

JAVA - ENCAPSULATION

Encapsulation is one of the four fundamental OOP concepts. The other three are inheritance, polymorphism, and abstraction.

Encapsulation is the technique of making the fields in a class private and providing access to the fields via public methods. If a field is declared private, it cannot be accessed by anyone outside the class, thereby hiding the fields within the class. For this reason, encapsulation is also referred to as data hiding.

Encapsulation can be described as a protective barrier that prevents the code and data being randomly accessed by other code defined outside the class. Access to the data and code is tightly controlled by an interface.

The main benefit of encapsulation is the ability to modify our implemented code without breaking the code of others who use our code. With this feature Encapsulation gives maintainability, flexibility and extensibility to our code.

Example:

Let us look at an example that depicts encapsulation:

```
/* File name : EncapTest.java */
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;
    }
}
```

The public methods are the access points to this class' fields from the outside java world. Normally, these methods are referred as getters and setters. Therefore any class that wants to access the variables should access them through these getters and setters.

The variables of the EncapTest class can be access as below::

```
/* File name : RunEncap.java */
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());
}
}
```

This would produce the following result:

```
Name : James Age : 20
```

Benefits of Encapsulation:

- The fields of a class can be made read-only or write-only.
- A class can have total control over what is stored in its fields.
- The users of a class do not know how the class stores its data. A class can change the data type of a field and users of the class do not need to change any of their code.