# **Packages in Java**

A package as the name suggests is a pack(group) of classes, interfaces and other packages. In java we use packages to organize our classes and interfaces. We have two **types of packages in Java**: built-in packages and the packages we can create (also known as user defined package).

Built In Package Example:

Import java.util.Scanner

Here java is the top level package, util is the sub-level package and Scanner is a class which is present in the sub-package util.

Advantages of Using a Package in Java:

* Reusability: While developing a project in java, we often feel that there are few things that we are writing again and again in our code. Using packages, you can create such things in form of classes inside a package and whenever you need to perform that same task, just import that package and use the class.
* Better Organization: Again, in large java projects where we have several hundreds of classes, it is always required to group the similar types of classes in a meaningful package name so that you can organize your project better and when you need something you can quickly locate it and use it, which improves the efficiency.
* Name Conflicts: We can define two classes with the same name in different packages so to avoid name collision, we can use packages

Creating a package:

Example:

Class Addition:

**package** myproject.calculations;

**public** **class** Addition {

**public** **int** add(**int** a, **int** b) {

**return** (a+b);

}

}

Class Substraction:

**package** myproject.calculations;

**public** **class** Substraction {

**public** **int** substract(**int** a, **int** b) {

**return** (a-b);

}

}

Class add:

**package** FPPackage;

**import** myproject.calculations.\*;

**//import** myproject.calculations.Addition;

**public** **class** add {

**public** **static** **void** main(String[] args) {

Addition a1 = **new** Addition();

**int** a = a1.add(3, 4);

System.***out***.println(a);

}

}

**Using a Fully Qualified Name instead of importing a package:**

Example:

**package** FPPackage;

//import myproject.calculations.\*;

**public** **class** add {

**public** **static** **void** main(String[] args) {

myproject.calculations.Addition a1 = **new** myproject.calculations.Addition();

**int** a = a1.add(3, 4);

System.***out***.println(a);

}

}

1. When do you use a fully qualified name instead of importing a package?

Sometimes class name conflict may occur. For example: Lets say we have two packages **abcpackage** and **xyzpackage** and both the packages have a class with the same name, let it be JavaExample.java. Now suppose a class import both these packages like this:

import abcpackage.\*;

import xyzpackage.\*;

This will throw compilation error. To avoid such errors you need to use the fully qualified name method that I have shown above. For example

abcpackage.JavaExample obj = new abcpackage.JavaExample();

xyzpackage.JavaExample obj2 = new xyzpackage.JavaExample();

1. If we create a class inside a package while importing another package then the package declaration should be the first statement, followed by package import. For example:

package abcpackage;

import xyzpackage.\*;

1. A class can have only one package declaration but it can have more than one package import statements. For example:

package abcpackage; //This should be one

import xyzpackage;

import anotherpackage;

import anything;

1. The wild card import like package.\* should be used carefully when working with subpackages. For example: Lets say: we have a package **abc** and inside that package we have another package **foo**, now **foo** is a subpackage.

classes inside abc are: Example1, Example 2, Example 3  
classes inside foo are: Demo1, Demo2

So if I import the package **abc** using wildcard like this:

import abc.\*;

Then it will only import classes Example1, Example2 and Example3 but it will not import the classes of sub package.

To import the classes of subpackage you need to import like this:

import abc.foo.\*;

This will import Demo1 and Demo2 but it will not import the Example1, Example2 and Example3.

So to import all the classes present in package and subpackage, we need to use two import statements like this:

import abc.\*;

import abc.foo.\*;