**Java Access Modifiers**

An access modifier restricts the access of a class, constructor, data member and method in another class. In java we have four access modifiers:

1. default

2. private

3. protected

4. public

**Non-Access modifier**

There are five non-access modifiers available in Java, used to achieve many other functionalities.

* Modifier to finalizing the implementations of classes, methods, and variables (final)
* Modifier to create class methods and variables (static)
* Modifier for creating abstract classes and methods (abstract)
* Modifiers for use in threads (synchronized and volatile)

**1. Default access modifier**

When we do not mention any access modifier, it is called default access modifier.

The scope of this modifier is limited to the package only. This means that if we have a class with the default access modifier in a package, only those classes that are in this package can access this class.

No other class outside this package can access this class.

Similarly, if we have a default method or data member in a class, it would not be visible in the class of another package.

**Default Access Modifier Example in Java**

In this example we have two classes, Test class is trying to access the default method of Addition class, since class Test belongs to a different package, this program would throw compilation error, because the scope of default modifier is limited to the same package in which it is declared.

**//Addition.java**

package abcpackage;

public class Addition {

/\* Since we didn't mention any access modifier here, it would

\* be considered as default.

\*/

int addTwoNumbers(int a, int b){

return a+b;

}

}

Test.java

import abcpackage.\*;

public class Test {

public static void main(String args[]){

Addition obj = new Addition();

/\* It will throw error because we are trying to access

\* the default method in another package

\*/

obj.addTwoNumbers(10, 21);

}

}

Output:

Exception in thread "main" java.lang.Error: Unresolved compilation problem:

The method addTwoNumbers(int, int) from the type Addition is not visible

at xyzpackage.Test.main(Test.java:12)

**2. Private access modifier**

* The scope of private modifier is limited to the class only.
* Private Data members and methods are only accessible within the class
* Class and Interface cannot be declared as private
* If a class has private constructor then you cannot create the object of that class from outside of the class.

**Private access modifier example in java**

This example throws compilation error because we are trying to access the private data member and method of class ABC in the class Example. The private data member and method are only accessible within the class.

class ABC{

private double num = 100;

private int square(int a){

return a\*a;

}

}

public class Example{

public static void main(String args[]){

ABC obj = new ABC();

System.out.println(obj.num);

System.out.println(obj.square(10));

}

}

Output:

Compile - time error

**3. Protected Access Modifier**

Protected data member and method are only accessible by the classes of the same package and the subclasses present in any package. You can also say that the protected access modifier is similar to default access modifier with one exception that it has visibility in sub classes.

Classes cannot be declared protected. This access modifier is generally used in a parent child relationship.

**Protected access modifier example in Java**

In this example the class Test which is present in another package is able to call the addTwoNumbers() method, which is declared protected. This is because the Test class extends class Addition and the protected modifier allows the access of protected members in subclasses (in any packages).

Addition.java

package abcpackage;

public class Addition {

protected int addTwoNumbers(int a, int b){

return a+b;

}

}

Test.java

package xyzpackage;

import abcpackage.\*;

class Test extends Addition{

public static void main(String args[]){

Test obj = new Test();

System.out.println(obj.addTwoNumbers(11, 22));

}

}

Output:

33

**4. Public access modifier**

The members, methods and classes that are declared public can be accessed from anywhere. This modifier doesn’t put any restriction on the access.

**public access modifier example in java**

Lets take the same example that we have seen above but this time the method addTwoNumbers() has public modifier and class Test is able to access this method without even extending the Addition class. This is because public modifier has visibility everywhere.

Addition.java

package abcpackage;

public class Addition {

public int addTwoNumbers(int a, int b){

return a+b;

}

}

Test.java

package xyzpackage;

import abcpackage.\*;

class Test{

public static void main(String args[]){

Addition obj = new Addition();

System.out.println(obj.addTwoNumbers(100, 1));

}

}

Output: 101

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 |