1. **Package Definition.**
2. **Package Types.**
3. **Coding Standard for package Name**
4. **Package Creation.**
5. **Access class from another class.**
6. **Sub packages.**

**1.Package Definitions:**

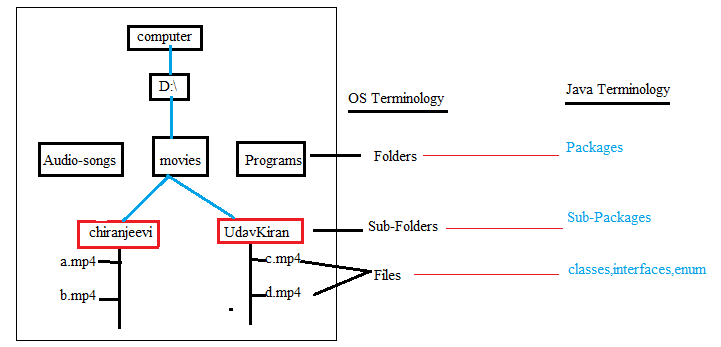
Def1: The package is folder.

Def2: The package is container.

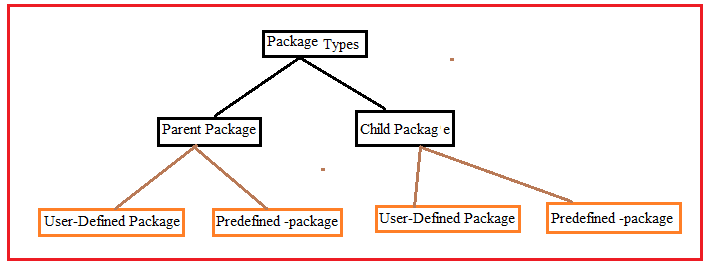
The package have related classes, enums, interfaces and sub packages. The package is used to separate the related classes from existed classes.

General Example to package:

We generally organizes similar kind files as one group using folder or directory concept in computer. The folder wise organization is nothing but a package concept in java .The files which is in the folder are classes, interfaces. The sub folders which is in the folder have to be considered as sub packages.



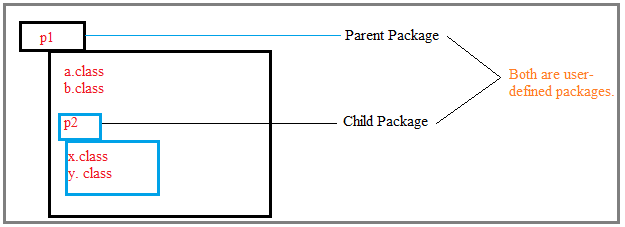
**2.Package Types:**



2.1.Parent Package:- The package which contains another sub-package is called parent package.

2.2.Child Package:- The package which is created inside other package is said to be a child package.

2.3.User-Defined Package:- The package which is created by developer is called user-defined package.



2.4.Pre-defined package/Built-in package:- Java language has already defined packages. These packages are said to be built-In packages.

Examples:

1.java.util.

2.java.lang.

3.java.awt. …etc.

**3. Coding Standard for package Name:**

i.



Example:

RaosInstitute.Arithmetic;

ii. Consider the organization website url and put name to package.



Example:

Com.raos.Arithmetic;

Note: The package name should have only small case letters.

Note:- The package should have minimum 3 words which should be separated by (.).

**4.Package Creation:-**

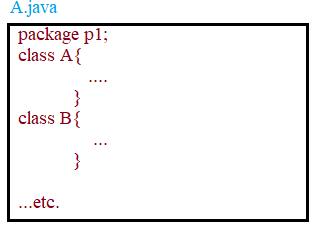
**Syntax:**

****

The package should be first statement in .java file. One java file has only one package statement.

Example:1

Step1: Creation



Step2: compilation.



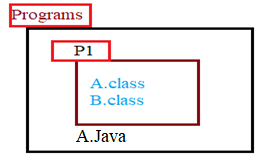
Where (.) represents current working directory.

Note: 1.compiler does not create package physically in current working directory just with javac command.

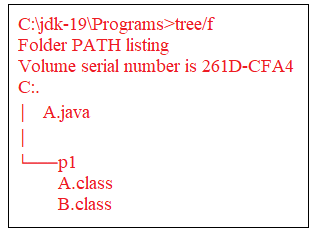
2. -d functionality is creating package with name mentioned in java file and moving all files(.class) in that package, and finally storing that package in given address .If package already exists with name then it does not create new one and it moves all files(.class) in that existed package.



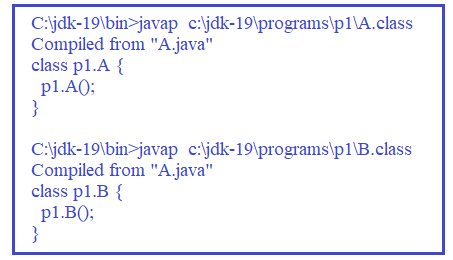
The current working directory(programs) is said to be a “default Package”.



After compilation , programs folder structure will be like this:



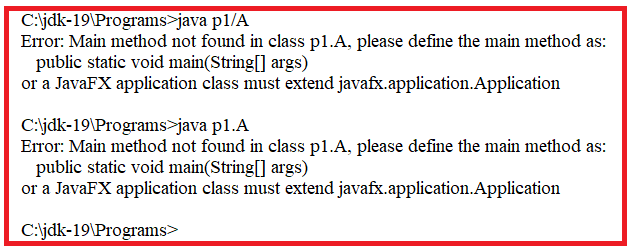
Note: The compiler will replaces class name and its constructor name as packagename.className and packageName.constructorName.



Step3: Execution.

Syntax to executed packaged class

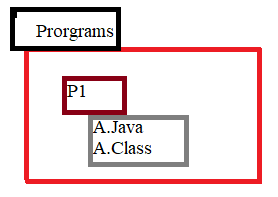




Note: Actually A.java did’t have main() method. That’s why we got error message.

Error: main Method not found in class p1.A.

Example:2 Package P1 already existed. In that place A.class and A.java files.



Step1: create Folder named “P1”.

Step2. Create “A.java” file in P1.

package P1;

public class A

{

public static void main(String args[])

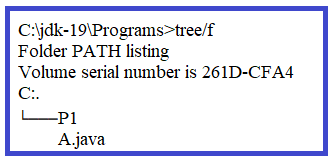
{

System.out.println("A-m1");

}

}

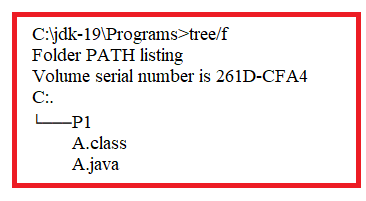
Step3.Now Programs folder structure is:



Step4. Compile the A.java file.



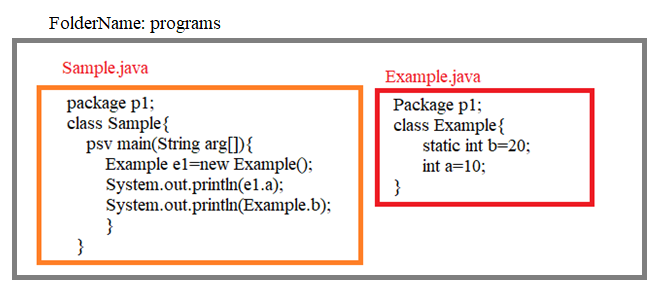
Step5. Now Programs folder structure is:



Step6: Run the A.class file.



**5. Accessing class from another class:** If two classes are in same package then one class directly access another class by its name.



>javac -d . sample.java

Step1. Compiler search for Example class definition in sample.java, if not found,

Step2. Compiler search for Example.class file in same package , if not found,

Step3. Compiler search for Example.java file in same package. if found, compiler compiles example.java. if not found,

Step4. Compiler search for Example.class file in imported packages, if not found,

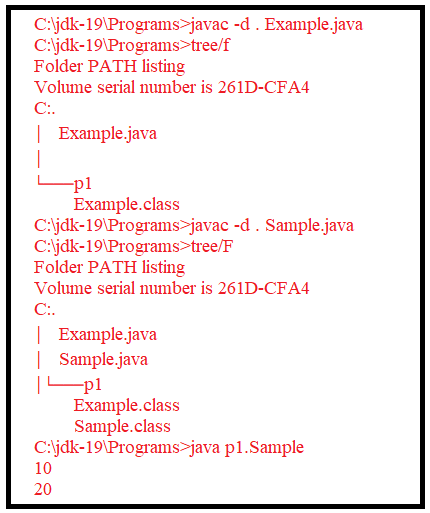
Step5. It search for Example.java file in imported packages, if not found then compiler terminates sample.java file compilation by throwing

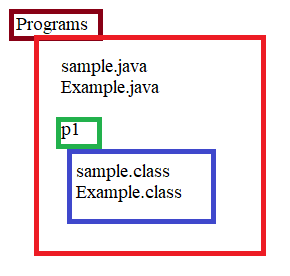
CE:can not find symbol.

Step6. If example.java file is found in imported packages, then compiler search for Example class definition in example.java file. It found compiles otherwise terminates sample.java file compilation by throwing

CE: can not find symbol.

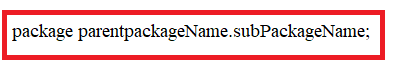
Compilation ,Execution and output of above programs:



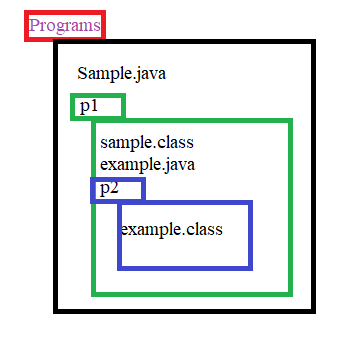


**6.SubPackage:** The package which is created inside a other package is called sub package.

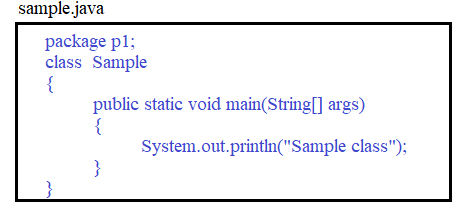
Syntax:



Example: create a following package structure.



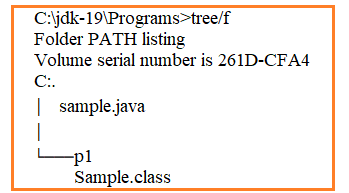
1.



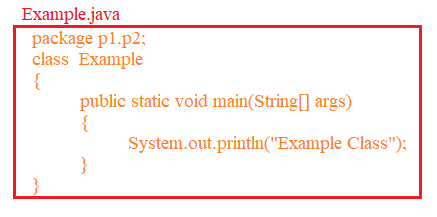
2. compile the sample file.



After compilation “programs” folder structure will be as below:



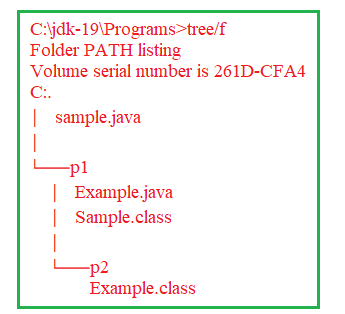
3.



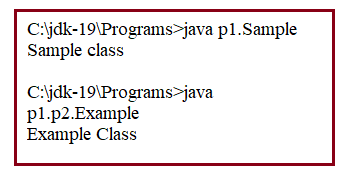
4. compile the example.java



After compilation, “Programs” folder structure will be as below.



5.Execute the two classes.



6.1.Accessing one package class from another package class: If we want to access one package class from another package class, we must follow 3 rules.

R1: class should be public.

R2: The method or variable in class we want to access should be public.

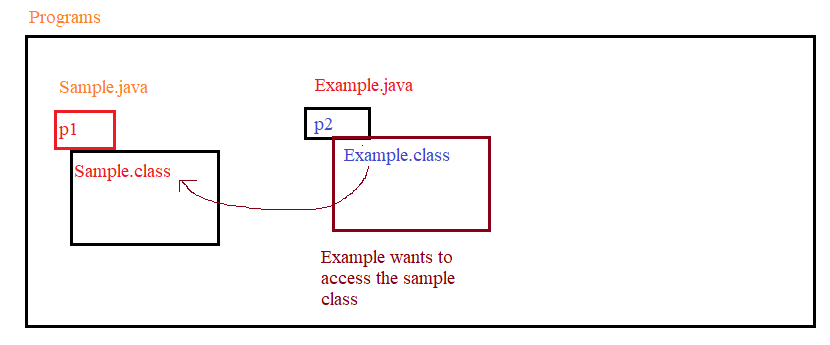
R3: In Another class, we should use “fully qualified name”to access this class.

Note: 1.The two packages may resides in same parent package.

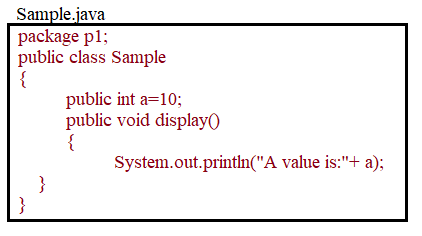
2. The two packages may resides in different parent packages.

3. There may be parent/child relationship between two packages.

Example:1



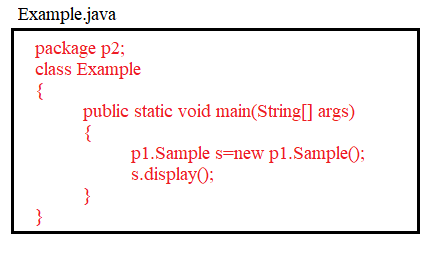
1.



2. Compile Sample.java



3.



4. compile Example.java



5. Run Example program.

