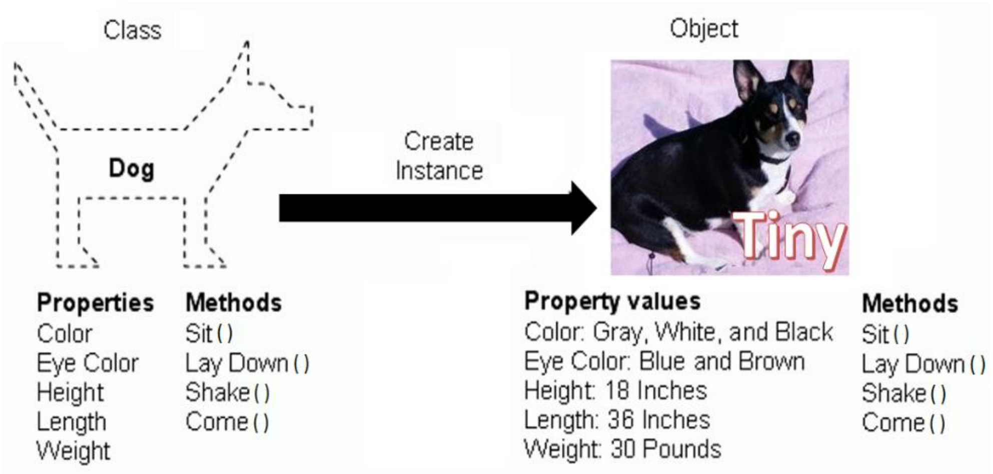
Object-Oriented Design

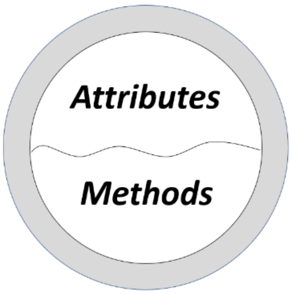
What is a Class?

Creating a Class is like creating a variable, it's just a definition of a data type until it gets assigned a value, once you have an instance of a Class, we call it an Object. If our class is "Dog", we could have objects "Tiny", "Rover", "Rusty", and "Fido".



What is an Object?

An object has:
Attributes and Methods



- *Attributes* (or *Features*) are a collection of variables.
- *Methods* (or *Behaviours*) are a collection of functions.
- The Attributes and Methods are *encapsulated* or contained in the object.
- The object can be configured so that some Attributes and Methods are private to the object, and others are visible to other objects, this is *Information Hiding*.
- The public elements (Attributes and Methods) of the class are referred to as the *Interface* (or *Public Interface*).

Object-Oriented Concepts

Abstraction
Abstraction means dealing with the level of detail that is most appropriate to a given task. For example, a driver and a mechanic interact with a car at different levels of abstraction.

Inheritance
Inheritance means that one class can inherit attributes and methods from another class. So we look for classes that have a lot in common, and create a single class with those commonalities.

Composition
Composition means collecting several objects together to create a new one. It is usually a good choice when one object is part of another object.

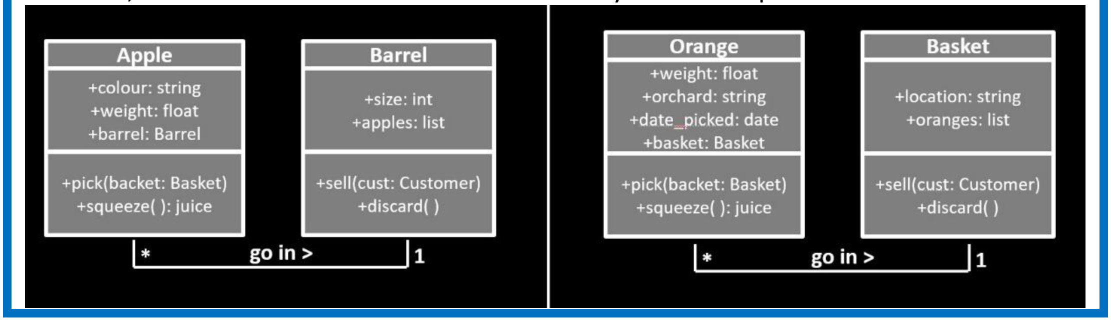
Polymorphism
Polymorphism is the ability to treat a class differently depending on which subclass is implemented. The appropriate subclass is determined based on parameters passed in.

We'll look at these concepts in more detail later.

Object-Oriented Design

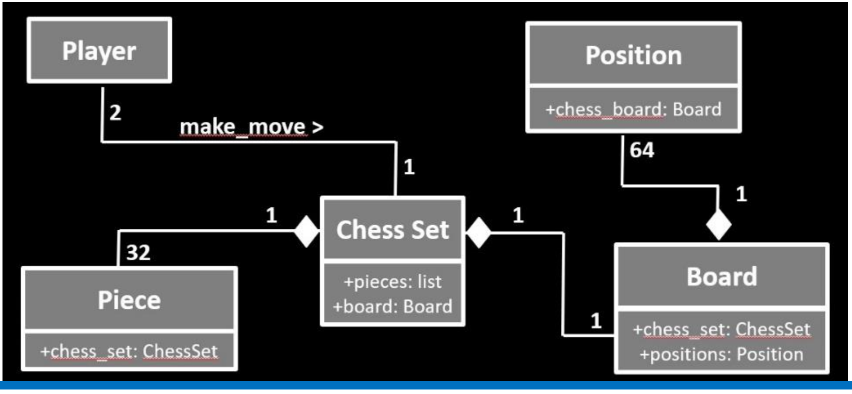
Class Diagrams

A Class Diagram shows the design of a system showing the classes, and their attributes and methods. It also shows the relationship between classes, with a line (association) between classes, with a verb to describe the relationship, and cardinality is indicated by numbers, in this case 1…* indicates a one-to-many relationship.



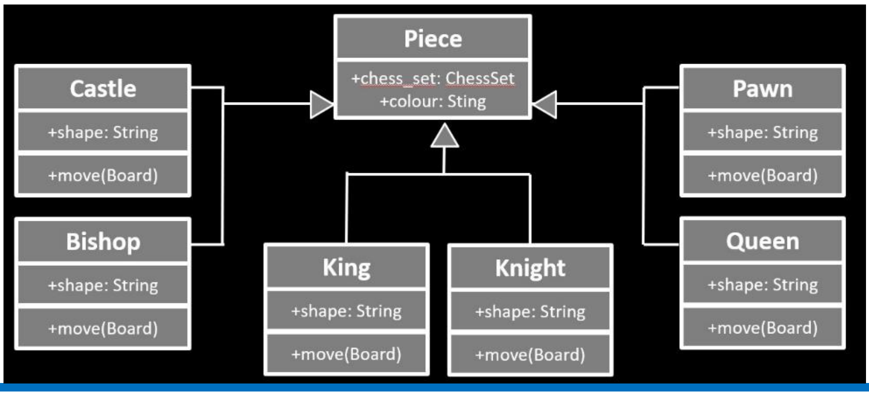
Class Diagrams (composition)

A Class Diagram can show composition using a solid diamond to indicate that one class is composed of another class, with the cardinality indicated by numbers.



Class Diagrams (Inheritance)

A Class Diagram can show inheritance using an arrowhead to indicate that one class is inherits from another class.



We'll look at these concepts in more detail later.