Encapsulation

Encapsulation in Java or object oriented programming language is a concept which enforce protecting variables, functions from outside of class, in order to better manage that piece of code and having least impact or no impact on other parts of program due to change in protected code. *Encapsulation in Java* is visible at different places and Java language itself provides many construct to encapsulate members. You can completely encapsulate a member be it a variable or method in Java by using private keyword and you can even achieve a lesser degree of encapsulation in Java by using other access modifier like protected or public.

True value of encapsulation is realized in an environment which is prone to change a lot and we know that in software requirements changes every day at that time if you have your code well encapsulated you can better manage risk with change in requirement.

i.e maintained in just one place and not scattered around code is easy to change. this can be better explained with a simple example of encapsulation in Java.

we all know that constructor is used to create object in Java and constructor can accept argument. Suppose we have a class Loan has a constructor and than in various classes you have created instance of loan by using this constructor. now requirements change and you need to include age of borrower as well while taking loan. Since this code is not well encapsulated i.e. not confined in one place you need to change everywhere you are calling this constructor i.e. for one change you need to modify several file instead of just one file which is more error prone and tedious, though it can be done with refactoring feature of advanced IDE wouldn't it be better if you only need to make change at one place ? Yes that is possible if we encapsulate Loan creation logic in one method say createLoan() and client code call this method and this method internally crate Loan object. in this case you only need to modify this method instead of all client code.

Example of Encapsulation in Java

```
class Loan{
    private int duration;  //private variables examples of encapsulation
    private String loan;
    private String borrower;
    private String salary;

    //public constructor can break encapsulation instead use factory method
    private Loan(int duration, String loan, String borrower, String salary){
        this.duration = duration;
        this.loan = loan;
        this.borrower = borrower;
        this.salary = salary;
    }

    //no argument consustructor omitted here

   // create loan can encapsulate loan creation logic
    public Loan createLoan(String loanType){

     //processing based on loan type and than returning loan object
      return loan;
    }

}
```

Read more: http://javarevisited.blogspot.com/2012/03/what-is-encapsulation-in-java-and-oops.html#ixzz2RMrtGCZG