# Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….


We Educate
We Develop

# Encapsulation in Java

## => Encapsulation :-

-> Encapsulation is the process by which variables and methods are wrapped or binded into a single unit

-> Encapsulation is technically hiding the data from other classes and these data can be accessed only through the member functions of its own class

-> Real world example : capsule, mobile

-> Technically every java class is an example of encapsulation

-> Main example of encapsulation in java is "JavaBean" classes

-> Encapsulation is achieved by declaring variables as private and public getter and setter methods

-> For example :

```
class Employee
{
    private int salary;

    public void setSalary(int salary)
    {
        this.salary=salary;
    }
    public int getSalary()
    {
        return salary;
    }
}
```

-> In IDE's (eclipse and netbeans) getter and setter methods can be generated directly so its very easy to generate encapsulation in IDE's

-> Encapsulation = data hiding + abstraction

**-> Advantages of encapsulation :-**

1. Data Hiding
2. Increase flexibility
3. Reusability
4. Testing code is easy

---

**=> JavaBean Class :-**

-> JavaBean class is used to encapsulate the data into single object

-> Fules for JavaBean class :-

1. Must implement Serializable interface
2. Class must contain public no-argument constructor
3. Class must have all private variables

4. Class must contain public getter and setter methods

---

=> Tightly Encapsulated Class :-

-> A class is tightly encapsulated class if and only if it have all the variables as private

---

**=> Interview Questions :-**

1. What is difference between Abstraction & Encapsulation

-> Abstraction : It hides the implementation (details)

Encapsulation : It hides the data (information)

-> Abstraction : It is achieved by "abstract class" and "interface"

Encapsulation : It is achieved by using "JavaBean" class

-> Abstraction : In case of abstraction we have to use "abstract" keyword

Encapsulation : In case of encapsulation we have to use access modifiers (private & public)

-> Abstraction : It resolves the issue at design level

Encapsulation : It resolves the issue at implementation level

## => Which are 6 main pillers in OOP's

1. Classes
2. Objects & Methods
3. Inheritance
4. Polymorphism
5. Abstraction
6. Encapsulation

Smart Programming
We Educate
We Develop

# Company Links & Contacts

**Company Name:** Smart Programming (+91 62838-30308)

**Address :** Chandigarh & Mohali (Punjab), India

**Websites:** https://www.smartprogramming.in/

https://courses.smartprogramming.in

**Android App:**
https://play.google.com/store/apps/details?id=com.smartprogramming

**YouTube Channel:**
https://www.youtube.com/c/SmartProgramming