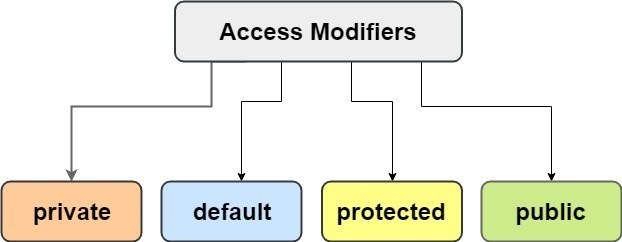
Access modifiers

Access modifiers in java specify the scope of a class, constructor, variable , method or data member. There are four types of access modifiers available in java:

1. **Private**
2. **Default –**No keyword required
3. **Protected**
4. **Public**



##### **1. Private Access Modifier –**

The private access modifier is accessible only within class. 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.
* Classes or interface can not be declared as private.

In this example, we will create two classes A and B within same package p1. We will declare a method in class A as private and try to access this method from class B and see the result.

|  |  |
| --- | --- |
|  | //Java program to illustrate error while using class from different package with  //private modifier  package p1;    class A  {     private void display()      {          System.out.println("Simple Snippets");      }  }    class B  {     public static void main(String args[])        {            A obj = new A();            //trying to access private method of another class            obj.display();        }  } |

**Output**

error: display() has private access in A  
obj.display();

##### **2. Default Access Modifier –**

If you don’t use any modifier, it is treated as **default** bydefault. The default modifier is accessible only within package.

* The data members, class or methods which are not declared using any access modifiers i.e. having default access modifier are accessible **only within the same package**.

In this example, we will create two packages and the classes in the packages will be having the default access modifiers and we will try to access a class from one package from a class of second package.

//Java program to illustrate default modifier

package p1;

//Class MyClass1 is having Default access modifier

class MyClass1

{

void display()

{

System.out.println("Hello World!");

}

}

Run on IDE

//Java program to illustrate error while

//using class from different package with

//default modifier

package p2;

import p1.\*;

//This class is having default access modifier

class MyClass2

{

public static void main(String args[])

{

//accessing class MyClass1 from package p1

MyClass1 obj = new MyClass1();

obj.display();

}

}

**Output**

Compile time error

##### **3. Protected Access Modifier –**

The protected access modifier is specified using the keyword **protected**.

* The **protected access modifier** is accessible within package and outside the package but through inheritance only.
* The protected access modifier can be applied on the data member, method and constructor.
* It can’t be applied on the class.

In this example, we will create two packages p1 and p2. Class A in p1 is made public, to access it in p2. The method display in class A is protected and class B is inherited from class A and this protected method is then accessed by creating an object of class B.

//Java program to illustrate

//protected modifier

package p1;

//Class A

public class A

{

protected void display()

{

System.out.println("Simple Snippets");

}

}

Run on IDE

//Java program to illustrate

//protected modifier

package p2;

import p1.\*; //importing all classes in package p1

//Class B is subclass of A

class B extends A

{

public static void main(String args[])

{

B obj = new B();

obj.display();

}

}

**Output**

Simple Snippets

##### **4. Public Access Modifier –**

The public access modifier is specified using the keyword **public**.

* The **public access modifier** is accessible everywhere. It has the widest scope among all other modifiers.
* Classes, methods or data members which are declared as public are **accessible from every where** in the program. There is no restriction on the scope of a public data members.

//Java program to illustrate

//public modifier

package p1;

public class A

{

public void display()

{

System.out.println("Simple Snippets");

}

}

package p2;

import p1.\*;

class B

{

public static void main(String args[])

{

A obj = new A;

obj.display();

}

}

* **Output**

Simple Snippets

Understanding all Access Modifiers Accessibility using Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Access Modifier** | **within class** | **within package** | **outside package by subclass only** | **outside package** |
| **Private** | Y | N | N | N |
| **Default** | Y | Y | N | N |
| **Protected** | Y | Y | Y | N |
| **Public** | Y | Y | Y | Y |