**Class Fields**

We have now discussed both instance methods and static methods as part of the class definition. In addition, we have already seen the use of instance fields, which store information of each individual instance (or object).

For example,

```python
class Fraction:
    """ The Fraction class creates an object with a numerator & denominator."""

    def __init__(self, n, d):
        """This method creates the num and den fields."""
        self.num = n
        self.den = d
```

has two instance fields, num and den. Each *Fraction* object we create will have its own unique numerator and denominator value, stored in its own instance of these fields.

Contrast this with the class field. The primary difference between an instance field and class field is that there is only a single copy of the class field, regardless of how many objects are created using that class.

Class fields can be useful for storing quantities that are common and shared by the members of the class.

Example 1 – To create a class *Car*, we might want various fields to track the speed (speedometer), distance travelled (odometer), and amount of fuel in the tank (fuel gauge). These fields will vary from car to car, so they should be instance fields. On the other hand, if we wish to track the current price of gas, it will be the same for every car, so a class field might be more appropriate.

```python
class Car:
    """ The Car class creates an object with a specific speed, distance travelled, and fuel tank reading."""

    #Constant Class Fields
    FUEL_CAPACITY = 50

    #Variable Class Fields
    gas_price = 1.22  # $/litre

    def __init__(self, speed, distance, fuel):
        """This method creates the speed, distance, and fuel tank instance fields."""
        self.speed = speed
        self.distance = distance
        self.fuel = fuel
```
Assuming these cars are physically identical, they would have characteristics shared between all cars (class field), but unlike the price of gasoline, the values would not change. For example, the fuel tank capacity could be represented as a constant class field.

Class fields can be referenced in a manner similar to static methods:
<class identifier>.<field identifier>

For example,
Car.FUEL_CAPACITY

Questions and Exercises 1.8
1. What is the difference between the declaration of a class field and that of an instance field?
2. State the convention that is used to distinguish constant class field identifiers from variable class field identifiers.
3. From outside a class, how would we refer to (a) a class field? (b) an instance field?
4. (a) Write an accessor method called get_price for the gas_price field shown in Example 1. 
(b) Write a mutator method called set_price to assign a new value to the gas_price field.
(c) Write a statement that could be used to set a variable current_price to the value of the gas_price field (from outside the class).
(d) Write a statement that could be used to set the gas_price field to the value of 1.45 (from outside the class).
5. Suppose that a class Account is to be created to maintain records of savings accounts in a bank. State, with reasons, which of the following should be class fields and which should be instance fields.
   (a) minimum balance (b) account number (c) current balance (d) interest rate