Explanation : **NEED OF OBJECT** :

Class has its own variable, methods. Let’s say class ‘calculator’ has a method called ‘addition’, it does addition of 2 numbers. If I want to use a method of adding 2 numbers in a new class, I will not write code of addition again in a new class. Instead of that, I will create object of ‘calculator’ class and will use the already written ‘addition’ method. Object is nothing but the representator of a class; it has all properties of that class. Using object of a class you can use all the contents / stuff of that class like method, variable.

**Class** : A class is an entity that determines how an object will behave and what the object will contain. In other words, it is a blueprint or a set of instruction to build a specific type of object.

**Object :** is an instance of a class Class is a template or blueprint from which objects are created. So object is the instance (result) of a class. Example in eclipse.

**Static classes :** Java doesn't allow you to create top-level static classes; only nested (inner) static classes.





