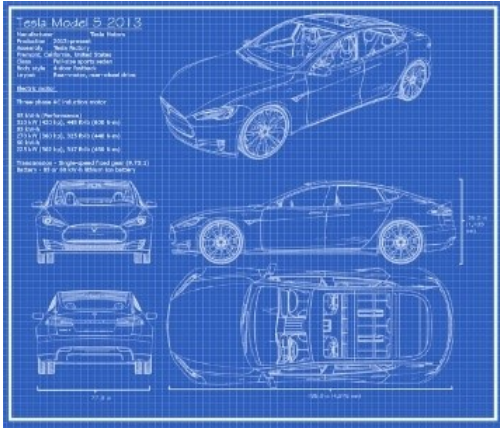


What are Classes?

A class represents the definition—the blueprint if you like—used to create objects. Objects are simply variables, created (or **instantiated**) from this blueprint.

Like a structure, a class describes the **attributes of an object**: the kinds of data it stores internally. To design a Car class, you specify the physical characteristics that car shares: its serial number, body type, color, type of interior, engine size, etc.



Such attributes are stored as the object's **data-members** (**instance variables** in Java).

A class also **describes and implements the behaviors of its objects**: the kinds of operations that each object in the class can perform. When you define a class, you need to specify **what the object can do**, providing an explicit list of its possible behaviors.

These are specified as **embedded functions**, called **member functions** in C++; in Java these are called **methods**. Member functions contain the **interface** (as prototypes in the class definition), and the instructions (appearing in the **implementation**) that tells each particular object how to complete a particular task.

To ask an object to perform some action, **invoke** or **call** one of its member functions. To emphasize the fact that objects represent fairly self-contained, autonomous units, the process of invoking a method is often called **sending a message** to the object.



This course content is offered under a [CC Attribution Non-Commercial](#) license. Content in this course can be considered under this license unless otherwise noted.