SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)
Approved by AICTE, New Delhi, Affiliated to Anna University, Chennai
Accredited by NAAC-UGC with 'A++' Grade (Cycle III) &
Accredited by NBA (B.E - CSE, EEE, ECE, Mech&B.Tech.IT)
COIMBATORE-641 035, TAMIL NADU

UNIT IV
INHERITANCE AND POLYMORPHISM
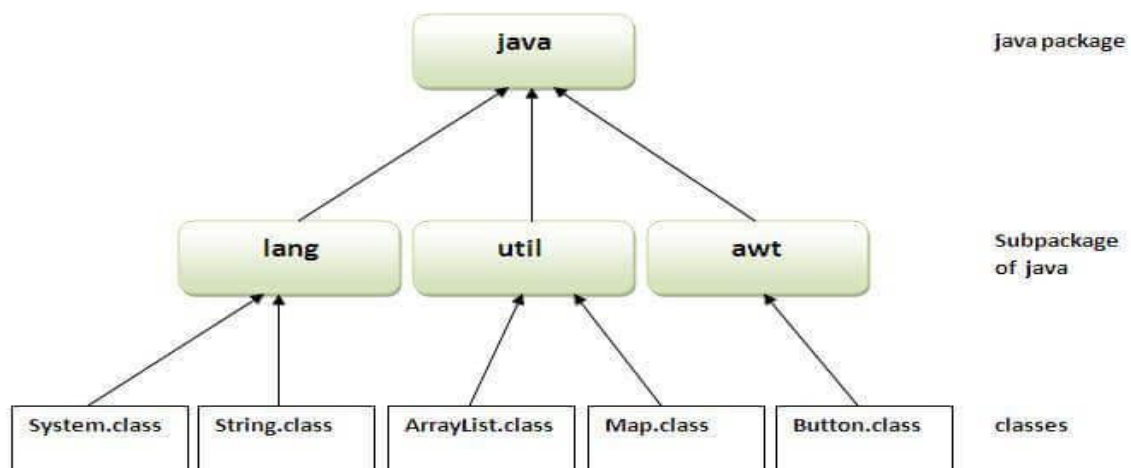
JAVA PACKAGE

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

Package in java can be categorized in two form, built-in package and user-defined package.

There are many built-in packages such as java, lang, awt, javax, swing, net, io, util, sql etc.

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.

Simple example of java package

The **package keyword** is used to create a package in java.
//save as Simple.java

```
    package mypack;
    public class Simple{
     public static void main(String args[]){
       System.out.println("Welcome to package");
      }
    }
```

How to compile java package

If you are not using any IDE, you need to follow the **syntax** given below:

    javac -d directory javafilename

For **example**

    javac -d . Simple.java

The -d switch specifies the destination where to put the generated class file. You can use any directory name like /home (in case of Linux), d:/abc (in case of windows) etc. If you want to keep the package within the same directory, you can use . (dot).

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.

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
1. //save by A.java
2. **package** pack;
3. **public class** A{
4.   **public void** msg(){System.out.println("Hello");}

5. }
1. //save by B.java
2. **package** mypack;
3. **class** B{
4.   **public static void** main(String args[]){
5.   pack.A obj = **new** pack.A();//using fully qualified name
6.   obj.msg();
7.   }
8. }
Output:Hello