Instance Variables or Object Level Variables
The variables within class bind with “self” is called instance variable or object level variable.
This variable defines property of an object.
Instance variables are created or accessed within instance method or object level method.

Object properties are defined in different ways

1. Using instance method
2. Using constructor
3. Using class dictionary
4. Creating dynamically
5. Using property class

Example:

```
class Employee: # Data type name
    def create_properties(self):
        self.empno=None # I.V
        self.ename=None # I.V
        x=100 # local variable



emp1=Employee()
emp1.create_properties()
emp2=Employee()
emp2.create_properties()
print(emp1.empno,emp1.ename)
print(emp2.empno,emp2.ename)
comp1=complex()
print(comp1.real,comp1.imag)
```

Output:

```
None None
None None
0.0 0.0
```

Constructor
Constructor is a special method defined inside class.

Constructor is instance method or object level method.
Constructor is a magic method and this method is executed automatically when ever object of class is created.
It is a predefined method, the name of this method is **__init__,** any method which is prefix and suffix with __ is called magic method.
Constructor is used for creating instance variables or initialization of object.
Within class constructor is defined,

1. Without parameters
2. With parameters

Constructor without parameters does not receive values.
Constructor with parameters receives values.

Example of constructor without parameters

```
class Alpha:
   def __init__(self):
       print("object created")



a1=Alpha()
a2=Alpha()
```

Output:

```
object created
object created
```

Example of constructor without parameters or arguments:

```
class Student:
    def __init__(self):
        self.rollno=None
        self.name=None



stud1=Student()
stud2=Student()
comp1=complex()
```

```
print(stud1.rollno,stud1.name)
print(stud2.rollno,stud2.name)
print(comp1.real,comp1.imag)
```

Output

```
None None
None None
1.0    0.0
```

Example of constructor with parameters:

```
class Student:
    def __init__(self,r,n):
        self.rollno=r
        self.name=n



stud1=Student(101,"Naresh")
stud2=Student(102,"Suresh")
print(stud1.rollno,stud1.name)
print(stud2.rollno,stud2.name)
```

Output:

```
101 Naresh
102 Suresh
```

Constructor is executed or invoked only one time on object.

Example:

```
class Time:
    def __init__(self):
        self.hh=0
        self.mm=0
        self.ss=0
```

```
t1=Time()
t2=Time()
print(t1.hh,t1.mm,t1.ss)
print(t2.hh,t2.mm,t2.ss)
```

Output:

```
0 0 0
0 0 0
```

Example of constructor with parameters

```
class Time:
    def __init__(self,h,m,s):
        self.hh=h
        self.mm=m
        self.ss=s



t1=Time(10,5,20)
t2=Time(12,40,10)
print(t1.hh,t1.mm,t1.ss)
print(t2.hh,t2.mm,t2.ss)
```

Output:

```
10 5 20
12 40 10
```

Block of code which has to be executed on creation object is defined inside constructor method.

Example of constructor and instance method

```
class Calculator:
    def __init__(self,n1,n2):
        self.num1=n1
        self.num2=n2
    def add(self):
```

```
        print(f'sum of {self.num1} and {self.num2} is
{self.num1+self.num2}')
    def sub(self):
        print(f'diff of {self.num1} and {self.num2} is {self.num1-
self.num2}')


calc1=Calculator(10,5)
calc1.add()
calc1.sub()
calc2=Calculator(5,2)
calc2.add()
calc2.sub()
```

Output:

```
sum of 10 and 5 is 15
diff of 10 and 5 is 5
sum of 5 and 2 is 7
diff of 5 and 2 is 3
```