

Practical 03 - Encapsulation

We have already discussed a about encapsulation while discussing OOPs concepts.

The whole idea behind encapsulation is to hide the implementation details from users. If a data member is private it means it can only be accessed within the same class. No outside class can access private data member (variable) of other class. However if we setup public getter and setter methods to update (for e.g. void setSSN(int ssn))and read (for e.g. int getSSN()) the private data fields then the outside class can access those private data fields via public methods. This way data can only be accessed by public methods, thus making the private fields and their implementation hidden for outside classes. That's why encapsulation is known as data hiding.

```
public class EncapsulationDemo{
    private String empName;

    //Getter and Setter methods

    public String getEmpName(){
        return empName;
    }

    public void setEmpName(String newValue){
        empName = newValue;
    }
}

public class EncapsTest{
    public static void main(String args[]){
        EncapsulationDemo obj = new EncapsulationDemo();
        obj.setEmpName("Mario");
        System.out.println("Employee Name: " + obj.getEmpName());
    }
}
```

Exercise 3-1: Develop a code for the following scenario.

“An encapsulated class contains three variables to store Name, Age and Salary of the employee. develop getters and setters to set and get values.”

```
public class Employee {
    private String Name;
    private int age;
    private double salary;

    public String getName() {
        return Name;
    }
    public int getAge() {
        return age;
    }
}
```

Practical 03 - Encapsulation

```
}  
    public double getSalary() {  
        return salary;  
    }  
  
    public void setName(String Name) {  
        this.Name=Name;  
    }  
    public void setAge(int age) {  
        this.age=age;  
    }  
    public void setSalary(double salary) {  
        this.salary=salary;  
    }  
}
```

Develop a test class to test your code

```
public class test {  
  
    public static void main(String[] args) {  
        Employee Employee =new Employee();  
  
        Employee.setName("lula");  
        Employee.setAge(22);  
        Employee.setSalary(25000);  
  
        System.out.println(Employee.getName());  
        System.out.println(+Employee.getAge());  
        System.out.println(+Employee.getSalary());  
    }  
}
```

Now modify the same code by trying to replace the setters using a constructor.

```
public class Employee {  
    private String Name;  
    private int age;  
    private double salary;
```

Practical 03 - Encapsulation

```
public Employee(String Name , int age, double salary) {
    this.Name=Name;
    this.age=age;
    this.salary=salary;
}

//getter method
public String getName() {
    return Name;
}
public int getAge() {
    return age;
}
public double getSalary() {
    return salary;
}
}
```

```
public class test {

    public static void main(String[] args)
    {
        Employee Employee =new Employee("lula", 22 , 25987);

        System.out.println(Employee.getName());
        System.out.println(+Employee.getAge());
        System.out.println(+Employee.getSalary());
    }
}
```

Exercise 3-2: Code for the last example that we have discussed during the class. We need the following Output. (Use Netbeans code generation option where necessary)

Employee Name: xxxxx (Use setter to set and getter to retrieve)

Basic Salary: xxxxx (Use setter to set and getter to retrieve)

Bonus: xxxxx (You may use the constructor to pass this value)

Bonus Amount: xxxxx (Develop a separate method to calculate Bonus amount. Bonus amount is the total of Bonus and Basic Salary)

Practical 03 - Encapsulation

```
public class Employee {
    private String Name;
    private double bouns;
    private double salary;

    public Employee(String name, double salary , double boune)
    {
        this.Name=name;
        this.salary=salary;
        this.bouns=bouns;
    }

    //getter method
    public void setName(String Name) {
        this.Name=Name;
    }

    public String getName() {
        return Name;
    }

    public void setsalary(double salary) {
        this.salary=salary;
    }

    public double getSalary() {
        return salary;
    }

    public double getBouns() {
        return bouns;
    }
    public double calculateBonusA() {
        return salary+bouns;
    }
}
```

```
public class test {

    public static void main(String[] args)
    {
        Employee Employee =new
Employee("lula",5000.00,1000.00);
    }
}
```

Practical 03 - Encapsulation

```
        System.out.println("Employee Name  
"+Employee.getName());  
        System.out.println("Basic Salary  
"+Employee.getSalary());  
        System.out.println("Bouns "+Employee.getBouns());  
  
        double bounsAmount = Employee.calculateBonusA();  
        System.out.println("Bouns Amount "+bounsAmount);  
    }  
}
```

E.g.

Employee Name: Bogdan
Basic Salary: 50000
Bonus: 10000
Bonus Amount: 60000