Access Modifiers in Java

Access Modifiers in Java – In this Java Tutorial, we shall learn how modifiers are helpful in controlling the access to a class or members of a class.

There are two types in access modifiers:

- **Class Level Modifiers** – Access Modifiers that control access to a class.
  - public
  - default A diagrammatic representation is provided below. We shall see each of the modifiers in detail in this Java Tutorial.

- **Class-Member Level Modifiers** – Access Modifiers that control access to the members of a class, namely methods/routines/behaviors, variables.
  - public
  - private
  - protected
  - default

### Class Level Modifiers

The syntax to specify a modifier for Class is as shown below.

```java
modifier class ClassName{ ... }
```

<table>
<thead>
<tr>
<th>modifier</th>
<th>could be public / default</th>
</tr>
</thead>
<tbody>
<tr>
<td>class</td>
<td>is the standard keyword to define a class</td>
</tr>
<tr>
<td>ClassName</td>
<td>is the name of the class</td>
</tr>
</tbody>
</table>

**Class Level Modifier – public**

The modifier, public, makes the class visible/accessible to the classes everywhere. A sample code snippet to make a class, which is accessible to public, is shown below.

```java
SampleClass.java
public class SampleClass{
    ...
}
```

**Class Level Modifier – default**

By default, if no modifier is given to a class, its visible only to the classes present in same package as that of the class. A sample code snippet to make a class, which is accessible by default only to classes in the same path as of itself, is shown below.
Diagrammatic representation of Class Modifiers in Java

The following diagram depicts the public, default accessibility of SampleClass.java.

- If class definition of SampleClass.java is `public class SampleClass{}`, then any of the classes ClassA, ClassB, ClassC, ClassD or any other class could access SampleClass.
- If class definition of SampleClass.java is `class SampleClass{}`, then only ClassA and ClassB could access SampleClass.

Class Member Level Modifiers:

Going forward, we refer variables and methods of a Class to Class Members.

Syntax

Syntax for definition of a variable with access modifier is shown below:

```
access_modifier data_type variable_name [= initial value];
```

Example: `public int i = 10;
```

Syntax for definition of a method with access modifier is shown below:

```
access_modifier return_type method_name(arguements){ … }
```

Example: `public int getPrimeNumber(int nth){..};`

Overview of Class Member Level Access Modifiers

Following table gives an overview of Accessibility to different scopes for Class Member Level Access Modifiers.
<table>
<thead>
<tr>
<th>Access to</th>
<th>public</th>
<th>protected</th>
<th>default</th>
<th>private</th>
</tr>
</thead>
<tbody>
<tr>
<td>Other members of the same class</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Classes in the same package</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Its Sub-classes</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Classes in other packages</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
</tbody>
</table>

**Class Member Modifier – public:**

If a class member is declared public, then the class member is accessible to:

- Other members of the same class ☐
- Classes in the same package ☐
- Its Sub-classes ☐
- Classes in other packages ☐

The following example java program shows how to make class members public.

```java
public class AccessModifiers {

    // variables that are public
    public int age = 2;
    public String name = "TutorialKart.com";

    /**
     * This method is declared public.
     * @return Persons information
     */
    public String getEntityInfo() {
        return name + " is " + age + " years old.";
    }

    public static void main(String[] args) {
        System.out.println(new AccessModifiers().getEntityInfo());
    }
}
```

**Class Member Modifier – protected:**

If a class member is declared protected, then the class member is accessible to:

- Other members of the same class ☐
- Classes in the same package ☐
- Its Sub-classes ☐
- Classes in other packages ☐

The following example java program shows how to make class members protected.

```java
AccessModifiers.java
```
public class AccessModifiers {

    // variables that are protected
    protected int age = 2;
    protected String name = "TutorialKart.com";

    /**
     * This method is declared protected.
     * @return Persons information
     */
    protected String getEntityInfo(){
        return name+" is "+age+" years old."
    }

    public static void main(String[] args) {
        System.out.println(new AccessModifiers().getEntityInfo());
    }
}

Class Member Modifier –default:

If no modifier is specified, default access is given to the class member. If a class member is declared default, then the class member is accessible to:

- Other members of the same class
- Classes in the same package
- Its Sub-classes
- Classes in other packages

The following example Java program shows how to give default access to class members.

public class AccessModifiers {

    // variables that have no access modifier
    int age = 2;
    String name = "TutorialKart.com";

    /**
     * This method is declared with default access modifier.
     * @return Persons information
     */
    String getEntityInfo(){
        return name+" is "+age+" years old."
    }

    public static void main(String[] args) {
        System.out.println(new AccessModifiers().getEntityInfo());
    }
}

Class Member Modifier –private:

If a class member is declared public, then the class member is accessible to:

- Other members of the same class
- Classes in the same package
- Classes in other packages
The following example java program shows how to give private access to class members.

```java
/**
 * Access Modifiers of Class and its members in Java
 */
public class AccessModifiers {

    // variables that are private
    private int age = 2;
    private String name = "TutorialKart.com";

    /**
     * This method is declared private.
     * @return Persons information
     */
    private String getEntityInfo()
    {
        return name + " is " + age + " years old.";
    }

    public static void main(String[] args) {
        System.out.println(new AccessModifiers().getEntityInfo());
    }
}
```

**Conclusion:**

In this Java tutorial, we have learned **Access Modifiers in Java**. In our next tutorial, we shall learn **Arithmetic Operators** and **Relational Operators** in Java programming language.
Java Programming Concepts
- Java - Data Types
- Java - Variable Types
- Java - Access Modifiers
- Java - Arithmetic Operators
- Java - Relational Operators
- Java - Decision Making
- Java Loops : While, Do While, For
- Java forEach

Java Exception Handling
- Java - Call Stack
- Java - Try Catch
- Java - ArithmeticException
- Java - ArrayIndexOutOfBoundsException
- Java - ArrayStoreException
- Java - NullPointerException
- Java - NumberFormatException
- Java - StringIndexOutOfBoundsException

Java Design Pattern
- Java - Singleton Pattern

Java - Working with Files
- Java - List files and sub-folders in a folder
- Java - Read File contents line by line using Stream
- Java - Read File contents line by line using BufferedReader

Java String Operations
- Java - String
- Java - String Operations
- Java - Print String to Console
- Java - Read Input from Console to String
- Java - Concatenate two Strings
<table>
<thead>
<tr>
<th>Java - Check if Strings are Equal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java - Find Index of First Occurrence of Sub-string</td>
</tr>
<tr>
<td>Java - Find Index of Nth Occurrence of Sub-string</td>
</tr>
<tr>
<td>Java - Replace First Occurrence of Sub-string</td>
</tr>
<tr>
<td>Java - Replace All Occurrences of Sub-string</td>
</tr>
<tr>
<td>Java - Reverse a String</td>
</tr>
</tbody>
</table>

**Java ArrayList**

<table>
<thead>
<tr>
<th>Java - Print all Elements of ArrayList</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java - Delete Nth Element of ArrayList</td>
</tr>
<tr>
<td>Java ArrayList - Insert Element at Specific Position</td>
</tr>
<tr>
<td>Conver Java String Array to ArrayList</td>
</tr>
</tbody>
</table>

**Java Date**

<table>
<thead>
<tr>
<th>Java - Date and Time</th>
</tr>
</thead>
</table>

**Java MySQL**

<table>
<thead>
<tr>
<th>Java JDBC - List all databases in MySQL Server</th>
</tr>
</thead>
</table>