

understanding was not correct. If you are one of them who doesn't understand about Encapsulation, then this article is for you.

So, what exactly the encapsulation means bro? Ok. Let we have some coffee, and we continue discuss about Encapsulation.



Cool, the coffee vending machine has given tasty brewed coffees. Nice coffee! Back to our discussion, here is a simple question I'd like to ask you. Why don't you give a thought on how did this coffee vending machine worked?

You dropped in the coin, and the coffee came out of the machine in a beautiful coffee cup. So, why didn't you see how the where the coffee beans, how it is prepared? You really don't know (unless you are the one who designed the machine ;-)) how things are made inside the machine, but finally you get a coffee!! Yeah don't you see a sense of Encapsulation? So, here what is actually hidden? The Coffee beans? **No, the way how the coffee prepared is hidden.**

So, Encapsulation is hiding of process; hence you hide the data. Not the vice versa.

From the definition, **Encapsulation** is “**Information Hiding**”, the information is how **something is happening** inside an object? It's just a mere **data hiding concept** (though it includes data hiding).

So, when you model this coffee vending machine, you don't want to show out the way how a coffee is made. Isn't it? So, you don't want to show where the coffee beans, milk and the sugar are. Right?

```
Class CofeeVendingMachine
{
    Public getCoffee( int typeOfCoffee )
    {
        prepareCoffee(typeOfCoffee )
    }

    Private prepareCoffee ( int typeOfCoffee )
    {
        //coffee making process. Unnecessary to make it public.
    }
}
```

So, in the CofeeVendingMachine class, prepareCoffee() is hidden and getCoffee() is made public, so that you will be able to send a request to the machine.

Since you have hidden the process, you don't have any need to show how the coffee & milk are mixed and how it's been brewed. If you need a stylish design, even you can show that.

```
Class CofeeVendingMachine
{
    Public getCoffee( int typeOfCoffee )
    {
        prepareCoffee(typeOfCoffee )
    }

    Private prepareCoffee ( int typeOfCoffee )
    {
        //coffee making process. Unnecessary to make it public.
    }

    Private Milk, CoffeePowder, Sugar ;
}
```

The Coffee Vending Machine encapsulates the internal process and the ingredients (data) used in the process.

Object Oriented Languages provides encapsulation via modifiers such as “private”, “protected”. Sometimes, volatile and transient also acts as encapsulating modifiers, but they are language specifics.

Encapsulation doesn't stop just with OOP languages. The concept extends itself to the latest technology such as Webservices, Service Oriented Architecture. If you closely watch, any real world object has encapsulation in it, which the Object Oriented Programming tries to model.

Tagged under:

[« Object Oriented Programming - Abstraction \(/article/object-oriented-programming-abstraction.html\)](#)
[Google Docs adds print preview » \(/article/google-docs-adds-print-preview.html\)](#)

Related Posts

- [Understanding how jars are loaded into JVM from a directory in Linux \(/article/understanding-how-jars-are-loaded-into-jvm-from-a-directory.html\)](#)
- [When to re-factor your code \(/article/when-to-refactor-the-code.html\)](#)
- [IE Heisenbug \(/article/IE-heisenbug.html\)](#)
- [Go lang - how to install packages \(/article/go-how-to-install-packages.html\)](#)
- [New parents should read Freakonomics, than watching it. \(/article/young-parents-should-read-freakonomics.html\)](#)

I like it([https://twitter.com/intent/tweet?text=Nice Read: Object Oriented Programming – Encapsulation is not just hiding data!&url=http://www.maheshsubramaniya.com/article/object-oriented-programming-encapsulation-is-not-just-hiding-data.html&via=msubramaniya](https://twitter.com/intent/tweet?text=Nice+Read:+Object+Oriented+Programming+–+Encapsulation+is+not+just+hiding+data!&url=http://www.maheshsubramaniya.com/article/object-oriented-programming-encapsulation-is-not-just-hiding-data.html&via=msubramaniya))

Published on: 09 Jun 2008