

Package, in simpler words - folder where your class lives in

```
package 10.codelex.syntax;
```

Static field - field which can be considered global for all instances of dogs

```
/**  
 * Basic class syntax example  
 */
```

Class declaration
ALWAYS matches constructor name

Static field declaration,
given type of String and name DOG_EMOJI.
Uppercase because it is static & final - constant

```
public class Dog {  
    private static final String DOG_EMOJI = "\uD83D\uDC15";
```

```
    private String name;  
    private int age;
```

Instance variables
- has a type and name defined

```
    public Dog(String name, int age) {  
        this.name = name;  
        this.age = age;  
    }
```

Constructor, there can be no argument
or argument (as in example)
constructors which can accept values.
It is used for the creation of new objects.

Methods - describes behaviour of the dog.

```
    public double calculateAgeInDogYears(int coefficient) {  
        return (double) age / coefficient;  
    }
```

Method which returns
double and accepts
int parameter

```
    public void introduce() {  
        System.out.println("Hello! This is dog, my name is "  
            + name + " and I am " + age + " years old " + DOG_EMOJI);  
    }
```

Method which is not
returning anything
and does not need
an argument
to be called also

```
    public static void main(String[] args) {  
        int coefficient = 7;  
        Dog bob = new Dog("Bob", 5);  
        bob.introduce();  
        double ageInDogYears = bob.calculateAgeInDogYears(coefficient);  
        System.out.println("Bob's age in dog years is " + ageInDogYears);  
    }
```

Static method - this method
does not belong to the current dog

We are using a constructor to
create new instances of dogs

Calling method on bob object
and passing previously
defined argument

Calling no argument method
on previously created bob object

Assigning value of newly created dog
to a variable named bob of type Dog