Syntax :- import packagename.* ;

or

import packagename.classname ;

import packagename.subpackagename.* ; ← subpackage

Types

Built-In Package — User Defined Package *imp

java.net
java.lang
java.applet

* package is a keyword. It is used to create package

```
package pack;                      // pack can be any name
public class Addition              // should be declared public
 {
    public void add ( int a, int b)              or        public static void add (....)
     {
        int c;
         c = a+b;
         Sopln (" sum ="+c);
     }
 }
```

Output

```
javac -d . Addition.java
```

name of filename

```
package pack;
public class Subtraction
 {
    public void diff ( int a, int b)
     {
        int c = a-b;
        Sopln ("Difference ="+c);
     }
}
```

SUB-PACKAGE

```
package pack.subpack;
public class Multiplication
 {
    public void product( int a, int b)
      {
         int c = a*b;
         Sopln ("Multiplication="+c);
      }
}
```

```
javac -d . Multiplication.java
```

(now package will be created in subpack folder)

```
package pack.subpack;
public class Division
 {
    public void div( int a, int b)
      {
         int c = a/b;
         Sopln ("Result of Division ="+c);
      }
 }
```

How to use those package?

```
import pack.Addition;
class Main
   {
      public static void main ( string ar[ ])
        {
           Addition ob1 = new Addition();
           ob1.sum(23,45);
        }
   }
```

If we want add another package :-

```
import pack.Addition;
import pack.Subtraction;
import pack.subpack.Multiplication;
class Main
   {
      public static void main ( string ar[ ])
        {
          Addition ob1 = new Addition();
          ob1.sum(23,45);
          Subtraction ob2 = new Subtraction();
          ob2.diff(93,43);
         Multiplication ob = new Multiplication();
          ob2.product(2,5);
        }
   }
```

If we put static in packages ; then we dont need to create objects.

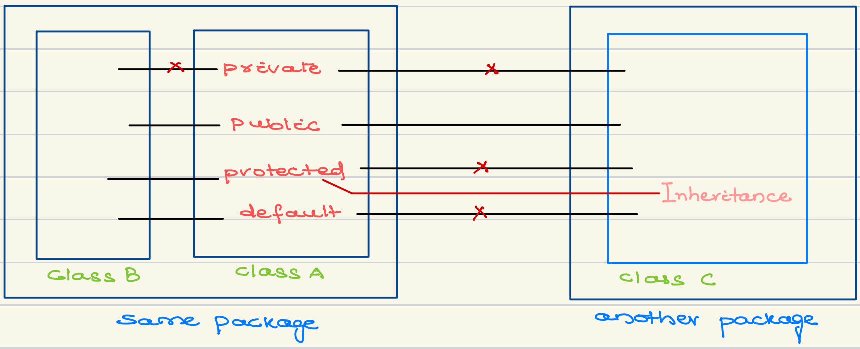
```
package pack.subpack;
public class Multiplication
 {
    public void static product( int a, int b)
     {
        int c = a*b ;
        Sopln ("Multiplication="+c);
     }
 }

import pack.subpack.Multiplication ;
class Main
 {
    public static void main ( String args[ ] )
     {
        Multiplication.product ( 10 , 3 ) ;
     }
 }
```

Access Specifier / Visibility Modifier :-

Types of Access Specifier :-

- Private
- Public
- Protected
- Default



The class path is an environment variables that tells the java compiler where to look for class files.

Syntax to set class path :-

```
set CLASSPATH = file location ;.; ↵
 javac P1.java ↵
 java Main ↵
```

→ ADD ONE INTERFACE AND ITS IMPLEMENTING CLASS INSIDE THE PROGRAM :-

```
package mypack ;
public interface Fact
{
    public void factcal (int f) ;
}
```

```
javac -d . Fact.java ↵
```

*imp

```
package mypack ;
import mypack.Fact ;
public class Factorial implements Fact
{
  public void factcal (int f)
   {
     int p = 1 ;
     for ( int i = 1 ; i<= f ; i++)
      {
        p = p * f ;
      }
      Sopln (`Factorial of " + f + " = " + p) ;
     }
  }
```

```
javac -d . Factorial.java ↵
```

* If we want to use to use import & package statement together, the package statement will be the first statement. (otherwise it will show error)

```
import mypack.Factorial ;
class Main1
{
   public static void main ( String ar[])
    {
       Factorial ob = new Factorial ( ) ;
        ob. factcal (5) ;
      }
 }
```

```
javac P2.java
java Main1
```

/* Output

```
Factorial of 5 = 120
```

```
package mypack ;
import mypack.Fact ;
public class Factorical implements Fact
{
   public void factcal (int f)
     {
       int p=1 ;
       for ( int i= 1 ; i<= f ; i++)
         {
           p= p*f ;
         }
          Sopln ("Factorical of " + f + " = "+p) ;
        }
    }
```