# Intro to Class and Objects

1. A class
   1. Is in its own .java file (at this stage)
2. Its attributes become
   1. Private instance variables with
   2. Each with a get and a set method
   3. Use ***this.***to change the instance variable
3. A class can have one or more Constructor methods they
   1. Are the same name as the class
   2. Do not have a return type specified
   3. Often set some of the attributes of the newly instantiated object
   4. Constructors can call each other using this()
4. A class can have both private and public methods that receive and return values
5. Static variables and methods are Class elements and there is only one of them for all of the objects of that class.
6. The class is now a java type available in your project.
7. You can use it to instantiate objects
8. You can use your new type in much the same way as any other type – e.g. make an ArrayList of the objects

# Inheritance

1. Inheritance is when a class is used as the *parent* of another class
2. A super class is one which is used as a parent.
   * It can be exactly the same as any other class
3. The sub-class **extends** the super class
   * Add the keyword extends and the superclass name after the subclass class name. E.g.:  
        
     **public class Tomato extends Vegetable{ ..}**
   * Subclass constructors must call the super class constructor as the first line of code.
   * The subclass has all the attributes, methods, types of the super class plus any new ones it declares
4. Sub-classes can also be super classes. There is no practical limit to the depth of the inheritance hierarchy.
5. A class can only extend one other class. See next lecture for how to achieve polymorphism (which lets a class inherit from multiple sources).

# Polymorphism and Interfaces

Know how to implement and use interfaces to achieve polymorphism

* 1. Interface: definition ‘the point where two things meet and interact’
     + In programming, an interface defines a set of methods and in some cases data that an object will provide such that other objects can interact with it.
  2. Polymorphism: definition ‘many forms’
     + In programming, polymorphism means that the same method call may do different things depending on the exact class of the object on which the call is made