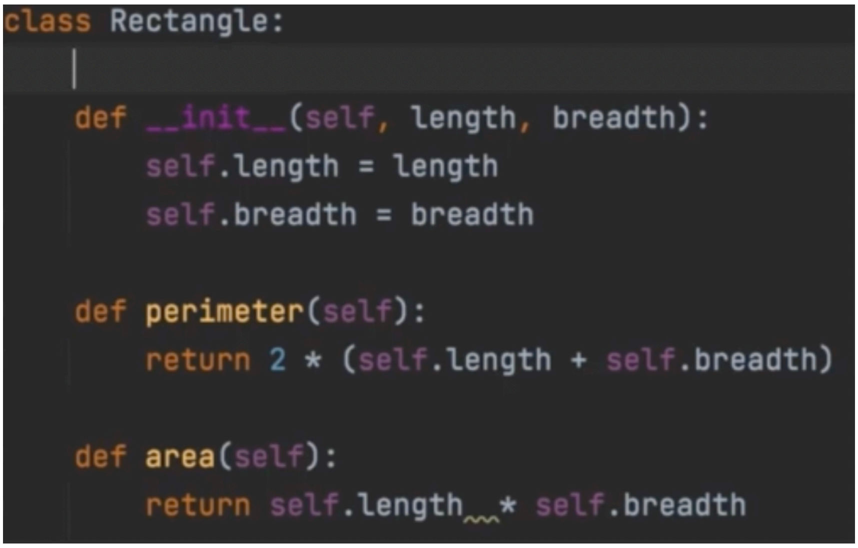
Class and Static Variable



```
class Rectangle:

    def __init__(self, length, breadth):
        self.length = length
        self.breadth = breadth

    def perimeter(self):
        return 2 * (self.length + self.breadth)

    def area(self):
        return self.length * self.breadth
```

When we declare a variable inside a class but outside any method, it is called as class or static variable in python.

- self.length and breath are instance variable.

  def perimeter(self)

- Class variables can be accessed from within the class as well as from outside the class.

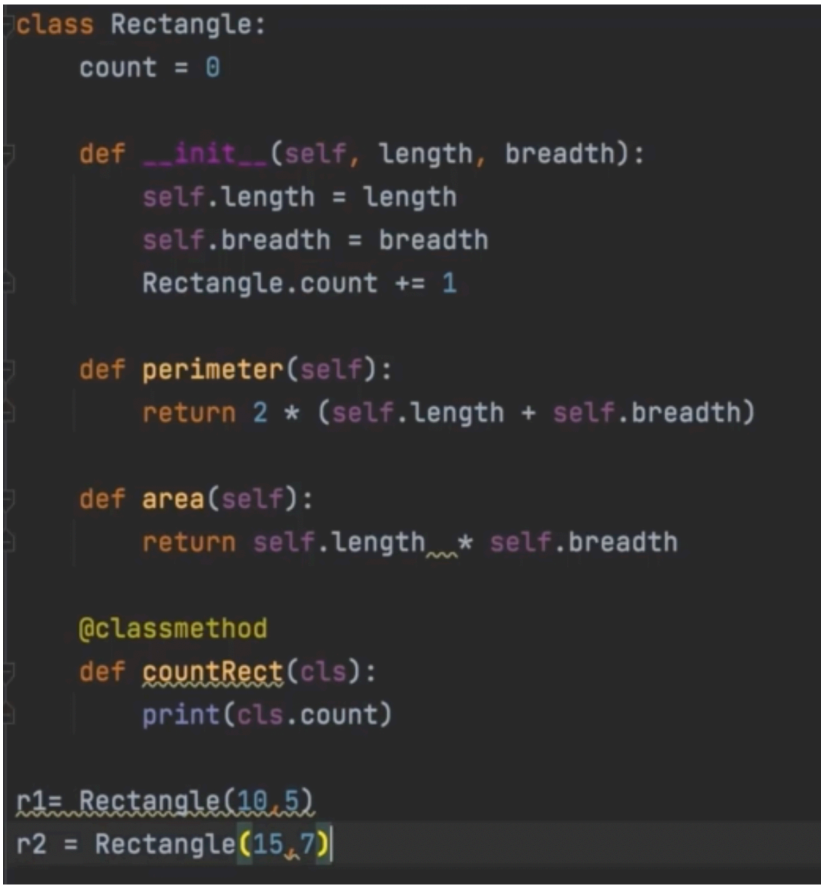
class and static method can access class variables. Each method has its own way of accessing static variables.

There are two ways of accessing class variables from inside class method-

- cls.ClassVariable
- ClassName.ClassVaraiable

There is only one way to access class variables from static method-

- ClassName.ClassVariable



```
class Rectangle:
    count = 0

    def __init__(self, length, breadth):
        self.length = length
        self.breadth = breadth
        Rectangle.count += 1

    def perimeter(self):
        return 2 * (self.length + self.breadth)

    def area(self):
        return self.length * self.breadth

    @classmethod
    def countRect(cls):
        print(cls.count)

r1= Rectangle(10,5)
r2 = Rectangle(15,7)
```

- Here count= 0
- It is known as class or static cause its not inside the method
- The counter belongs to rectangle r1 and r2 that means count variable is common to both objects (r1 and r2)
- Count will belong to class (Rectangle) so it is also static(fixed, single copy to both r1,r2)
- @classmethod (decorator)