



CLASS

Terminology continued



BASIC TERMINOLOGY

- »» Class
- »» Object
- »» Attribute
- »» Method

CLASS AND ITS OBJECTS

- »» **Class** is a "plan" for creating objects in Python.
- »» It defines the **structure and behavior** of the objects that will be instantiated from it.
- »» **Object** is an instance of a class.
- »» They have their own unique state (values of **attributes**) and behavior (**methods** they can execute).



ATTRIBUTES AND METHODS

- »» **Attributes** are *variables* defined within the class, representing object properties.
 - »» Examples: age, year, size, weight, color, location
- »» **Methods** are *functions* defined within the class, representing object behavior.
 - »» Examples: get_age(), start_engine(), calculate_distance(), close(), run()

DOT NOTATION

- »» In Python, **dot notation** is a way to access attributes and methods of an object.
- »» It involves using a dot (.) followed by the attribute or method *name* to access or modify the attributes and call methods associated with a particular object.

- »» Accessing attributes:

```
object.attribute  
car.year, car.model, car.max_speed
```

- »» Calling methods:

```
object.method()  
car.start_engine(), car.stop(), car.brake(), car.accelerate()
```

CREATING OBJECTS WITH CONSTRUCTOR

- `__init__` is a *special* method in Python classes.
- It is called a **constructor**.
- A constructor is automatically called when an object is created.
- Using a constructor, you can set the attributes of an object.
- In Python, when you want to call anything belonging to the class, you need to use the word `self` to refer to that instance.



CREATING AN OBJECT

- »» Lets create a Car object and two instances of that car
- »» See File: L6E1.py