Package:

Packages lets us organize our classes by functionality or relationships.

We can define a package by just including **package** command as the first statement in Java source file. If we omit the package keyword, all the classes. Will be put in The default package, which has no name.

- package mypackage; (here mypackage is the name of package)
- package pkg1[.pkg2[.pkg3]] – e.g., package a.b.c; We just have to specify a period Between each package name in order To create a hierarchy.

Packages and Member Access:

- Packages act as container for classes and other packages. Whereas classes act as container for data and code. The class is Java smallest unit of abstraction.
- Java addresses 4 categories of visibility for class members.
    - Subclass in the same package.
    - Subclass in the different package.
    - Non subclasses in the different package.
    - Classes that are neither in the same package nor subclasses.

	Private	No Modifier	Protected	Public
Same class	Yes	Yes	Yes	Yes
Same package subclass	No	Yes	Yes	Yes
Same package non-subclass	No	Yes	Yes	Yes
Different package subclass	No	No	Yes	Yes
Different package non-subclass	No	No	No	Yes

Java has four access modifiers and they are public, private, protected and no modifier.

- As a thumb rule, anything declared as public can be accessed from different classes and different packages.
- Whereas anything declared as private cannot be accessed outside its class.
- In cases where no access specifier is explicitly mentioned, these can be accessed by Other classes and subclasses in the same package, but not from a different package. This is no modifier.
- If you want your elements to be seen outside your current package, but only to those classes that subclass your class directly, then we need to use protected access modifier.