 When I started to code, one of the first things that disturbed me was the term **OOP.**The heavy words and complex definition are enough to scare any beginner. I read about  **OOP**in many books and sites but nothing helped much, more or less everyone talked about the same things, Classes and Objects but nothing made much of sense to me. The picture became clear only after I started using it in my code. Object Oriented Programming is a vast concept and I too have many things yet to explore about this paradigm.

Here I will write about my understanding of **OOP .**Also further I will write about a project and will explain the code which helped me understand the basics behind Object Oriented approach. I will explain the same project in Java and JavaScript (Yes JavaScript is object oriented too).

Programming is a process of giving the computer a set of instructions to perform. The set of instructions are intended to solve one or more problem statements.

Types of approach

**Procedural Programming :**

If we write the desired set of instructions in such a way that they can be performed step by step to reach the final outcome, we can call it a procedural approach. In this we write procedures, which contain a series of computational steps and it follows a top to down approach to compute the output.

Example:

Step : 1 Call the Lift

Step : 2 if Lift has arrived && Space is available

Step : 2A        Get inside the lift

Step : 2B          If Desired floor button not pressed

Step : 2B1           Press the desired floor button

Step : 3  While desired floor not reached - Keep waiting

Step : 4  Get out of the lift.

**Object Oriented Programming :**

Before defining the concept of Objects and Classes, Lets read a story, about a man called Mr. X

He wakes up by the alarm in the morning, haves his breakfast, kisses good bye to his lovely wife. He gets out of his home, calls the lift of his apartment, gets down on the road, waits for a cab, gets in, goes to some place, again waits for a bus, gets in, goes to office, takes the lift in the office building, reaches his desk, works and then does all the things in reverse to get back to his home. Takes his dinner, kisses the wife a good night, sets an alarm and sleeps.  Sweet story right ?

Now think of this as a problem statement,what if we have to code the whole scenario ? We can see that we have already written a piece of code for getting down from home using the lift, similarly we can write other lines of code to wake up, to kiss the lovely wife, to take a cab, to take a bus, to go up using “office” lift, work and more......

Now what if we have to write this for Mr Y and Miss A ?  or say for every one living in a society, or in a city or a state ? Things will get more complicated as we will get a bigger picture, now the subject may be a Mr or a Miss , some one may have a not so lovely wife :). Few people might not have lifts or may be living on the ground floor, few may be using their own vehicles and so on. Now writing code for all individuals separately not only seems a tedious task but a waste of time too. **Writing long pieces of code is not what any one should worry about,(operating systems have millions of lines of code.)**its the output and efficiency that matters, never write a line longer if it could have been written in short and hence improved the time and space complexity.

If we cut down the above scenario and think it logically we can relate many things. First of all, all subjects in the statement are living beings, so all will have some properties with regard to the problem

All will have some gender

All will be living somewhere (floor number)

All will either have a lift or wont have it

All will either use own vehicle to travel or will use the public transport

All will have either have spouse or will be single

and many more things in common. Similarly if take all the lifts,whether it be Mr. X’s apartment lift, or Mr Y’s office’s lift or any other lift they all will have many things in common.

All lifts will have buttons

All will have floor indicators

All will have a maximum capacity

Now even looking with a broader look at the actions involved in the problem statement we can even generalize the actions. Be it getting down or up in the lift or waiting for a cab or waiting for a bus, all these involve few common steps at a broader level.

Calling / Waiting for **something**

To check if space is available in that **thing**

To get in that **thing**

To wait for the **destination**

To get out of that **thing**

**It will be a better idea if we mange to write same structure of code and then re use it again and again as per the need. Object Oriented Programming was developed with real world objects as inspiration. All of the world is full of objects. And all objects despite having individual existence share some properties in common with other objects forming a group. The term “Classes” was coined to describe a group with various attributes and action capabilities. Objects are the instances of these Classes, which have their own existence with individual value for all the properties of the Class.**

Complicated ? We will break down the terms in the next blog .