# Java OOPs Concepts

## OOP’s Fundamentals 1 -- What is Object ?

# When we say Object in Java, for a moment forget Java. In general when we say object in real world what sort of examples can come in our mind, many right e.g. Mobile, Laptop, Head Phones, Pen, Shoes etc.

# Now let’s pick up one object and visualize it from Design perspective. Let’s take not too simple not too complex scenario, Mobile as an Example:-

# Mobile as Object

# It is made up of:

# LCD Display Screen

# Battery

# OS

# Power Button

# Headset Jack

# ……………………………..

# What it is supposed to do:

# Allow me to make a phone call

# Allow me to receive a phone call

# Allow me to check my messages

# Check Email

# Internet Surfing

# Play Games

# ………………………………….

# Now coming back to Java, from a beginner’s perspective if I have been assigned a task to design a Java Class which should replicate Mobile Object how will I go about it. Let’s try to make a rough skeleton of it.

Mobile

{

ownBatteryObject

ownOSInstance

ownPowerButton

ownHeadSetJack

ownLCDDisplayScreen

* Above are the properties which Mobile is supposed to have, In Java terms Inside Mobile Class these will be instance variables. Note these are not actual functionalities of Mobile class, it’s important to clear this what we have defined is what are the Mandatory items or in programming terms other Objects required to help Mobile achieve its target, what is the target of mobile, let’s define that in the form of methods

makePhoneCall()

{

}

receivePhoneCall()

{

}

surfInternet()

{

}

* These are the functions which Mobile is supposed to perform

}

**Summary of an Object**

* Whenever someone says we need to design a class or object in programming terms
* Step 1 – Understand the objective first what this object is supposed to serve
* Step 2 – Once motive is clear, figure out the core components required to fulfill the objective of Object
* Step 3 – Once core components are clear basically instance variables, then define the functionalities i.e methods for the object.