

Self and Construct

Self

- Self is the **reference** to the current value
- It is the first argument of the constructor
- If self is not given PVM will take the 1st parameter value as self by default
- You can write different names for self but self is recommended
- When you create 2 object the self will refer to the current object that is being called i.e, the self value will be same for these 2 object but it will show values of the method being called (i.e, 1st or 2nd)
- You can verify this by checking the ID of self in both method call

```
class Cuboid:
    def __init__(self, l, b, h):
        print(id(self))
        self.length = l
        self.breadth = b
        self.height = h

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

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

    def volume(self):
        return self.length * self.breadth * self.height

c1 = Cuboid(10, 5, 3)
print(id(c1))
```

```
/Users/PycharmProject
140228827119568
140228827119568
```

Constructor

- In a method if `__init__` is used then that method becomes constructor method as `__init__` is a constructor.
- When you create an object of this class then automatically constructor method is called and initialised
- If a class having attributes then its mandatory to have `__init__` method. Therefore every class must have `__init__`
- When `__init__` is not given in a class then PVM will provide default constructor which will not have any property
- You can write more than 1 constructor but only 1 will execute but its not recommended to do so
- You can give default argument in parameter and when you call the method you can either give values or it will consider the default values i.e you can make constructor using different Parameters
- Constructors are not overloaded

```
class Cuboid:
    def __init__(self, l, b, h):
        print(id(self))
        self.length = l
        self.breadth = b
        self.height = h

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

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

    def volume(self):
        print(id(self))
        return self.length * self.breadth * self.height

c1=Cuboid(10,5,3)
print(id(c1))
c1.volume()
```

```
140496912371664
140496912371664
140496912371664
```

Program explanation(1 and 2):

- A class of cuboid is created , in method we are writing a constructor it takes 1st parameter as self, and we are initialisation it by writing its properties I.e length, breath and height , we are printing the ID of self as well to show that only one self is used for multiple methods call.
- Then we are defining a method for lidarea , total area and volume and initialisation it with self .
- When printing the results for lidarea , total area and volume we can see that the ID of self is constant while ans for theses 3 methods are different