CYDEO

Encapsulation

OOP Principles

- There are 4 Object Oriented Programming (OOP) principles:
    - Encapsulation
    - Inheritance
    - Abstraction
    - Polymorphism

- OOP: A design pattern of the applications in an organized and modular way

Encapsulation (Data Hiding)

- An object hides its internal data from code that's outside the class
- Only the current class' methods can directly access and make changes to the instance variables
- Hide an instance variable by giving private access modifier, and making the methods that access those fields public
- These public methods are called getters & setters (accessor and mutator)

Private & Public Access modifiers

Access modifier	Description
private	When the private access modifier is applied to a class member, the member can not be accessed by code outside the class.
public	When the public access modifier is applied to a class member, the member can be accessed by code inside the class or outside.

Getters & Setters

- Both are public instance methods, used to protect our data and make our code more secure
- Getter is used for reading the private data (instance variable) only
- Setter is used for writing (modifying) the private data (instance variable) only

Getters Method

- An instance return method that returns the private instance variable
- Does not pass any parameter
- Return type must match with the data type of the private instance variable

```
public class Person{

    private int age;

    public int getAge(){
        return age;
    }

}
```

Setter Method

- An instance method with the return type of void
- Passes a parameter, and parameter will be assigned to the private instance variable
- Data type of the parameter must match with the data type of the private instance variable

```
public class Person{

    private String name;

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

}
```

Encapsulation Example

- Attributes of Person class objects can only be accessed or modified by getters and setters

```java
public class Person{

    private String name;
    private int age;

    public int getAge(){
        return age;
    }

    public void setAge(int age){
        this.age = age;
    }

    public String getName(){
        return name;
    }

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

}
```

```java
public class Test{

    public static void main(String[] args) {

        Person person1 = new Person();
        person1.setName("Mike");
        person1.setAge(30);

        System.out.println("Name: " + person1.getName());
        System.out.println("Age: " + person1.getAge());

    }
}
```