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. Evelop getters and setters to set and get values . Develop a test class to test your code."

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

public class Employee

{

private String name;

private int age;

private float salary;

//getter and setter

```
    public String getName()
  {
        return name;
    }
    public void setName(String name)
    {
        this.name = name;
    }
   public int getAge()
    {
        return age;
    }
    public void setAge(int age)
    {
        this.age = age;
    }
    public float getSalary()
    {
        return salary;
    }
    public void setSalary(float salary)
    {
        this.salary = salary;
    }
}
```

Main class

```
public class TestEmployee {
    public static void main(String[] args) {
```

```
        Employee emp = new Employee();
        emp.setName("John Doe");
        emp.setAge(30);
        emp.setSalary(50000.0);
        System.out.println("Employee Name: " + emp.getName());
        System.out.println("Employee Age: " + emp.getAge());
        System.out.println("Employee Salary: " + emp.getSalary());
    }
}
```

Replace the setters using a constructor.

```
public class Employee {
    private String name;
    private int age;
    private float salary;

    public Employee(String name, int age, float salary) {
        this.name = name;
        this.age = age;
        this.salary = salary;
    }
Public void Displaydetails()
{
       System.out.println ("Employee Name: " + emp.Name);
        System.out.println("Employee Age: " + emp.Age);
        System.out.println("Employee Salary: " + emp.Salary);
}
}
```

Main class

```
public class TestEmployee {

   public static void main(String[] args) {

      Employee emp = new Employee("John Doe", 30, 50000.0);

      Emp.Displaydetails();

}

}
```

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: xxxx (Use setter to set and getter to retrieve)

Bonus: xxxx (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)

E.g.

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

```
public class Employee {
   private String name;
   private double basicSalary;
   private double bonus;

   public Employee(String name, double basicSalary, double bonus) {
      this.name = name;
      this.basicSalary = basicSalary;
      this.bonus = bonus;
   }

   public String getName() {
      return name;
```

```
    }

    public void setName(String name) {
        this.name = name;
    }

    public double getBasicSalary() {
        return basicSalary;
    }

    public void setBasicSalary(double basicSalary) {
        this.basicSalary = basicSalary;
    }

    public double getBonus() {
        return bonus;
    }

    public void setBonus(double bonus) {
        this.bonus = bonus;
    }

    public double calculateBonusAmount() {
        return basicSalary + bonus;
    }
}
```

Main class

```
public class EmployeeTest {
    public static void main(String[] args) {
        Employee emp = new Employee("Bogdan", 50000, 10000);

        System.out.println("Employee Name: " + emp.getName());
        System.out.println("Basic Salary: " + emp.getBasicSalary());
        System.out.println("Bonus: " + emp.getBonus());
        System.out.println("Bonus Amount: " + emp.calculateBonusAmount());
    }
}
```