Encapsulation is the process of hiding information details and protecting data and behavior of an object from misuse by other objects. In Java, encapsulation is done using access modifiers (public, protected, private) with classes, interfaces, setters, getters.

In Java, the access modifier private is used to protect data and behaviors from outside. Example Let’s modify the Person class above to prevent the attributes name and age from being modified by other objects:

Class Person {

private String name;

private int age;

}

Here, the fields age and name can be only changed within the Person class. If someone attempts to make a change like this:

Person p = new Person();

p.name = “Tom”; // ERROR!

The code won’t compile because the field name is marked as private.

But what if we want the other objects to be able to read the name and age? In this case, we provide methods whose name in the form of getXXX() - so called **getters** in Java. Hence we add two getters to the Person class:

class Person {

private String name;

private int age;

public String getName() {

return name;

}

public String getAge() {

return age;

}

}