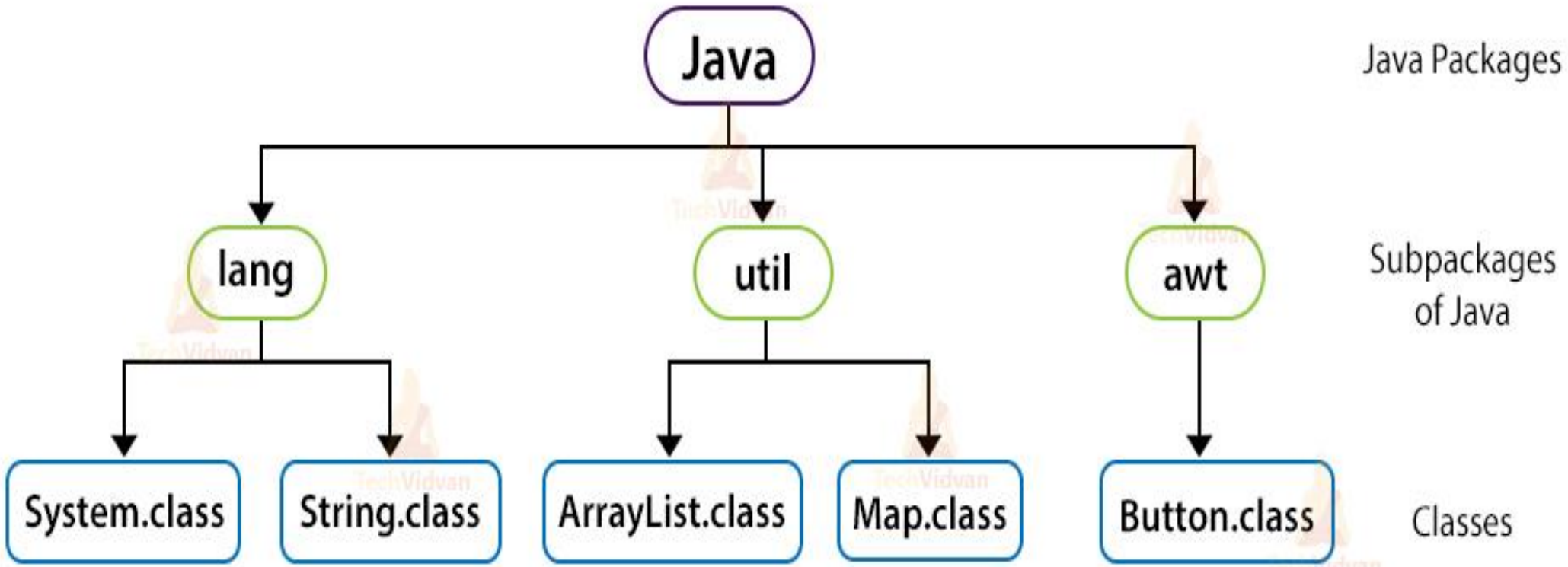
# Packages in Java

# What is Package in Java?

- **PACKAGE in Java** is a collection of classes, sub-packages, and interfaces.
- It helps organize the classes into a folder structure and make it easy to locate and use them.
- More importantly, it helps improve code reusability.
- Each package in Java has its unique name and organizes its classes and interfaces into a separate namespace, or name group.
- Although interfaces and classes with the same name cannot appear in the same package, they can appear in different packages. This is possible by assigning a separate namespace to each Java package.

# Types of Packages in Java

# Built-in Packages in Java



# Java API packages or built-in packages

- **java.lang:** It contains classes for primitive types, strings, math functions, threads, and exceptions.
- **java.util:** It contains classes such as vectors, hash tables, dates, Calendars,Scanner etc.
- **java.io:** It has stream classes for Input/Output.
- **java.awt:** Classes for implementing Graphical User Interface – windows, buttons, menus, etc.
- **java.net:** Classes for networking
- **java. Applet:** Classes for creating and implementing applets

# User-defined packages

- As the name suggests, these packages are defined by the user.
- We create a directory whose name should be the same as the name of the package.
- Then we create a class inside the directory.

# Compiling a Java Package

## javac -d . Example.java

-d specifies the destination where to locate the generated class file.

You can use any directory name like /home (in case of Linux), C:/folderName (in case of windows), etc.

If you want the package to be present in the same directory, you can use the **dot ( . )**

# Accessing Packages or Classes from Another Package

- If we want to access all the classes and interfaces of an existing package then we use the **import** statement.
- We can do it in three different ways:

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

- By using * after the import statement, we can access all the classes of the package but not the sub-packages.
- import packageName.*;

# Example MyClass.java

```
package p1; //package
class MyClass
{
	public void printName(String name)
	{
		System.out.println(name);
	}
}
```

# MyClass1.java

```
import p1.*; //importing all the classes
public class MyClass1
{
public static void main(String args[])
{
// Initializing the String variable with a value
String name = “CSE";
// Creating an instance of class MyClass from another package.
MyClass obj = new MyClass();
obj.printName(name);
}
}
```

# output

```
C:\Users\nhrao\Desktop\packages>javac -d . MyClass.java

C:\Users\nhrao\Desktop\packages>javac MyClass1.java

C:\Users\nhrao\Desktop\packages>java MyClass1
CSE

C:\Users\nhrao\Desktop\packages>
```

# Using a Fully qualified name

```
package p1; //package
class MyClass
{
	public void printName(String name)
	{
		System.out.println(name);
	}
}
```

# MyClass1.java

```
public class MyClass1
{
public static void main(String args[])
{
// Initializing the String variable with a value
String name = “CSE";
// Creating an instance of class MyClass from another package.
p1.MyClass obj = new p1.MyClass();
obj.printName(name);
}
}
```

# output

```
C:\Users\nhrao\Desktop\packages>javac -d . MyClass.java

C:\Users\nhrao\Desktop\packages>javac MyClass1.java

C:\Users\nhrao\Desktop\packages>java MyClass1
CSE

C:\Users\nhrao\Desktop\packages>
```

# Calculator Example Using Packages (Calculator.java)

```
package cal;
public class Calculator
{
public int add(int a, int b)
{
return a+b;
}
public int sub(int a, int b)
{
return a-b;
}
public int mul(int a, int b)
{
return a*b;
}
public int div(int a, int b)
{
return a/b;
}
}
```

# CalDemo.java

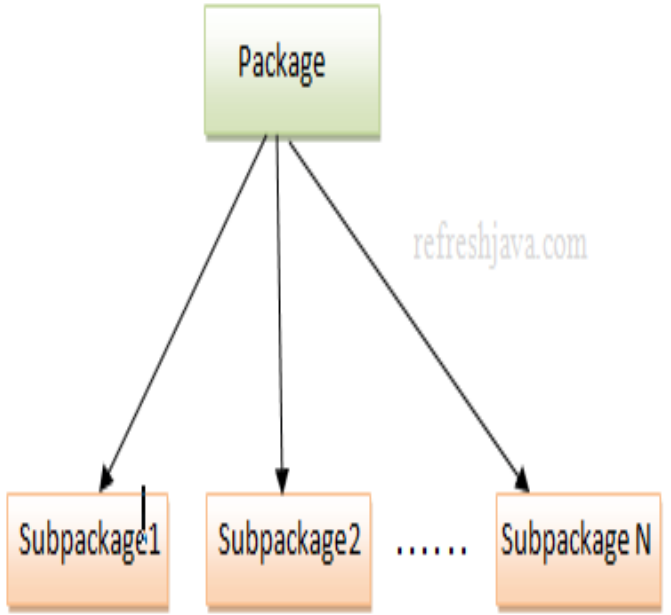
```
import cal.Calculator;
class CalDemo
{
public static void main(String a[])
{
Calculator c=new Calculator();
System.out.println("Sum="+c.add(10,20));
System.out.println("Sub="+c.sub(10,20));
System.out.println("Mul="+c.mul(10,20));
System.out.println("Div="+c.div(10,20));
}
}
```

```
C:\Users\nhrao\Desktop\packages>javac -d . Calculator.java

C:\Users\nhrao\Desktop\packages>javac CalDemo.java

C:\Users\nhrao\Desktop\packages>java CalDemo
Sum=30
Sub=-10
Mul=200
Div=0
```

# Subpackage in java

- Package inside the package is called the **subpackage**.
- It should be created **to categorize the package further**.
- **Example:**

```
package LearnJava.corejava;
class Simple{
  public static void main(String args[]){
   System.out.println("Hello from subpackage");
  }
}
```



# How to import Sub packages?

- // To import all classes of a sub package
- import packagename.subpackagename.*;
- // To import specific class of a sub package
- import packagename.subpackagename.classname;