Java – Objects & Classes

• **Object** - Objects have states and behaviours. Example: A dog has states - colour, name, breed as well as behaviours – wagging the tail, barking, eating. An object is an instance of a class.

• **Class** - A class can be defined as a template/blueprint that describes the behaviour/state that the object of its type support.

**Objects in Java**

Let us now look deep into what are objects. If we consider the real-world, we can find many objects around us, cars, dogs, humans, etc. All these objects have a state and a behaviour.

If we consider a dog, then its state is - name, breed, colour, and the behaviour is - barking, wagging the tail, running.

If you compare the software object with a real-world object, they have very similar characteristics.

Software objects also have a state and a behaviour. A software object's state is stored in fields and behaviour is shown via methods.

So in software development, methods operate on the internal state of an object and the object-to-object communication is done via methods.

**Overriding in Java.**

In any object-oriented programming language, Overriding is a feature that allows a subclass or child class to provide a specific implementation of a method that is already provided by one of its super-classes or parent classes.

**Java Adapter Classes**

Java adapter classes *provide the default implementation of listener* [*interfaces*](https://www.javatpoint.com/interface-in-java). If you inherit the adapter class, you will not be forced to provide the implementation of all the methods of listener interfaces. So it *saves code*.

The adapter classes are found in java.awt.event, java.awt.dnd and javax.swing.event [packages](https://www.javatpoint.com/package). The Adapter classes with their corresponding listener interfaces are given below.