

## Encapsulation

The meaning of **Encapsulation** is to make sure that "sensitive" data is hidden from users. In other words the variables of a class will be hidden from other classes, and can be accessed only through the **methods** of their current class. Therefore, it is also known as **data hiding**. To achieve this, you must:

- declare class variables/attributes as **private**
- Provide public **get** and **set** methods to access and update the value of a private variable.

Why Encapsulation?

- Better control of class attributes and methods.
- Class attributes can be made **read-only** (if you only use the **get method**), or **write-only** (if you only use the **set method**)
- **Flexible**: the programmer can change one part of the code without affecting other parts
- Increased security of data.

**Example:** Following is an example that demonstrates how to achieve Encapsulation.

```
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 **setXXX()** and **getXXX()** methods are the access points of the instance variables of the EncapTest class. 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 accessed using the following program –

```

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());
    }
}

```

### **Output**

Name: James Age: 20

**Ex:** create rectangle class with two private instance variables (length and width) and prevent less than 0 entry? what we can do if the class has private method?

**Ex:** write java program to create class time with two private instance variables (hours (0-12), minute (0-59), and seconds (0-59)).

**Ex:** write java program to create class Student and declare variables stdName, stdRollNo, and stdId (10 bits )as private.