Packages:

* Packages are similar to folders in your computer holding related files, making them easier to find, manage, and protect.
* Java Packages group together related classes, interfaces, and sub-packages

## **Characteristics of Java Packages**

Packages in Java are characterized by their ability to prevent naming conflicts and control access to classes and interfaces. They provide a layered structure for your applications, which promotes code modularity and reusability. Effective use of the packages helps to design a well-structured, efficient software system.

Important characteristics of Java packages are:

**1. Code Organization**

Java packages are used for code organization. They bundle related classes and interfaces into packages.

**2. Namespace Management**

Each package in Java creates a new namespace, which helps avoid name conflicts. Two classes in different packages can share the same name without conflict because the package name forms part of the class's full name.

**3. Access Control**

By using access modifiers like public, private, protected, and default, you can control the visibility and accessibility of your classes, interfaces, and their members.

**4. Code Reusability**

Since related classes and interfaces are organized into packages, they can be reused easily across different parts of an application or even across dissimilar applications.

**5. Ease of Distribution**

Packages help in bundling related classes and interfaces together into a single distributable unit (like a JAR file), which is useful while dispensing your application.

**6. Built-In and User-Defined**

Java comes with many built-in packages like java.util, java.io, java.math, etc., which provide a vast array of ready-to-use classes and interfaces. However, you can also define your own packages to suit the requirements of your application.

**7. Hierarchical Structure**

Packages in Java are hierarchical, meaning one can contain other packages, forming a directory-like structure. This structure is reflected in their naming convention.

The **package** ‘keyword’ is used for creating a new package in Java. It should be the first line of code in your Java file. Here's the general syntax:

**package** packageName;

## **Naming Convention for Java Packages**

Java has some widely accepted rules for naming packages to ensure consistency and avoid conflicts:

**1. Lowercase letters**

Package names should be in lowercase to avoid conflict with class names and interfaces.

**2. Unique names**

To avoid conflicts, use unique package names. A common practice is to use the reverse of your domain name, as these are distinctive.

**3. No keywords**

Java keywords should not be used as package names.

For example, if your website is ‘www.mywebsite.com’, a good package name could be ‘com.mywebsite.mypackage’.

In an e-commerce application, you can have different modules such as user management, product management, and order management. Each of these can be represented as a package:

**// User Management**

**package com.ecommerce.user;**

**public class User {**

**// User-related code**

**}**

**// Product Management**

**package com.ecommerce.product;**

**public class Product {**

**// Product-related code**

**}**

**// Order Management**

**package com.ecommerce.order;**

**public class Order {**

**// Order-related code**

**}**