

Programming Paradigms

Procedural programming: Specially in large, complicated programs modularity is generally desirable. Inputs are usually specified in terms of arguments and outputs delivered as return value.

eg:-

```
int max(int n1, int n2){  
    if (n1 > n2)  
        return n1;  
}
```

inputs: int n1 and int n2

output (type): int

Object oriented programming (OOP): OOP is a programming paradigm that uses objects and their interactions to design applications and computer programs.

program: for a single task

application: a set of programs

It is based on several techniques including encapsulation, modularity, polymorphism, and inheritance.

Fundamental concepts of OOP:

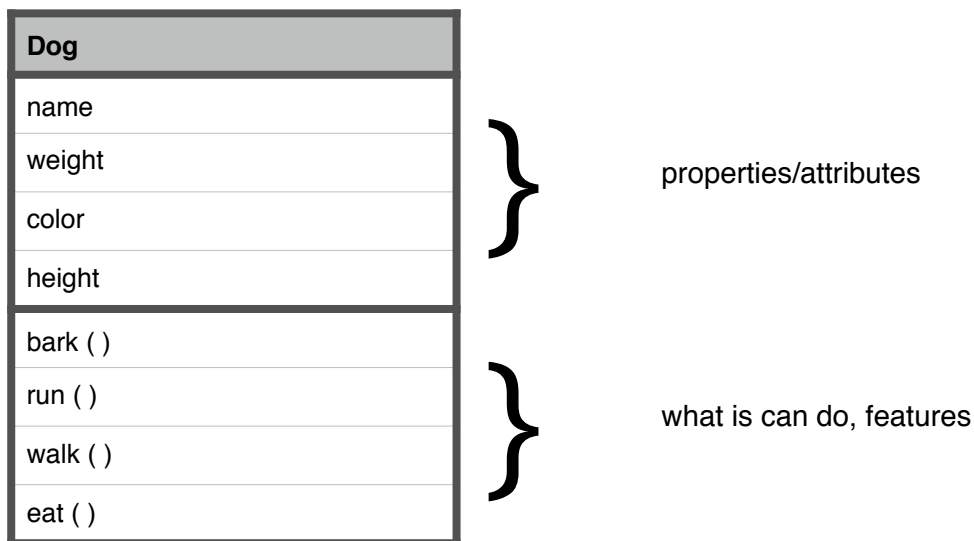
Class: A class defines the abstract characteristics of a thing (real-world object) including its characteristics (properties or attributes) and its behavior (what it can do, features).

eg:-

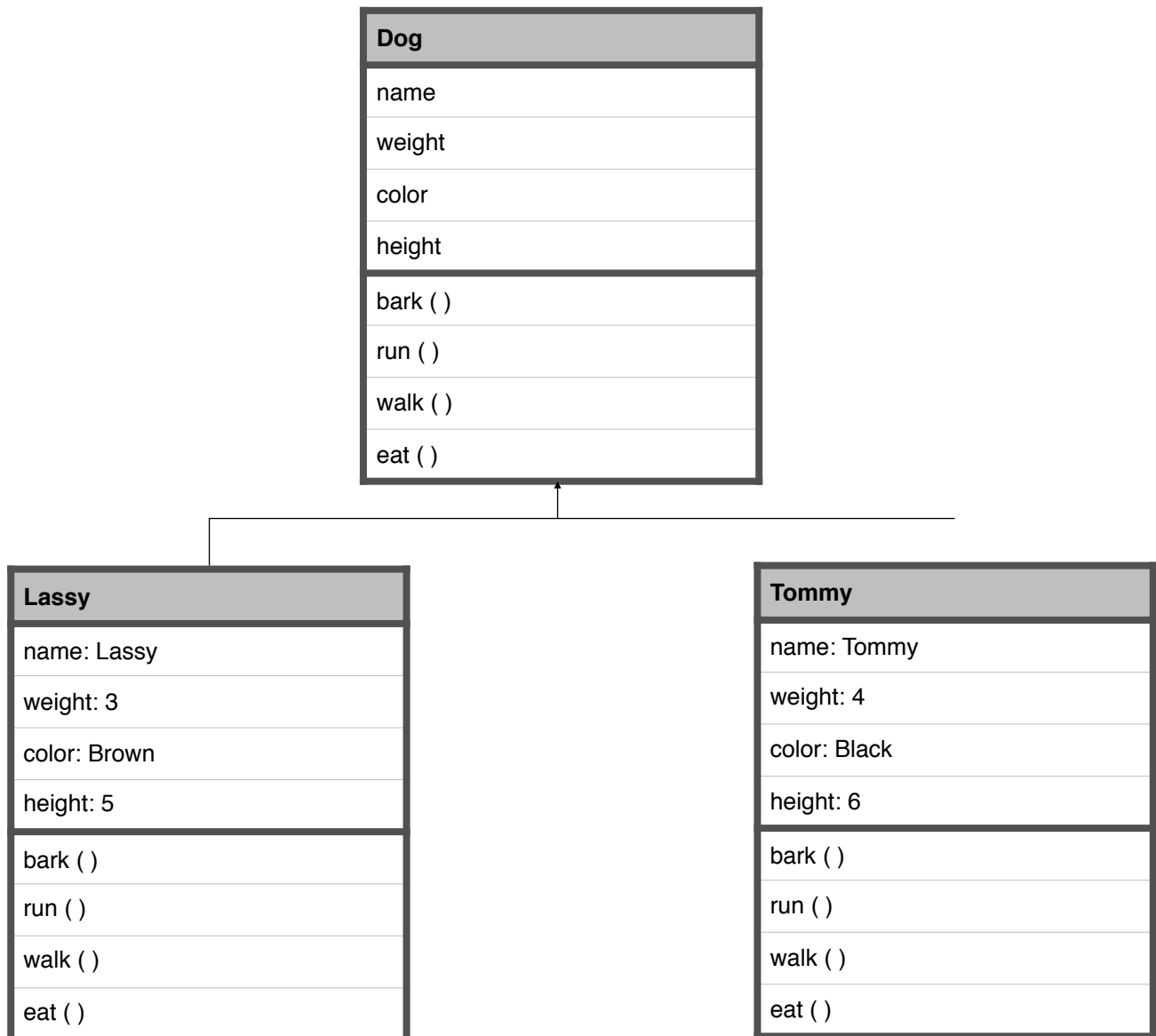
a class called "Dog"

attributes/properties: color, height, weight, name

what it can do: bark, run, eat, walk



Objects: A particular instance of a class. The class of a Dog defines all possible dogs listing the characteristics and behavior they can have. The object “Lassy” is one particular dog with particular version of characteristics.



The object “Lassy” is an instance of the Dog class. The set of values of the attributes of a particular object is called its state.

Methods: An object’s attributes “Lassy” being a dog has the ability to bark. So bark () is one of Lassy’s methods, She ay have other methods as well such as sit (), eat ().

Message passing: The process by which an object sends data to another object or asks the other object to invoke a method.

We will talk about Inheritance, Encapsulation and Polymorphism in future classes. For the time being read from Internet and they to understand.

Comparison of OOP and procedural programming:

Procedural	OOP
focus on breaking down a problem into data structures and subroutines (functions, methods)	breakdown the problem into objects

Terms used in OOP and procedural programming:

Procedural	OOP
functions	methods
modules (set of functions grouped together)	objects
variables	attributes

Identifying classes for a given problem:

1. read the description of the problem domain written in textual language
2. identify names and noun phrases (candidate classes)
3. identify verbs (candidate methods)
4. identify properties (candidate attributes)
5. refine candidate classes until you are satisfied

exercise: draw a class diagram for the following description.

The weather station sends the summary of the wether data that has been collected from the instruments in the collection period to the wether data collection system. The data send are the maximum, minimum and average ground and air temperature, Maximum, minimum and average air pressure and wind speed, the total rainfall and the wind direction samples at 5 minutes rate.

While you are trying to do this consider one-to-one, one-to-many and many-to-many relationships between classes. Read and study about one-to-one, one-to-many and many-to-many relationships from Internet.