Access specifiers

Access level modifiers determine whether other classes can use a particular field or invoke a particular method. In java, there are two levels of access controls.

* **At top level:** public or package-private(where user mentions no specifier i.e. no explicit specifier mentioned)
* **At member level:** public, private, protected or package-private(no explicit specifier mentioned)

A class may be declared with the modifier public, in which case that class is visible to all classes everywhere. If a class has no modifier (the default, also known as *package-private*), it is visible only within its own package (packages are named groups of related classes). Now, this is the main difference between a public class and a class with no specifier. But as you can see, the difference comes into scenario when the class is part of a package. So, in case of normal use, there is no problem between declaring a class as public and declaring a class with no access specifier.

**At the member level:**

* **Private members:** A privatemember of a class cannot be accessed directly via the objects of the class. It can only be accessed by a public member function of the class. Also, private members cannot be inherited.
* **Protected members:** A protected member of a class can be inherited. But it cannot be accessed directly via the objects of base class (superclass) as well as objects of derived class (subclass). To access a protected member of a class, public functions are needed of the base class as well as derived class.
* **Public members:** A public member of a class can be inherited as well as it can directly accessed by the objects of base class and derived class.
* **No specifier:** When a member of a class has no specifier, it literally behaves as a private member of a class. (This is when the package concept does not come into scenario)