```
# instance variable and instance method
class Product:

    # instance method
    def getDetails(self, productId, productName, productPrice):
        self.pId = productId # Instance variable
        self.pName = productName
        self.pPrice = productPrice
        print(self.pId, self.pName, self.pPrice,"rs") # 101 Samsung 10000 rs

p = Product()
p.getDetails(101, "Samsung", 10000)
```

```
# constructor with arguments
class Product:

    # constructor with arguments
    def __init__(self, productId, productName):
        self.pId = productId # Instance variable
        self.pName = productName # Instance variable
        print(self.pId, self.pName)

Product(101, "Samsung") # Instance creation
Product(102, "LG") # Instance creation
Product(103, "Vivo") # Instance creation

101 Samsung
102 LG
103 Vivo
```

```python
# constructor with arguments, instance method
class Product:

    # constructor with arguments
    def __init__(self, productId, productName):
        self.pId = productId
        self.pName = productName

    # instance method
    def getDetails(self):
        print(self.pId, self.pName) # 101 Samsung

p = Product(101, "Samsung")
p.getDetails() # calling instance method

'''Note: we cannot access instance variables directly, need object reference'''
```

We can modify the instance variables using object reference
When we modify the instance variables of single object, it will not reflect another objects.
In python every object maintains a separate copy of instance variable

```
 # Modify Instance Variable
class Product:
    def productDetails(self, productId, productName):
        self.pId = productId
        self.pName = productName
        print(self.pId , self.pName)

p = Product()
p.productDetails(101, "Samsung")

# Modify Value
p.pId = 102
p.pName = "Lg"
print(p.pId, p.pName)
```

```
101 Samsung
102 Lg
```

We can access instance variables in two ways, using object ref and getattr() method

```
# getattr() method
class Product:

    def productDetails(self, productId, productName):
        self.pId = productId
        self.pName = productName

p = Product()
p.productDetails(101, "Samsung")

print('Product Id: ', getattr(p, 'pId'))
print('Product Name: ', getattr(p, 'pName'))
```

```
Product Id:  101
Product Name:  Samsung
```

```
# Adding dynamic value to instance variable
class Product:

    def productDetails(self, productId, productName):
        self.pId = productId
        self.pName = productName
        print(self.pId, self.pName)

p = Product()
p.productDetails(101, "Samsung")

# add new instance variable
p.productCost = 10000.00
print(p.pId, p.pName, p.productCost)

101 Samsung
101 Samsung 10000.0
```

Syntax: delattr(object, name)
object: the object whose attribute we want to delete.
name: the name of the instance variable we want to delete from the object.

```
# delattr() function
class Product:

    def productDetails(self, productId, productName):
        self.pId = productId
        self.pName = productName
        print(self.pId, self.pName)

p = Product()
p.productDetails(101, "Samsung")

# delete instance variable using delattr()
delattr(p.pId, 'productId') # AttributeError: 'int' object has no attribute 'productId'
```