**Encapsulation in java** is a process of wrapping code and data together into a single unit, for example capsule i.e. mixed of several medicines. 

The whole idea behind encapsulation is to hide the implementation details from users. If a **data member is private** it means it can only be accessed within the same class. No outside class can access private data member (variable) of other class.

However if we setup **public getter** and **setter** methods to update and read the private data fields then the outside class can access those private data fields via public methods.

This way data can only be accessed by public methods thus making the private fields and their implementation hidden for outside classes. That’s why encapsulation is known as **data hiding.**

**Example in eclipse.**

**Why to use encapsulation.**

**In bank application, users account balance will be a private data member, no any other class / user should be able to access balance of another person. But the account holder can access his account balance only after providing valid credentials to system, So after validation, account holder can access his balance amount. This is where encapsulation comes in picture.**

public class EncapTest {

private String name;

private String idNum;

private int age;

public int getAge() {

return age;

}

public String getName() {

return name;

}

public String getIdNum() {

return idNum;

}

public void setAge( int newAge) {

age = newAge;

}

public void setName(String newName) {

name = newName;

}

public void setIdNum( String newId) {

idNum = newId;

}

}

public class RunEncap {

public static void main(String args[]) {

EncapTest encap = new EncapTest();

encap.setName("James");

encap.setAge(20);

encap.setIdNum("12343ms");

System.out.print("Name : " + encap.getName() + " Age : " + encap.getAge());

}

}