<http://www.codeproject.com/Questions/298285/difference-between-data-hidding-encapsulation-abst>

[difference between data hidding, encapsulation, abstraction](http://www.codeproject.com/Questions/298285/difference-between-data-hidding-encapsulation-abst)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  |  |  |  |  | |

**See more:** [C#](http://www.codeproject.com/script/Answers/List.aspx?tab=active&tags=81)[programming](http://www.codeproject.com/script/Answers/List.aspx?tab=active&tags=932)[oops](http://www.codeproject.com/script/Answers/List.aspx?tab=active&tags=1176)

What is the difference between abstraction,encapsulation,data hiding?Where they implemented in c#.  
please explain in terms of general life example and Programming point of view. ?

**Posted** 11 Dec '11 - 4:02

[awaisdar](http://www.codeproject.com/script/Membership/View.aspx?mid=7759905)567

[Add a Solution](http://www.codeproject.com/Questions/298285/difference-between-data-hidding-encapsulation-abst" \l "NewEntryGroup)

**Comments**

[Monjurul Habib](http://www.codeproject.com/script/Membership/View.aspx?mid=3500082) - 11 Dec '11 - 23:08

is this the only thing you can do ??..down vote ??

7 solutions

* Top Rated
* [Most Recent](http://www.codeproject.com/Questions/298285/difference-between-data-hidding-encapsulation-abst?tab=mostrecent)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | [Sign Up](http://www.codeproject.com/Questions/298285/difference-between-data-hidding-encapsulation-abst#SignUp) to vote | bad | |  | | --- | | http://www.codeproject.com/script/Ratings/Images/stars-fill-md.png  http://www.codeproject.com/script/Ratings/Images/stars-empty-md.png | | good |  | |

Solution 8

There are a number of excellent answers here, but let me toss in another point of view, if I may. The jargon in this industry is a good example of data hiding, in many cases, and being a simple person of simple mind, perhaps I can simplify.  
   
First, Abstraction is simply separating the top level usefulness of a thing from the details of its implementation. As a user of a control, you really don't care how the control does its job, what kernel operations it invokes, what interrupts it might raise. You only care what methods it has for you to call, and what results to expect of them. That's abstraction at work; keeping the details away from the consumer of the object, so that the programmer is freed from worrying about details that aren't relevant to his assignment.  
   
Second, Encapsulation is simply combining the data members and functions into a single entity called an object. This may seem trivial to you, but when it was first proposed it was very hard to grasp. In the early days we had, within a program, a data region and a code region, dedicated at compile time. The concept of objects, which were self-contained entities comprised of their own functions and data items was really hard to embrace, but ultimately it was very useful to understand and adopt. When combined with the concept of abstraction, this gave us the opportunity to write truly reusable code. So long as we didn't change the methods and data members exposed to users, we could modify the internal implementations of our objects as much as we wanted to, without breaking any applications which depended on them.  
   
Lastly, Data Hiding has to do with restricting access to internal variables used by an object to perform its magic. Originally, data hiding was proposed to protect these internal data items from being modified by users of an object. The Private access modifier was introduced to provide that protection. The concept has evolved to embrace protection of all object data members from outside manipulation, by introducing the concept of Properties (in C#, at least). As a rule, member data items are not accessible outside of the class which defines them. The programmer must explicitly define a get or set method to allow another object to read or modify these values.  
   
Although many programmers today take these concepts for granted, they were world shaking ideas when they first appeared, and very difficult to grasp. I can recall when Turbo Pascal 5.5 was released, and that was my first introduction to OOP and these fundamental concepts. It shattered everything I knew about programming, and forced me into a whole new way of thinking about software. I wonder what the next great paradigm shift will bring? You'll probably be around to see it, but I might not, thankfully...

[Permalink](http://www.codeproject.com/Answers/298421/difference-between-data-hidding-encapsulation-abst)

**Posted** 11 Dec '11 - 17:32

[Roger Wright](http://www.codeproject.com/script/Membership/View.aspx?mid=3484)**138K**

**Comments**

[sanjy1](http://www.codeproject.com/script/Membership/View.aspx?mid=8479784) - 11 Dec '11 - 23:34

this is awsome. i must appreciate you. 5!

[awaisdar](http://www.codeproject.com/script/Membership/View.aspx?mid=7759905) - 11 Dec '11 - 23:35

nicely explained. thanks! do hit back with your answers

[Abhinav S](http://www.codeproject.com/script/Membership/View.aspx?mid=4498360) - 12 Dec '11 - 0:45

Very well answered. 5.

[R.SIVAA](http://www.codeproject.com/script/Membership/View.aspx?mid=8150469) - 18 Jul '12 - 4:08

Encapsulation and Data Hiding explanation and history given was really excellent.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | [Sign Up](http://www.codeproject.com/Questions/298285/difference-between-data-hidding-encapsulation-abst#SignUp) to vote | bad | |  | | --- | | http://www.codeproject.com/script/Ratings/Images/stars-fill-md.png  http://www.codeproject.com/script/Ratings/Images/stars-empty-md.png | | good |  | |

Solution 9

**Encapsulation:**  
  
means grouping up of related members (variables and functions) into a single unit called class.  
 **RealWorldExample:**  
In our everyday lives, most of us use a video recorder via either the controls on the front of it or via a remote control. We all know which buttons to press in order to watch a program or record a program, this is done via the interface we have with the video recorder. The manufacturer can change the internal workings of the hardware, but this would not often affect us as a user because the interface we have would be the same. That is a play button, a record button, fast forward, rewind, stop, pause, etc.  
 **Coding Encapsulation**  
  
The following code demonstrates a class containing some data and a method that acts upon the data (without data hiding):

http://www.codeproject.com/images/minus.gif Collapse | [Copy Code](http://www.codeproject.com/Questions/298285/difference-between-data-hidding-encapsulation-abst)

class Counter

{

public int Count;

public Counter()

{

Count = 0;

}

public void incCount()

{

Count++;

}

}

Although the member variable can be directly accessed, the data and its method is encapsulated in a class.  
  
**DataHidding:**  
  
Data hiding is linked to encapsulation; however data hiding is not encapsulation as it is sometimes described in this way.  
  
Data hiding is simply the means to remove direct access to an object’s information, by providing operations that perform actions on the data. This way any changes to the value of the data must come through the interface to the data, which is an operation. Thus we use access operations or properties.  
 **An Example**  
  
   
In our person object, access to the data forename is supplied through the access operations set forename() and get forename().  
Coding Data Hiding  
  
The following code demonstrates a class containing some data that is hidden:

http://www.codeproject.com/images/minus.gif Collapse | [Copy Code](http://www.codeproject.com/Questions/298285/difference-between-data-hidding-encapsulation-abst)

class Counter

{

private int Count;

public Counter()

{

Count = 0;

}

public void setCount( int newVal )

{

Count = newVal;

}

public int getCount()

{

return Count;

}

}

   
This piece of code is also encapsulated, showing that you can have encapsulation without data hiding, but you cannot have data hiding without encapsulation.  
  
   
**Abstraction:**  
  
   
  
As to abstraction, it's a philosophical category which you should be familiar first. In programming, this is… all you do in programming. When you create a method, this is abstraction, because you abstract out of the idea of doing some code here and not and allow to call it from many places. When you add a method parameter, you abstract the method out of the hard-coded value. Every type is abstraction, etc.