# Encapsulation in Java

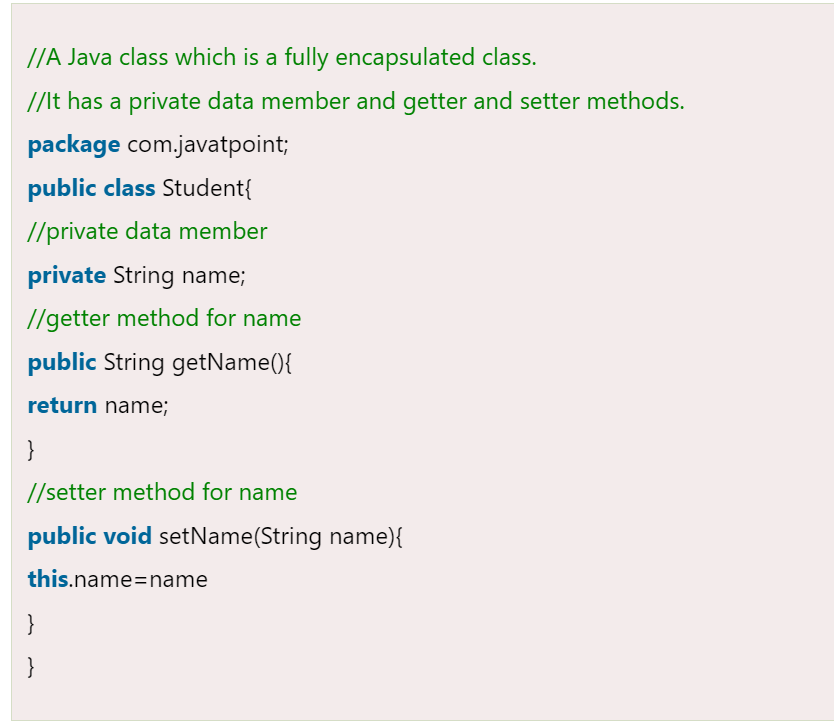
* **Encapsulation in Java** is a *process of wrapping code and data together into a single unit*, for example, a capsule which is mixed of several medicines.
* We can create a fully encapsulated class in Java by making all the data members of the class private. Now we can use setter and getter methods to set and get the data in it.
* The **Java Bean** class is the example of a fully encapsulated class.

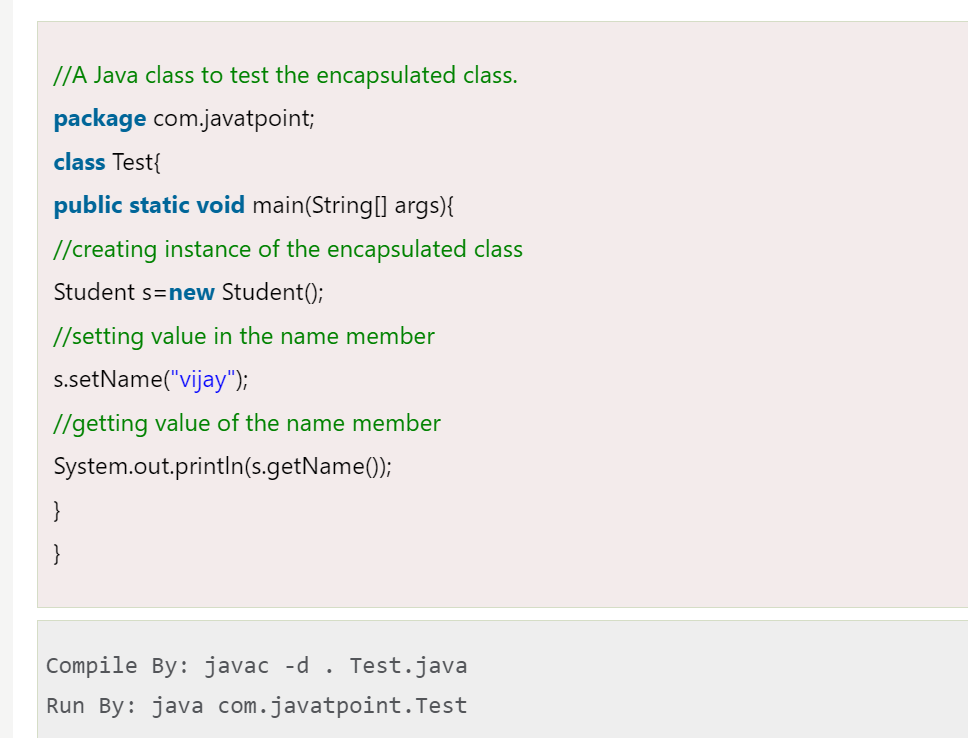
### Advantage of Encapsulation in Java

* By providing only a setter or getter method, you can make the class **read-only or write-only**. In other words, you can skip the getter or setter methods.
* It provides you the **control over the data**. Suppose you want to set the value of id which should be greater than 100 only, you can write the logic inside the setter method. You can write the logic not to store the negative numbers in the setter methods.
* It is a way to achieve **data hiding** in Java because other class will not be able to access the data through the private data members.
* The encapsulate class is **easy to test**. So, it is better for unit testing.
* The standard IDE's are providing the facility to generate the getters and setters. So, it is **easy and fast to create an encapsulated class** in Java.

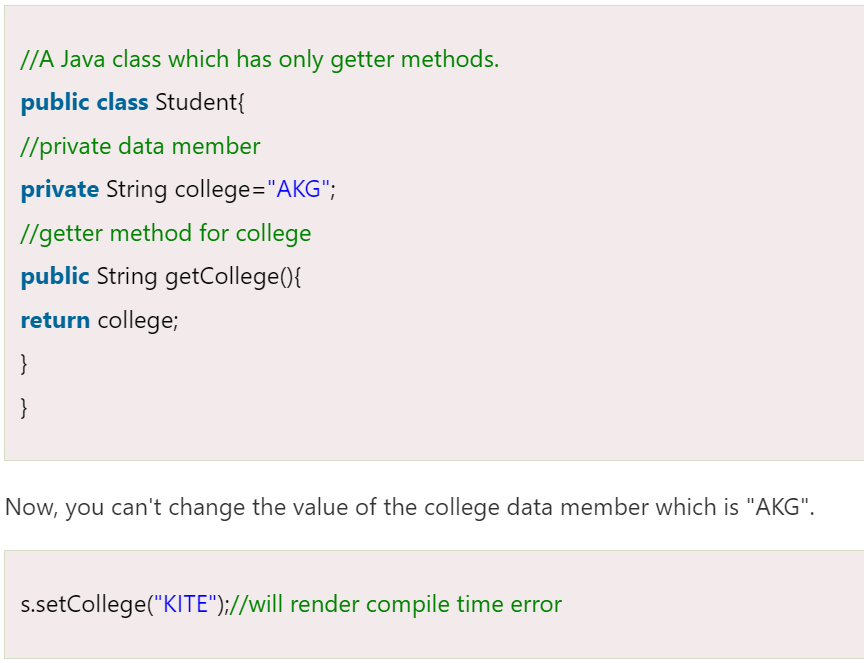
### Simple Example of Encapsulation in Java

Let's see the simple example of encapsulation that has only one field with its setter and getter methods.

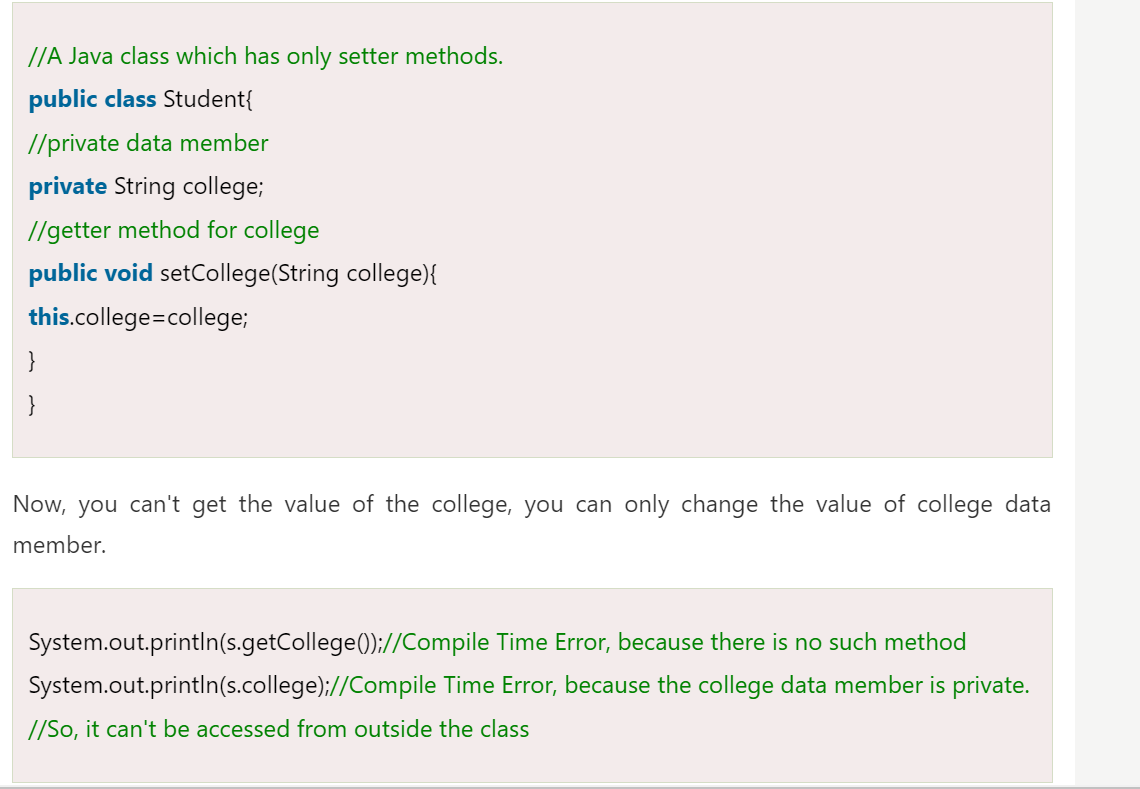




### Read-Only class



### Write-Only class



### Another Example of Encapsulation in Java

