**Access Specifiers in Java**

In Java, **access specifiers** (also called access modifiers) determine the **visibility** or **scope of a class, constructor, method, or variable.** They define how the members (variables, methods) of a class can be accessed from outside the class or package. Java provides four main types of access specifiers:

1. **Public**
2. **Private**
3. **Protected**
4. **Default** (no keyword)

**1. Public**

* **Syntax**: public
* **Scope**: A public member (class, method, or variable) is accessible from **anywhere** in the program.
* **Usage**: You use the public modifier when you want a method, variable, or class to be accessible across the entire application.

Example:

**public class MyClass {**

**public int myPublicVar = 10;**

**public void show() {**

**System.out.println("Public Method");**

**}**

**}**

In this example, both myPublicVar and show() can be accessed from any class in any package.

**2. Private**

* **Syntax**: private
* **Scope**: A private member is accessible **only within the class** where it is declared. It is not accessible from outside the class, not even from subclasses.
* **Usage**: Use private to hide internal implementation details from other classes.

Example:

**public class MyClass{**

**private int myPrivateVar = 20;**

**private void display() {**

**System.out.println("Private Method");**

**}**

**public void accessPrivate() {**

**System.out.println(myPrivateVar); // Accessible within the class**

**display(); // Accessible within the class**

**}**

**}**

In this example, myPrivateVar and display() can only be accessed within the MyClass class.

**3. Protected**

* **Syntax**: protected
* **Scope**: A protected member is accessible:
  + Within the same package
  + In **subclasses** (even if they are in different packages)
* **Usage**: protected is mainly used in inheritance scenarios to allow subclasses to access members of the parent class.

Example:

**public class Parent {**

**protected int myProtectedVar = 30;**

**protected void showProtected() {**

**System.out.println("Protected Method");**

**}**

**}**

**public class Child extends Parent {**

**public void accessProtected() {**

**System.out.println(myProtectedVar); // Accessible in the subclass**

**showProtected(); // Accessible in the subclass**

**}**

**}**

In this example, the Child class can access the protected members of Parent class even though they are in different classes.

**4. Default (No Specifier)**

* **Syntax**: No keyword (just declare the member without public, private, or protected)
* **Scope**: A member with default access is accessible **only within the same package**. It is often referred to as **package-private** visibility.
* **Usage**: Use default access when you want classes and methods to be accessible only within the same package but hidden from classes outside the package.

Example:

**class MyClass {**

**int myDefaultVar = 40; // Default access**

**void showDefault() {**

**System.out.println("Default Access Method");**

**}**

**}**

In this example, myDefaultVar and showDefault() can only be accessed from within the same package.

**Access Specifier Table**

| **Access Level** | **Class** | **Package** | **Subclass** | **Outside** |
| --- | --- | --- | --- | --- |
| **public** | Yes | Yes | Yes | Yes |
| **protected** | Yes | Yes | Yes | No |
| **default** | Yes | Yes | No | No |
| **private** | Yes | No | No | No |

**Conclusion**

* **Public**: Use when the member needs to be accessed globally.
* **Private**: Use when you want to hide details and prevent access from outside the class.
* **Protected**: Use when you want subclasses to access the member.
* **Default**: Use when access is needed only within the same package.

**1. If Statements**

* Write a program that checks if a number is positive, negative, or zero.
* Given a student's score, print whether they passed (score ≥ 50) or failed.
* **2. If-Else Statements**
* Write a program that checks if a number is even or odd.
* Given two numbers, find the largest of the two.
* Create a program that prints "teenager" if age is between 13 and 19, and "not a teenager" otherwise.

**3. For and While Loops**

* Write a program that prints numbers from 1 to 10 using a for loop.
* Use a while loop to print the first 10 multiples of 5.
* Write a program that calculates the factorial of a given number using both for and while loops.

**4. Class and Object**

* Create a class Car with attributes like model, year, and methods like start(). Create an object of this class and call its methods.
* Design a class Student with attributes name, age, and grade. Write a method to display the student's information.
* Create a class Book with methods to get the title and author. Instantiate the class and display the book's details.

**5. Inheritance**

* Create a base class Animal with a method makeSound(). Inherit this class into Dog and Cat classes, and override the makeSound() method for each subclass.
* Write a program that demonstrates multilevel inheritance. Create a class Person, inherit it into a class Employee, and then into a class Manager.
* Design a class hierarchy where Vehicle is the base class, and Car and Bike are derived classes. Add methods to each class and demonstrate method overriding.

**6. Access Specifiers**

* Create a class with private, protected, and public members. Write another class to show how these members are accessed.
* Create a program where two classes are in different packages. Use protected members and demonstrate how they are accessible from subclasses.
* Demonstrate the difference between public and private access specifiers by creating a class with both types of methods and attributes, and show how they can or cannot be accessed from outside the class.