Package in Java

A **java package** is a group of similar types of classes, interfaces and sub-packages.

- Built-in Packages (packages from the Java API)
- User-defined Packages (create your own packages)

The Java API is a library of prewritten classes, that are free to use, included in the Java Development Environment.

The library contains components for managing input, database programming, and much much more. The complete list can be found at Oracles website: https://docs.oracle.com/javase/8/docs/api/.

The library is divided into **packages** and **classes**. Meaning you can either import a single class (along with its methods and attributes), or a whole package that contain all the classes that belong to the specified package.

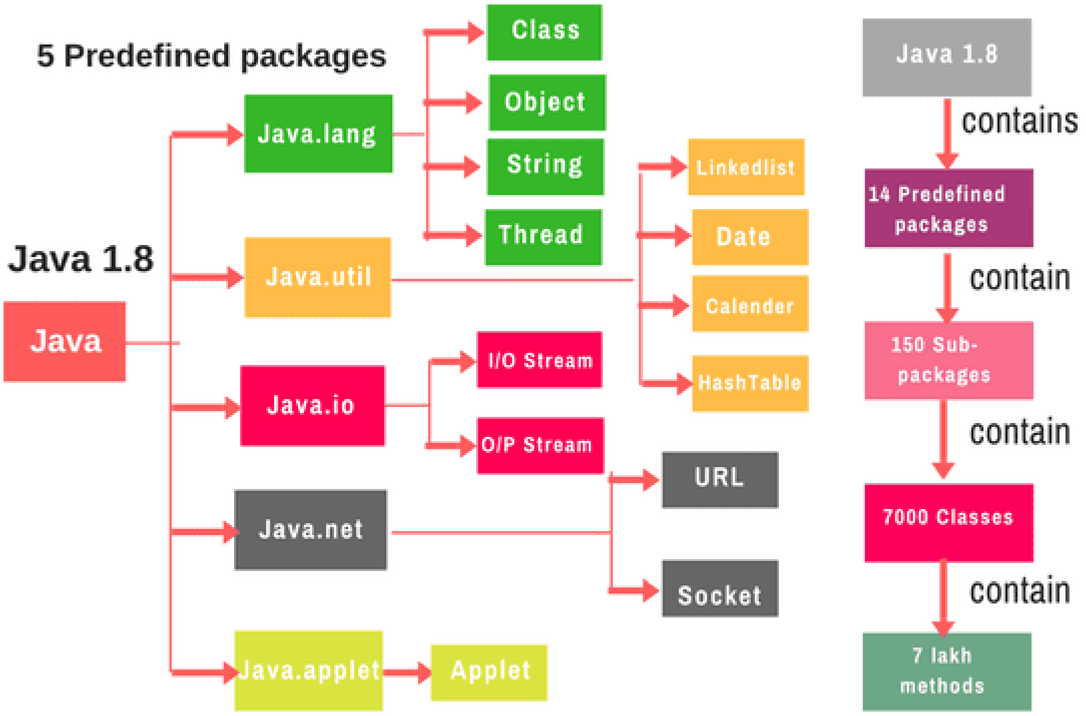
To use a class or a package from the library, you need to use the `import` keyword:

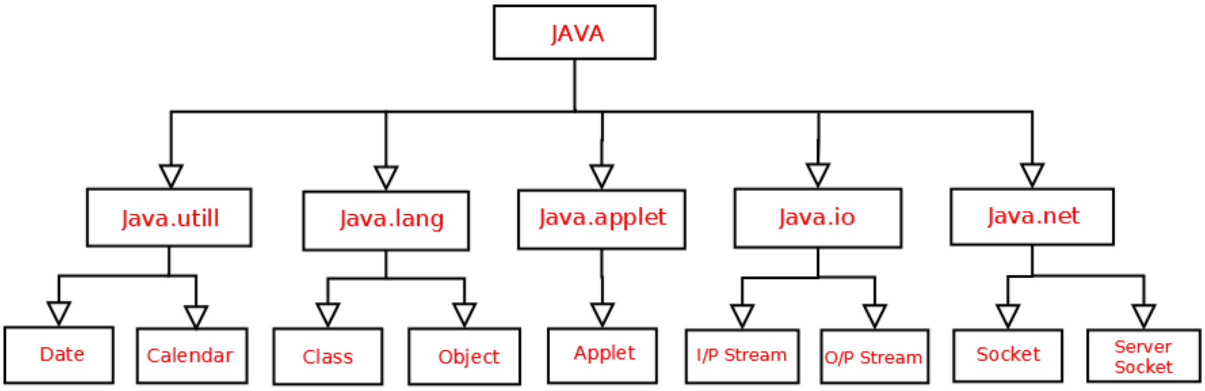
Advantage of Java Package

1) Java package is used to categorize the classes and interfaces so that they can be easily maintained.

2) Java package provides access protection.

3) Java package removes naming collision.





How to access package from another package?

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

1. import package.*;
2. import package.classname;
3. fully qualified name.

Package Naming Convention

- Package name must be in lower case that avoids conflict with the name of classes and interfaces.
- Organizations used their internet domain name to define their package names. For example, **com.debasish.mypackage**. Sometimes, the organization also uses the region after the company name to name the package. For example, **com.debasish.region.mypackage**.
- We use underscore in the package name if the domain name contains hyphen or other special characters or package names begin with a digit or reserved keyword.

Domain Name	**Package Name Prefix**
Hyphenated-name.example.org	org.example.hyphenated_name
Example.int	int_.example
123name.example.com	com.example._123name