Access Modifiers

Access Modifiers

- There are 4 access modifiers available in Java:
  - public
  - protected
  - default
  - private
- A top level Java class can have two access modifiers : public and default
- Variables, Constructors and methods can have all four access modifiers

Access Modifiers

Modifier	Class	Package	Subclass	Global
Public	✓	✓	✓	✓
Protected	✓	✓	✓	✗
Default	✓	✓	✗	✗
Private	✓	✗	✗	✗

Cybertek

default

- When no access modifier is specified for a class, method or data member – it is said to be having the default access modifier by default.
- The data members, class or methods which are not declared using any access modifiers are accessible only within the same package.

private

- The private access modifier is specified using the keyword private.
- The methods or data members declared as private are accessible only within the class in which they are declared.
- Any other class of same package will not be able to access these members.
- Top level classes can not be declared as private.

Encapsulation

OOP PRINCIPLES

- Encapsulation
- Inheritance
- Abstraction
- Polymorhism

- Object-oriented: Everything is considered to be an “object” which possess some state, behavior and all the operations are performed using these objects.

Encapsulation(Data Hiding)

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

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.

- 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 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;
    }

}
```

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

    Person p1 = new Person();
    p1.setAge(27);
    p1.setName("Mike");

    System.out.println(p1.getName() + p1.getAge());
}
```

Cybertek

- We can provide only getter in a class to make the variable immutable. (Read only)
- We can provide only setter in a class to make the class attribute write-only.