A.J.Pavithra

28658

Java Practical 3

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."

```
// Employee class (Encapsulated class)
public class Employee {
    // Private variables to store Name, Age, and Salary
    private String name;
    private int age;
    private double salary;
    // Getters for Name, Age, and Salary
    public String getName() {
        return name;
    }
    public int getAge() {
        return age;
    }
    public double getSalary() {
        return salary;
    }
    // Setters for Name, Age, and 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;
    }
}
```

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

```
// Employee class (Encapsulated class)
public class Employee {
    // Private variables to store Name, Age, and Salary
    private String name;
    private int age;
    private double salary;
    // Constructor to set values for Name, Age, and Salary during object creation
    public Employee(String name, int age, double salary) {
        this.name = name;
        this.age = age;
        this.salary = salary;
    }
    // Getters for Name, Age, and Salary
    public String getName() {
        return name;
    }
    public int getAge() {
        return age;
    }
    public double getSalary() {
        return salary;
    }
}
```

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

```
// Employee class (Encapsulated class)

public class Employee {

   // Private variables to store Name, Basic Salary, and Bonus

   private String name;

   private double basicSalary;

   private double bonus;


   // Constructor to set Name, Basic Salary, and Bonus during object creation

   public Employee(String name, double basicSalary, double bonus) {

      this.name = name;

      this.basicSalary = basicSalary;

      this.bonus = bonus;

   }
```

```
// Getter for Name
    public String getName() {
        return name;
    }

    // Getter for Basic Salary
    public double getBasicSalary() {
        return basicSalary;
    }

    // Getter for Bonus
    public double getBonus() {
        return bonus;
    }

    // Method to calculate the Bonus Amount (Bonus + Basic Salary)
    public double calculateBonusAmount() {
        return basicSalary + bonus;
    }
}
```

```
// EmployeeTest class to test the code

public class EmployeeTest {

   public static void main(String[] args) {

      // Create an instance of Employee using the constructor

      Employee emp = new Employee("Bogdan", 50000, 10000);


      // Display the Employee details using getters

      System.out.println("Employee Name: " + emp.getName());

      System.out.println("Basic Salary: " + emp.getBasicSalary());

      System.out.println("Bonus: " + emp.getBonus());


      // Calculate and display the Bonus Amount using the method
calculateBonusAmount()

      System.out.println("Bonus Amount: " + emp.calculateBonusAmount());

   }

}
```