# UNIT-IV

# Inheritance, Packages & Interfaces

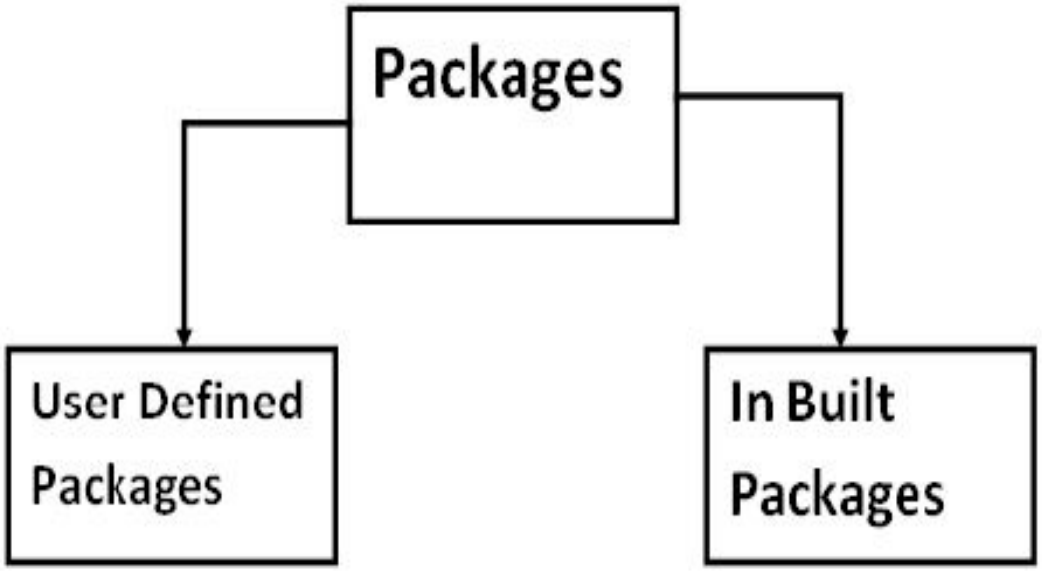
# Package

- Packages are used in Java in order to **prevent naming conflicts**, to **control access**, to make **searching/locating and usage of classes**, **interfaces, enumerations and annotations easier, etc.**

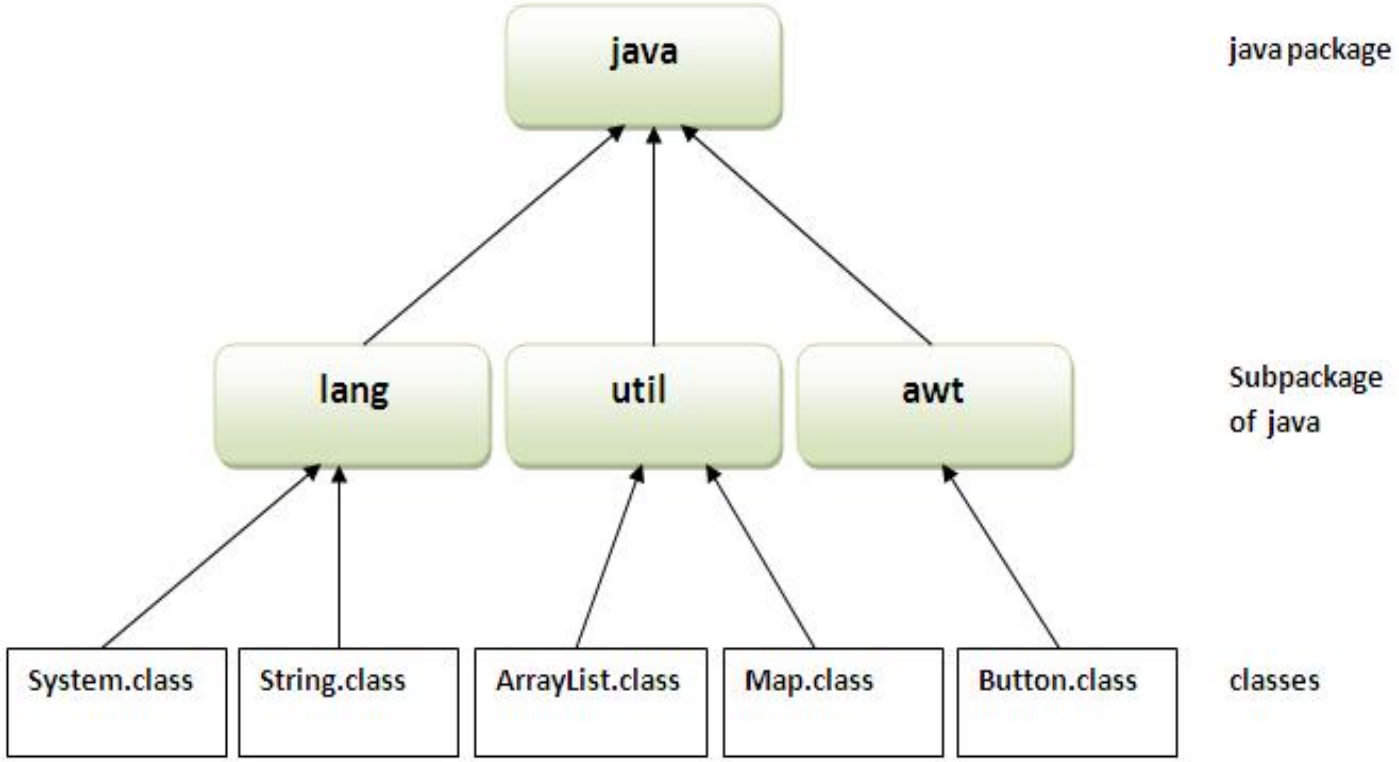
- A **Package** can be defined as a **grouping of related types (classes, interfaces, enumerations and annotations) providing access protection** and **namespace management.**

# Package

- Basically,  there are **2 types** of packages in JAVA.



# In Built Package



# Package

- **Advantage :**
  - Java package is used to **categorize** the classes and interfaces so that they can be easily maintained.
  - Java package **provides access protection.**
  - Java package **removes naming collision.**

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

# Package

- **Creating package Syntax:**

  **package package_name;**

- **Example:**

  **package employee;**

# Package

```
package first;

public class Simple // simple.java

{

     public static void main(String args[])

     {

    System.out.println(“Package is created");

   }

}
```

# Package

- **Compile java package:**

  javac -d Destination_folder file_name.java

- **After compile package,**
  - The folder with the given package name is created in the specified destination,
  - the compiled class files will be placed in that folder.

# Package

- **Compile :** javac -d . Simple.java
- **Run :** java first.Simple
- **Output:** Package is created

- **Note:**
  - The -d is a switch that tells the compiler where to put the class file. It represents destination.
  - The . represents the current folder.