**PACKAGES**

* It is repeatedly said that one of the main features of the [Object Oriented Programming](http://www.w3schools.in/java-tutorial/object-oriented-programming-oops/) is the ability to reuse the code already written by the programmer.
* One way of achieving is by extending class and using the interface.
* It has a limitation.
* What will you have to do if you have to use a class from another program without physically copying them into the program at the time of development? So another way of achieving the concept of reusability in Java is via the use of Packages.

**WHAT ARE PACKAGES?**

Packages in Java are groups of similar types of classes, interface and sub packages. It is a way of grouping a variety of classes or interfaces collectively. The grouping is usually done according to functionality. The Java packages act as containers for Java classes.

There is also a term named sub-packages. Package inside the package is called the sub-package. It should be created to categorize the package further.

**Here's are the benefits of organizing classes:**

1. The classes contained in the packages of another program can be easily reused.
2. Packages also allow programmers to separate design from coding.
3. In packages, classes can be declared uniquely compared with classes in other packages.
4. Java Packages provide a way to 'hide' classes thus preventing other programs or packages from accessing classes that are meant for internal use only.
5. Packages provide access protection.
6. Java package removes naming collision.

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 Packages:**

import java.util.Scanner

Here:  
→ **java** is a top level package  
→ **util** is a sub package  
→ and **Scanner** is a class which is present in the sub package **util**.

**User- Defined Package**

**The package keyword is used to create a package in java.**

package mypack;

public class Simple{

public static void main(String args[]){

System.out.println("Welcome to package");

}

}

**ACCESSING PACKAGE**

There are three ways to access the package from outside the package:

* import package.\*;
* import package.classname;
* fully qualified name.

**1) Using packagename.\***

* If you use package.\* then all the classes and interfaces of this package will be accessible but not subpackages.
* The import keyword is used to make the classes and interface of another package accessible to the current package.

Example of package that import the packagename.\*

//save by A.java

package pack;

public class A{

public void msg(){System.out.println("Hello");}

}

//save by B.java

package mypack;

import pack.\*;

class B{

public static void main(String args[]){

A obj = new A();

obj.msg();

}

}

Output:Hello

**2) Using packagename.classname**

If you import package.classname then only declared class of this package will be accessible.

Example of package by import package.classname

//save by A.java

package pack;

public class A{

public void msg(){System.out.println("Hello");}

}

//save by B.java

package mypack;

import pack.A;

class B{

public static void main(String args[]){

A obj = new A();

obj.msg();

}

}

Output:Hello

**3) Using fully qualified name**

If you use fully qualified name then only declared class of this package will be accessible. Now there is no need to import. But you need to use fully qualified name every time when you are accessing the class or interface.

It is generally used when two packages have same class name e.g. java.util and java.sql packages contain Date class.

Example of package by import fully qualified name

//save by A.java

package pack;

public class A{

public void msg(){System.out.println("Hello");}

}

//save by B.java

package mypack;

class B{

public static void main(String args[]){

pack.A obj = new pack.A();//using fully qualified name

obj.msg();

}

}

Output:Hello

**Sub packages in Java**

A package inside another package is known as sub package.