ENCAPSULATION

Java Package :

is a group of similar types of classes, interfaces and sub-packages.

built-in
(lang, util, awt etc)

User-defined
(package abhi;)

> Advantages

> Java package is used to categorize the classes and interfaces, so they can be easily maintained.

> provides access protection

> removes naming collision.

> How to compile package in java -

```
javac -d directory javafilename
javac -d . javafilename
```

. represents current folder.

-d is a switch specifies the destination where to put the generated class file.

>Ways to access the package from outside the package —

> <u>import package.*</u> ;
then all the classes and interfaces of this package will be accessible but not subpackages.

> <u>Using package.classname</u>

then only declared class of this package will be accessible.

> <u>Using fully qualified name</u>
- then only declared class of this package will be accessible.
- there is no need to import.
- It is generally used when two packages have same class name eg java.util and java.sql packages contain Date class.

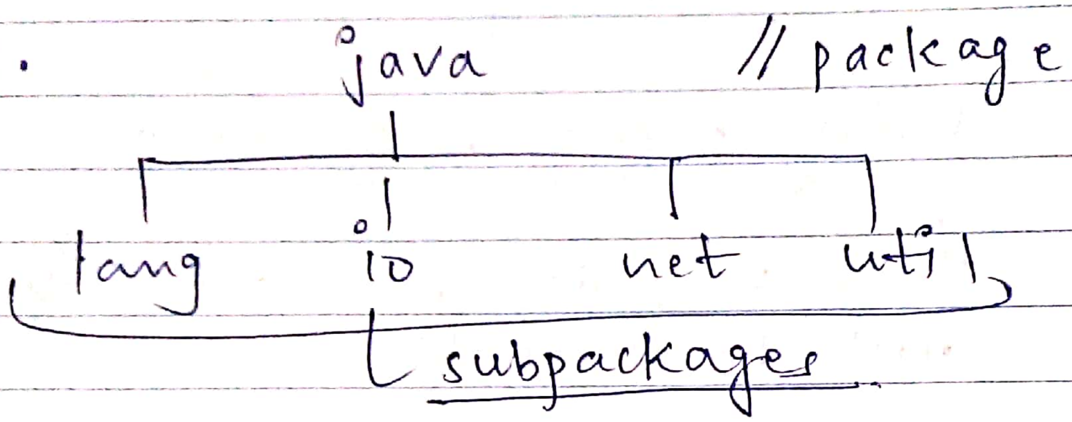
<u>Note</u>:
If you import a package, subpackages will not be imported.

Subpackage in Java :

> Package inside the package is called the subpackage.

> It should be created to categorize the package further.

Eg.



How to send the class file to another directory or drive -

>javac -d directorypath filename.java

eg. c:\classes

Two ways now

(1) > set classpath = c:\classes

> java packagename.filename

or main method name containing class

(2) > by -classpath switch of java

eg java -classpath c:\classes; pack.Simple

> Ways to load the class files temporary and permament.

> Temporary
- By setting the classpath in cmd
- By -Classpath switch

> Permament
- By setting the classpath in enviroment variables.
- By creating the jar file, that contains all the class files, and copying the jar file in the jre/lib/ext folder.

If you want to put two public classes in a package, have two java source files containing one public class, but keep the package name same.

Java Static Import] Feature of Java5

> faciliate the java programmer to access any static member of a class directly. There is no need to qualify it bythe class name.

> less coding is required if you have to access any static member of class oftenly.

> If you overuse that static import feature, it makes program unreadable and unmaintable.

Eg.
```
• import static java.lang.System.*;
  class Example
  {
          pvsm(String args[])
          {
             out.println("Hi!");
             out.println("whatsup");
          }
  }
```

static import provides accessibility to static members of the class.

Access-Modifiers in Java :-

Modifiers

access modifiers

- specifies the accessibility or scope of a field, method, constructor, or class.
- default, private, protected & public

non-access modifiers

- they provide information about the behavious to the JVM.
- static, final, abstract, transient, volatile, native, synchronized.

private :- access level is only within the class.

- cannot be accessed from outside the class.
- any other class of same package will not be able to access these members.

Default :- access level is only within package.

- cannot be accessed from

outside the package.

> If you not specify any access modifier,
it will be default.

> Protected :- access level of a protected modifier is within the package and outside the package through child class.
- If you not make child class, it cannot be accessed from outside the package.

> public :- access level is everywhere.

Java Access Modifiers with Method Overriding :-

If you are overriding any method, overridden method (i.e declared in subclass) must not be more restrictive.

```
class A
{
 protected void msg(){
  SOPn("Hello Abhi");
 }
}
```

```
public class Main extends A
{
    void msg ()
    {                                        // C.T.
        SOPn("Hello Abhishek");              Error
    }

    pvsm (String args[])
    {
        Main obj = new Main();
        obj.msg();
    }
}
```

The default modifier is more restrictive than protected. That is why C.T. Error.

Access Modifier	Within class	within package	outside package by subclass only	~~mo~~ outsid packag
private	Y	N	N	N
Default	Y	Y	N	N
Protected	Y	Y	Y	N
Public	Y	Y	Y	Y