encapsulation vs abstraction real world example

Asked 9 years, 1 month ago Active 1 year, 4 months ago Viewed 221k times

45

For an example of encapsulation i can think of the interaction between a user and a mobile phone. The user does not need to know the internal working of the mobile phone to operate, so this is called abstraction. But where does encapsulation fit in to this example? Could someone please shed some light on this?

oop programming-languages

38

Share Edit Follow

edited Feb 4 '15 at 14:40
LittleBobbyTables - Au Revoir
30.5k ● 25 ● 99 ● 113

asked Aug 22 '12 at 12:18
crowso
1,919 ● 8 ● 31 ● 38

2 Easy, encapsulation of the details of each handset allow apk writers to develop one app that works across all android devices. – asawyer Aug 22 '12 at 12:21

1 well i prefer a example that has nothing to do with programming. – crowso Aug 22 '12 at 12:26

1 You do understand stack overflow is for programming questions right? – asawyer Aug 22 '12 at 12:27

28 @asawyer - how is this not programming related? understanding this is critical to being a competent developer.... using a diverse spectrum of examples to understand concepts is critically important. if i were to make hasty judgements - i'd say you must not be a very good developer. – mson Aug 22 '12 at 17:48

The user does not need to know the internal working of the mobile phone to operate, THIS IS CALLED ENCAPSULATION (information hiding) and not abstraction. Why do you feel that it is same as abstraction? – variable Mar 27 '14 at 4:48

17 Answers

Active Oldest Votes

73

Encapsulation is a way to achieve *"information hiding"* so, following your example, you don't *"need to know the internal working of the mobile phone to operate"* with it. You have an *interface* to use the device behaviour without knowing implementation details.

Abstraction on the other side, can be explained as the capability to use the same *interface* for different objects. Different implementations of the same interface can exist. Details are hidden by *encapsulation*.

Share Edit Follow

edited Jul 27 '16 at 15:26

answered Aug 22 '12 at 13:58
daviooh
21k ● 35 ● 137 ● 226

63

Abstraction : you'll never buy a "device", but always buy something more specific : iPhone, GSII, Nokia 3310... Here, iPhone, GSII and N3310 are concrete things, device is abstract.

Encapsulation : you've got several devices, all of them have got a USB port. You don't know what kind of printed circuit there's back, you just have to know you'll be able to plug a USB cable into it.

Abstraction is a concept, which is allowed by encapsulation. My example wasn't the best one (there's no real link between the two blocks).

You can do encapsulation without using abstraction, but if you wanna use some abstraction in your projects, you'll need encapsulation.

Share Edit Follow

edited Sep 29 '18 at 17:20
Rishabh
13 ● 3

answered Aug 22 '12 at 12:26
zessx
65.7k ● 28 ● 123 ● 149

You can do encapsulation without using abstraction, but if you wanna use some abstraction in your projets, you'll need encapsulation – Sameer Kazi May 5 '16 at 8:57

@zessx. I am not sure how we can do encapsulation without abstraction. Whenever we are doing encapsulation it will lead to abstraction in some way or other. Is n't it ? – M Sach Oct 22 '16 at 11:31

@MSach I think my "encapsulation" example was not so good, because I talked about "several devices". Take *one* phone, which has a USB port. We're hidding you sensitive information, as the printed circuit, but you've access to a simplified and secured USB interface. This has nothing to do with abstraction. – zessx Oct 22 '16 at 12:15

@zessx But even `We're hidding you sensitive information, as the printed circuit,..` we abstracting out the unnecessary details. I agree your example is not complete abstraction but its abstraction in some fashion. Though its a debatable topic always :). i posted mine answer at stackoverflow.com/questions/8960918/... based on summary of various similar links on SO – M Sach Oct 22 '16 at 13:19

24

In General words,Abstraction is Just Hiding the complex things behind a particular Procedure to make the procedure look simple. Example:**Monitor ON/OFF**::--The user doesn't need to know much about all the chips functioning that happens when Monitor is switched ON or OFF..All he needs to know is On Function ON-Monitor is On and on function OFF-Monitor is off...

Or Better Look for a car--Everyone Knows that There's a special Gear machine Which changes the gear,nobody bother to know what all functionality undergoes for a gear to change..So,That's abstraction(avoiding unwanted implementations to prevent Complexity).

So,If a developer provides a good abstraction, users won't be tempted to peek at the object's internal mechanisms.

Abstraction is achieved by making class abstract having one or more methods abstract. Which is nothing but essential characteristic which should be implemented by the class extending it. `e.g.` when you inventing/designing a car you define a characteristics like car should have 4 doors, break, steering wheel etc… so anyone uses this design should include this characteristics. Implementation is not the head each of abstraction. It will just define characteristics which should be included.

Encapsulation is restricting a user to follow a particular procedure to access control of a particular process.It Just provides safety and ensures system robustness.

Example:We can consider The HR in a company as a person that works on the principle of Encapsulation `.i.e.` we cannot talk to other departments directly we need to communicate through them through HR.This ensures security and better maintenance of company's records.

Together we can take example of a `UNDER CONSTRUCTION BUILDING..` where we can say that things like 'no. of managers' required,Types of Materials,No of workers etc as abstraction as they need to there in every Building Construction.

But,at the same time,Inclusion of every such field into a CONTRACTOR which acts as a mediator between the workers and the Building-Investor can be looked upon as Encapsulation. As,It hides all the above properties into one Entity.

> Hence If you would have understood till now you can say that abstraction is just a subset of ENCAPSULATION.i.e.Every entity that performs abstraction is encapsulated internally but every thing that shows encapsulation need not be abstraction always.

e.g. `.ToString()` Method defined in almost every class is implementation of `Abstraction` because We don't the functionaltiy Within,all we care is that it changes almost everything to string.And as it assembles a s a unit,it is encapsulated too..But,The private members that we hide and access through `Properties` is an example of encapsulation only as it is done basically keeping data security in mindd..!!

Hope This answers your Question..!!

Share Edit Follow

edited Mar 29 '13 at 10:06

community wiki
2 revs, 2 users 92%
Rahul Ranjan

Nice explanation. I was looking at an answer which explains these two terms with an example while also highlighting the subtle differences between them. This does so perfectly. Upvoted. Thanks. – user720694 Jun 17 '13 at 6:58

19

Encapsulation is to hide the variables or something inside a class, preventing unauthorized parties to use. So the public methods like getter and setter access it and the other classes call these methods for accessing

Abstraction involves the facility to define objects that represent abstract "actors" that can perform work, report on and change their state, and "communicate" with other objects in the system.

Consider the below real time example:

Encapsulation: **As a driver you know how to start the car by pressing the start button and internal details of the starting operations are hidden from you. So the entire starting process is hidden from you otherwise we can tell starting operation is encapsulated from you.**

OR

The driving wheel is encapsulated the process of rotating the wheel from you.

Abstraction:

Before mentioning anything about abstraction, we can take three different users here (I am calling them as entity)

1) You 2) Local Mechanic 3) Expert

You Entity: Since you know only to start the car by pressing a button and all other operations behind the scene are abstracted from you.

Local Mechanic Entity: Our local mechanic knows some of the implementation of starting the car, i.e. he can open car's bonnet and check the battery cable or chock etc. So in short Local Mechanic Entity knows some of the implementations of the car.

Expert Entity: Since our expert (Designer of the car) mechanic knows all the operations of our car, he can repair it very quickly. So in short Expert Entity knows all the implementations of the car.

The car's operation is completely abstracted from you and it is partially implemented to Local Mechanic Entity and fully implemented to Expert Entity. So you are an abstract class having only abstract methods, Local Mechanic Entity has extended You(Since he is also an ordinary user) and he implemented some of the methods and last our expert Entity extending Local Mechanic and implementing all the methods.

I think this is a good example.

Share Edit Follow

edited Jun 20 '20 at 9:12
Community Bot
1 • 1

answered Feb 4 '15 at 5:32

Sameer Kazi
15.8k • 2 • 31 • 46

1 Best example with easy understanding – Dhruvin shah May 18 '16 at 2:35

1 Best Explanation. I am looking for explanation with code. – Gopal Jun 15 '17 at 7:28

14

Everything has many properties and behaviours so take whatever object you want TV, Mobile, Car, Human or anything.

Abstraction:

1. Process of picking the essence of an object you really need
2. In other words, pick the properties you need from the object Example:
   a. TV - Sound, Visuals, Power Input, Channels Input.
   b. Mobile - Button/Touch screen, power button, volume button, sim port.
   c. Car - Steering, Break, Clutch, Accelerator, Key Hole.
   d. Human - Voice, Body, Eye Sight, Hearing, Emotions.

Encapsulation:

1. Process of hiding the details of an object you don't need
2. In other words, hide the properties and operations you don't need from the object but are required for the object to work properly Example:
   a. TV - Internal and connections of Speaker, Display, Power distribution b/w components, Channel mechanism.
   b. Mobile - How the input is parsed and processed, How pressing a button on/off or changes volumes, how sim will connect to service providers.
   c. Car - How turning steering turns the car, How break slow or stops the car, How clutch works, How accelerator increases speed, How key hole switch on/of the car.
   d. Human - How voice is produced, What's inside the body, How eye sight works, How hearing works, How emotions generate and effect us.

ABSTRACT everything you need and ENCAPSULATE everything you don't need ;)

Share Edit Follow

answered Dec 1 '15 at 5:40

Sandeep Kumar
1,415 • 1 • 15 • 23

2 its best definition till i read of abstraction Vs encapsulation..nice... – vimalraturi Oct 21 '16 at 4:38

2 `ABSTRACT everything you need and ENCAPSULATE everything you don't need` - simple and effective – Adrian Iftode Jun 13 '17 at 20:24

9

The wording of your question is odd - Abstraction vs Encapsulation? It should be - someone explain abstraction and encapsulation...

Abstraction is understanding the essence of the thing.

A real world example is abstract art. The artists of this style try to capture/paint the essence of the thing that still allows it to be the thing. This brown smear of 4 lines captures the essence of what a bull is.

Encapsulation is black boxing.

A cell phone is a great example. I have no idea how the cell phone connects to a satellite, tower, or another phone. I have no idea how the damn thing understands my key presses or how it takes and sends pictures to an email address or another phone number. I have no idea about the intricate details of most of how a modern smart phone works. But, I can use it! The phones have standard interfaces (yes - both literal and software design) that allows someone who understand the basics of one to use almost all of them.

How are the two related?

Both abstraction and encapsulation are underlying foundations of object oriented thought and design. So, in our cell phone example. The notion of a smart phone is an abstraction, within which certain features and services are encapsulated. The iPhone and Galaxy are further abstractions of the higher level abstraction. Your physical iPhone or Galaxy are concrete examples of multiple layers of abstractions which contain encapsulated features and services.

Share Edit Follow

answered Aug 22 '12 at 17:44

mson
7,646 • 6 • 37 • 70

8

Abstraction Means We focus on the essential qualities of some thing rather than one specific example and we automatically discard what is unimportant or irrelevant.

Example We are writing a bank account class,essential qualities of bank account are Opening date, Account title,Account number,Balance etc...

Encapsulation Means the idea of capsulation or surrounding some thing not just to keep the content together but also to protect and restrict form accessing out side.Along with secrecy It's about reducing dependencies between different parts of the application.

Example In our Bank account class Someone accessing the attribute of Balance and trying to change it ,Attempt can be successful if there is no encapsulation.

Share Edit Follow

answered Dec 1 '14 at 18:25

Shan
131 ● 2 ● 4

4

Encapsulation is hiding information.

Abstraction is hiding the functionality details.

Encapsulation is performed by constructing the class. Abstraction is achieved by creating either Abstract Classes or Interfaces on top of your class.

In the example given in the question, we are using the class for its functionality and we don't care about how the device achieves that. So we can say the details of the phone are "abstracted" from us.

Encapsulation is hiding WHAT THE PHONE USES to achieve whatever it does; Abstraction is hiding HOW IT DOES it.-

Share Edit Follow

answered Jun 27 '16 at 6:28

Navin Israni
1,151 ● 3 ● 13 ● 26

3

Encapsulation helps in adhering to Single Responsibility principle and Abstraction helps in adhering to Code to Interface and not to implement.

Say I have a class for Car : Service Provider Class and Driver Class : Service Consumer Class.

For Abstraction : we define abstract Class for CAR and define all the abstract methods in it , which are function available in the car like : changeGear(), applyBrake().

Now the actual Car (Concrete Class i.e. like Mercedes , BMW will implement these methods in their own way and abstract the execution and end user will still apply break and change gear for particular concrete car instance and polymorphically the execution will happen as defined in concrete class.

For Encapsulation : Now say Mercedes come up with new feature/technology: Anti Skid Braking, while implementing the applyBrake(), it will encapsulate this feature in applyBrake() method and thus providing cohesion, and service consumer will still access by same method applyBrake() of the car object. Thus Encapsulation lets further in same concrete class implementation.

Share Edit Follow

edited Feb 17 '15 at 21:32

Prav001
323 ● 1 ● 5 ● 11

answered Apr 26 '14 at 21:20

user3576978
47 ● 1

2 (-1) Too much technical, not as "Real World" related, as mentioned in the title – umlcat Dec 1 '14 at 18:59

3

I feel like encapsulation may make more sense to discuss when you see HOW NOT TO DO in programming. For example, consider a Car class as below.

```
class Car{
  public float speed =0;
  public boolean isReverse = false;
  public boolean isStarted = false;
}
```

The client code may use above car class as below.

```
class Main{
  public static void main(args String[]){
   Car car = new Car();
   // No need to start??
   car.speed = 100; // Turbo mode directly to 100
   car.speed = 0; // Turbo break
  }
}
```

See more at: http://brevitaz.com/encapsulation-example-benefits-java/

This is uncontrolled access to car speed and other variables. By encapsulation, Car class can have complete control over how the data variables within car class can be modified.

Any concrete entity that has some behavior is example of Encapsulation. The behavior is provided by wrapping up something and hiding something

from client.In case of mobile, it is signals, chips, circuits, battery and so on.

For abstraction of the same example - normal user may say I am ok with anything using which I can make calls and receive calls. This abstraction can be substituted by any concrete mobile. Check out Abstraction examples.

Share Edit Follow

edited Jun 16 '15 at 5:21

answered Sep 23 '13 at 16:42

Vishal Shukla

2,448 ● 1 ● 14 ● 18

I don't quite understand what you mean by your first sentence. It sounds like you are then going to provide examples of bad programming to illustrate where encapsulation is useful, but you didn't do that. Perhaps you should edit your response and either rephrase that initial sentence to mean something different, take it out of your response altogether, or add examples of what not to do in programming. – Derek Sep 23 '13 at 17:03

Really? I think I did "what not to do" in that example. Its bad practice to have public speed variable and directly changing it without use of accessor methods. – Vishal Shukla Dec 18 '14 at 11:24

My comment was left after your very first revision of your answer, in which you had no programming examples listed. Looking at the time stamps of the edits, you added the example code after I left my comment. – Derek Dec 19 '14 at 16:24

Apologies. Lost the track of that. – Vishal Shukla Dec 20 '14 at 9:13

1

Let me give my 2 cents of a real-world example-analogy close to IT.

Lets say you have a subscription site, e.g a wordpress site

Each user has a role, eg admin, subscriber and so on. Many users can be admins, subscribers etc..

So abstraction here is *reflected* in the fact that **any user with admin role can do a set of things, it does not matter which specific user this is** (this is an example of **abstraction**).

On the other hand, subscriber users do not have access to certain settings of the site, thus **some internals of the application are encapsulated for plain subscribers** (this is an example of **encapsulation**)

As one can see abstraction and encapsulation are relative concepts, they apply with respect to something specific.

One can follow this line of reasoning and explain **polymoprhism** and **inheritance**.

For example super-admin users could do all the things admin users could do, plus some more. Moreover, if admin roles get an update in functionality, super-admins would get the same update. Thus one can see here an example of **inheritance**, in that super-admin roles *inherit* all the properties of admin roles and extend them. **Note** that for most part of the site, admins are interchangeable with super-admins (meaning a super-admin user can easily be used in place of an admin user, but not vice-versa in general).

Share Edit Follow

answered Dec 16 '15 at 23:37

Nikos M.

7,052 ● 3 ● 28 ● 40

0

If you have seen important TV machine, TV connections and TV color tube is hidden inside the TV case which is not been exposed for viewers like us and exposed only neccessary things of a TV like TV Channel keys, TV volume keys, ON/OFF switch, Cable Switch and TV remote control for viewers to use it.This means TV machine, TV connections and TV color tube is an unwanted data and not needed for viewers to see is been hidden from outside the world

So encapsulation means hiding the important features of a class which is not been needed to be exposed outside of a class and exposing only necessary things of a class. Here hidden part of a class acts like Encapsulation and exposed part of a class acts like Abstraction.

Share Edit Follow

answered Jul 10 '14 at 14:30

user2734846

0

Abstraction

We use many abstractions in our day-to-day lives.Consider a car.Most of us have an abstract view of how a car works.We know how to interact with it to get it to do what we want it to do: we put in gas, turn a key, press some pedals, and so on. But we don't necessarily understand what is going on inside the car to make it move and we don't need to. Millions of us use cars everyday without understanding the details of how they work.Abstraction helps us get to school or work!

A program can be designed as a set of interacting abstractions. In Java, these abstractions are captured in classes. The creator of a class obviusly has to know its interface, just as the driver of a car can use the vehicle without knowing how the engine works.

Encapsulation

Consider a Banking system.Banking system have properties like account no,account type,balance ..etc. If someone is trying to change the balance of the account,attempt can be successful if there is no encapsulation. Therefore encapsulation allows class to have complete control over their properties.

Share Edit Follow

answered Mar 15 '17 at 13:19
Joby Wilson Mathews
8,550 ● 3 ● 46 ● 44

0

I guess an egg shell can be consider the encapsulation and the contents the abstraction. The shell protects the information. You cant have the contents of an egg without the shell.,,LOL

Share Edit Follow

answered Oct 4 '13 at 22:24
Humberto
1

1 Can you please provide some context by expanding your answer... this is most strange when read without. – Ben Oct 4 '13 at 22:44

0

Just think of *Abstraction* as hiding the keypad and display screen details, *Encapsulation* as hiding the internal circuitry that binds them.

Share Edit Follow

answered Sep 5 '14 at 9:35
user3757331
3 ● 2 ● 2

0

Abstraction

It is used to manage complexities of OOPs. By using this property we can provide essential features of an object to the user without including its background explanations. For example, when sending message to a friend we simply write the message, say "hiiii" and press "send" and the message gets delivered to its destination (her,friend). Here we see abstraction at work, ie we are less concerned with the internal working of mobile that is responsible for sending and receiving message

Share Edit Follow

edited Jun 25 '15 at 16:48
mccainz
3,398 ● 5 ● 33 ● 44

answered Jun 25 '15 at 16:10
Akash Singh
1

0

Abstration which hide internal detail to outside world for example you create a class(like calculator one of the class) but for end use you provide object of class ,With the help of object they will play and perform operation ,He does't aware what type of mechanism use internally .Object of class in abstract form .

Encapsulation is the technique of making the fields in a class private and providing access to the fields via public methods. If a field is declared private, it cannot be accessed by anyone outside the class, thereby hiding the fields within the class. For this reason, encapsulation is also referred to as data hiding.For example class calculator which contain private methods getAdd,getMultiply .

Mybe above answer will help you to understand the concept .

Share Edit Follow

answered Sep 19 '14 at 11:51
Apurva
1

Highly active question. Earn 10 reputation (not counting the association bonus) in order to answer this question. The reputation requirement helps protect this question from spam and non-answer activity.