

I and we look a
certain way- black



**ASSIGN
BUSTER**

I recently read this thought-provoking book “ Elegant Objects by Yegor Bugayenko”. The author gives some useful tips to write more object-oriented code. Below are some of my learnings from the book. An Object is a living organism like us - a human being. We have certain properties and we look a certain way- black hair, blue eyes, hands, legs. We can do stuff, such as walk, talk, eat, sleep.

An object is just like us, a thing that has properties and can do stuff. The properties of an object are variables, and the things an object can do are functions. Abstraction allows the object to show only essential features and hide details. Similarly, living organisms can talk, walk, eat but the inside mechanism is hidden. Encapsulation is binding of data and function together as a single unit. As in humans, our legs help us walk, our hands help us hold things. Inheritance allows an object to acquire all properties and behavior of parent object which is like a parent-child relationship in living organisms. Polymorphism allows objects to redefine a function to provide it with a new definition.

In humans, if we walk using our hand and not legs, we will change the parts used to walk. This is called overloading. If we wish to walk differently but using our legs, like everyone else. Then we can walk the way we want, this will be called overriding. Like living organisms, objects during their lifecycle interact with other objects, change their state and eventually die.

Following are some of the tips that I found interesting: 1. Avoid names ending with -er (Example : Logger , Loader, Reader). These names convey

what these objects do and not what they are. If it's history of logs, name it logHistory instead of logManager.

2. A class must have only one primary constructor that initializes the state of an object and the others just prepare the data and call primary constructors. This helps in avoiding code duplication.
3. Keep constructors code-free. Constructors should only contain assignments and no computations.
4. Encapsulate as little as possible. This will encourage more modularized code that will be easier to maintain.
- 5.

The objects must be immutable i. e. its state can't be modified after it is created. Immutable objects are thread-safe. Avoids temporal coupling between lines. Reordering lines will not have any effect on application logic , same immutable object will be used.