**Object:** object means a physical thing ex: person, dog car, bulb etc.

Each object has state and behaviour.

State describes the object. Name, color, height, weight, etc.

Behaviour is nothing but what an object can do. Speak, eat, and walk.

In Programming, state can be treated as data i.e variables.

Behaviours are nothing but functions/methods.

* Each object communicates with other object using their functions ex: a person communicate with other person by speaking and the other person by hearing.
* When dealing with object in programming data is important.

In OOP by interaction of object we can able to get result, that means to get results first we need to create the object.

Blueprint building //building is reality

Class object //object is reality

However to create building we need Blueprint, to create object we need class.

Vehicle car;

Animal dog;

Fruit apple;

Draw a vehicle? Draw an animal?

Animal is a living being it can walk, eat, feel and has legs.

Vehicle picks and moves goods or passengers and it has wheels.

**Class:** class is blueprint that describes state and behaviour of an object

**Encapsulation (Data hiding):**

Wrapping data and functions that operate on the data in order to secure/protect the data & it’s functions. To secure the data/function we need to impose access control.

Let’s take a person, for ex he is having o2, water, and blood, and liver, kidney, and heart which operate on the o2, water, and blood, all these are secured by our skin.

**Public** – public members accessible to everyone.

**Private** – private members accessible to you only, means within the class.

**Protected** – same as private with an exception that inherited class has access to the protected members; protected members can be accessed within the same package.

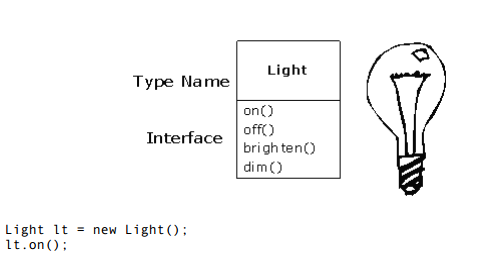
**Default (package)** – when you don’t apply the access specifiers this will come into pictures. Classes can access the members of other classes in the same package.

Exercise:

Shape class having four variables length, breadth, width, radius and all are having different access control, Circle from same package inherits Shape, Square from different package access these variables by creating object to shape.

Each class is having display method which will access to the variables.

The interface determines the requests that you can make for a particular object. However,

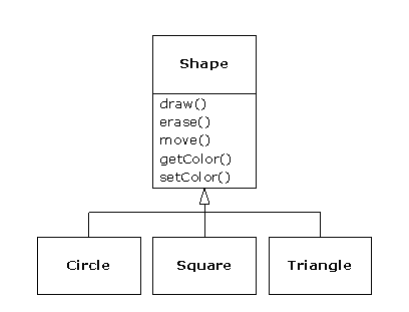
There must be code somewhere to satisfy that request.

**Inheritance:**

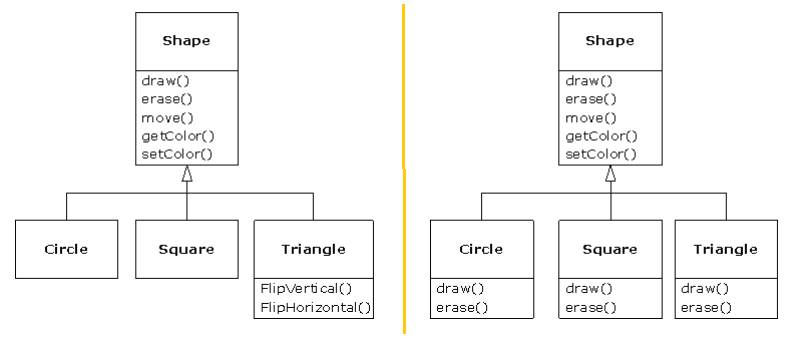
Inheritance can be defined as the process where one object acquires the properties of another.

When we talk about inheritance, the most commonly used keyword would be **extends** and **implements**. These words would determine whether one object IS-A type of another. By using these keywords we can make one object acquire the properties of another object.

You have two ways to differentiate your new derived class from the base class.



1. Add new methods to the derived class.
2. Change the behaviour of an existing base class method, its referred to as overriding.

****

**Abstraction:**

Now create a class and define an object for student class.

Call validate() to validate params

Call add() add the student.

Note:no need to show validate func to user, it can be added inside the add method.

Is nothing but showing what is necessary.

Showing what is necessary, show minimal.

class Student{

String name, String stream;

validate(String name,String stream);

add(String name,String stream);