In oops we have two different types of variables:

Diagram

Description automatically generated

Instance variable:

Let’s create a class car and create the init method. These are called instance variables because as the car/object changes the variables. Ex: by default the values are 10 and BMW, however we can change them

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

* Now what if I want to create a variable which is common for all the objects that’s where the class variables come into the picture.

Class variable also known as static variable:

Define the wheels outside the init method and it is common for all the objects, whereas the variables which are inside init are associated with the specific object and these can be changed at a specific object ex: we have changed the c1 mileage to 8 as seen before.

How to access the class variable wheels is c1.wheels as highlighted below.

Graphical user interface, application, website

Description automatically generated

Graphical user interface, application, Teams

Description automatically generated

Now we want to update the value of wheels from 4 to 5.



We have 2 namespaces:

1)class namespace: Where you will store all the class variables

2)object/instance namespace: where you will create all the instance variables

Word

Description automatically generated

Since wheels is a class namespace, to modify it we need to using a class as shown it below. Now the moment you change the value to 5, since it is shared it will be updated for both the objects, check it by running the code.

Graphical user interface, application

Description automatically generated

Conclusion: there are two types of variables, instance variables and class variables.

**Types of methods:**

**Text

Description automatically generated**

Unlike variables, here class and static are different in methods. So in total we have 3 different types of methods as shown above.

1. Instance method:

First create a student class which consists of 3 subjects marks(m1,m2,m3) which we want to take it from user 🡪 then create student objects s1,s2 🡪 create a class variable school name🡪 now we want to perform average of the marks, let’s create a method avg inside the student class.

Graphical user interface

Description automatically generated

Here the average is a instance method as we can see that we had passed self. Let’s check the average for 2 objects s1 and s2.

Avg is a instance method: there are 2 types of instance methods 1) accessor methods 2)Mutator methods

Get(accessor) which fetches or gets the values and set(mutator) which sets the value

Graphical user interface, text, application

Description automatically generated

**2) Class method:**

Why do we use class method: m1, m2 and m3 are instance variables which can be used with instance method.

Look at school variable: to work with this class variable(in our case it is school = ‘python’) we need to work with a **class method.** Recollect that if it’s value changes for one student it will change/affect all the students.

If we want to work with instance we will use the self keyword, whereas when we want to work with class we will mention **cls** as shown below. + when we need to create a class method, we have to use decorator ex: @classmethod as shown below. Call the method and check the output

Graphical user interface, text, application

Description automatically generated

So Instance methods work with instance variables, class variables work with class variables let’s look at static method.

1. **Static method:** are used when we are not dealing with instance or class variables.

Graphical user interface, text, application

Description automatically generated

class Student:

    school = 'python'

    def \_\_init\_\_(self, m1,m2,m3):

        self.m1 = m1

        self.m2 = m2

        self.m3 = m3

    def average(self):

        return (self.m1+self.m2+self.m3)/3

    @classmethod

    def getschool(cls):

        return cls.school

    @staticmethod

    def info():

        print('This is from student class, static method')

rishi = Student(45,55,100)

bunny = Student(78,88,98)

print(bunny.average())

print(Student.getschool())

Student.info()