

# Encapsulation

## Day 19 Assignment

1. What is Encapsulation in Java? Why is it called Data hiding?
2. What are the important features of Encapsulation?
3. What are getter and setter methods in Java Explain with an example
4. What is the use of this keyword explain with an example
5. What is the advantage of Encapsulation?
6. How to achieve encapsulation in Java? Give an example.

### 1. What is Encapsulation in Java? Why is it called Data hiding?

**Ans:** Encapsulation in Java is the process by which data (variables) and the code that acts upon them (methods) are integrated as a single unit.

Encapsulation = Data Hiding + Data Abstraction

In encapsulation, 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.

### 2. What are the important features of Encapsulation?

**Ans:** Features of Encapsulation are following

- Combine the data of our application and its manipulation at one place.
- A class can have complete control over its data members and data methods.
- The class will maintain its data members and methods as read-only.
- Encapsulation Allows the state of an object to be accessed and modified through behaviours.
- Reduce the coupling of modules and increase the cohesion inside them.

### 3. What are getter and setter methods in Java Explain with an example

**Ans:** Getters and setters are **used to protect your data, particularly when creating classes**. For each instance variable, a getter method returns its value while a setter method sets or updates its value. Given this, getters and setters are also known as accessors and mutators, respectively.

Eg-

```
package com.dataflair.encapsulation;
class TestEncapsulation {
    private String privateVar;
    TestEncapsulation() {
        privateVar = "java";
    }
    public void getVariable() {
        System.out.println(privateVar);
    }
    public void setVariable(String setvalue) {
        privateVar = setvalue;
    }
}
public class PrivateVariables {
    public static void main(String[] args) {
```

```
TestEncapsulation test = new TestEncapsulation();
test.setVariable("PHP");
test.getVariable();
//System.out.println(TestEncapsulation.privateVar);
}
}
```

## 4. What is the use of this keyword explain with an example

**Ans:** This keyword refers to the current object in a method or constructor. The most common use of this keyword is to eliminate the confusion between class attributes and parameters with the same name (because a class attribute is shadowed by a method or constructor parameter).

Use of this keyword are following

- Using 'this' keyword to refer current class instance variables
- Using this() to invoke current class constructor
- Using 'this' keyword to return the current class instance
- Using 'this' keyword as method parameter
- Using 'this' keyword to invoke current class method
- Using 'this' keyword as an argument in the constructor call

Eg-

class Test

```
{
    int a;
    int b;
    // Parameterized constructor
    Test(int a, int b)
    {
        this.a = a;
        this.b = b;
    }
    void display()
    {
        //Displaying value of variables a and b
        System.out.println("a = " + a + " b = " + b);
    }
    public static void main(String[] args)
    {
        Test object = new Test(10, 20);
        object.display();
    }
}
```

## 5. What is the advantage of Encapsulation?

Advantage of Encapsulation in Java are following

- **Protect Your Data**
  - With encapsulation, you can keep your data and codes safe from external inheritance.
  - For example, if any program runner tries to change the program, they can only interact with the getter and setter methods of the program. They will not have any idea to

change any specific variable or data and hinder the running of the program resulting in high security.

- **Easy to Test code**
  - The code which is encapsulated is simple to debug and easy to test for unit testing.
- **Flexible**
  - The encapsulated code is cleaner, flexible, and easy to change as per our needs. It means we can change the code read-only or write-only by getter and setter methods.
  - For example, if you don't define the setter method in the class then the fields can be made read-only whereas if you don't define the getter method in the class then the fields can be made write-only.
- **Easy to Reuse**
  - Encapsulation enables you to easily change the methods, reuse the code, and execute new requirements in your program.

## 6. How to achieve encapsulation in Java? Give an example.

**Ans:** There are two important ways through which you can achieve (or, implement) encapsulation in the Java

1. **By setting the instance variable of the class as private, so that it cannot be used directly by anyone from outside the class.**

Eg-

```
public class TestEncapsulation {
    private String privateVar;
    TestEncapsulation() {
        privateVar = "java";
    }
}

public class PrivateVariables {
    public static void main(String[] args) {
        System.out.println(TestEncapsulation.privateVar);
    }
}
```

2. **Set and get the values of variables/fields in the class using public setter and getter methods.**

Eg-

```
package com.dataflair.encapsulation;
class TestEncapsulation {
    private String privateVar;
    TestEncapsulation() {
        privateVar = "java";
    }
    public void getVariable() {
        System.out.println(privateVar);
    }
    public void setVariable(String setvalue) {
        privateVar = setvalue;
    }
}
```

```
public class PrivateVariables {  
    public static void main(String[] args) {  
        TestEncapsulation test = new TestEncapsulation();  
        test.setVariable("PHP");  
        test.getVariable();  
        //System.out.println(TestEncapsulation.privateVar);  
    }  
}
```