PACKAGES

NEED :

Packages are used in java in order to prevent naming conflicts , to control access , to make searching/locating and usage of classes , interfaces,enumerations and annotations easier.

In short, We use packages to avoid name conflicts, and to write a better maintainable code.

DEFINITION:

The package in java is used to group the related classes. Think of it as a folder in a file directory.

NOTE:

If you don't use a package statement in your program ,your type ends up in an unnamed packages.Generally speaking,an unnamed package is only for small (or) temporary applications.

TYPES:

- Built-in Package-the java API is a library of prewritten classes, that are free to use, included in the java development environment
- User defined package-a packages which are created by the programmer(user)
  We can add more classes to a created package by using package name at the top of the program and saving it in the package directory. We need a new java file to define a public class, otherwise we can add the new class to an existing .java file and recompile it.

  import java.util.*;

  util is a subpackage created inside java package

NOTE:
API-Application Programming Interface is a document which gives you the list of all the packages,classes, and interface ,along with their fields and methods.

ADVANTAGES:

- The Java package is used to categorize the classes and interfaces so that they can be easily maintained.
- Java package provides access protection.
- Java package removes naming collision.

- **Packages** provide reusability of code .
- To bundle classes and interfaces.
- We can create our own **Package** or extend the already available **Package.**

DISADVANTAGES:

- We cannot pass parameters to packages.
- Change in a function needs to be reflected in all the functions that use the changed function and hence the whole package needs to be recompiled.

CONSTRAINS:

- Package statement must be the first statement in the program even before the import statement.
- A package is always defined as a separate folder having the name as the package name.
- Store all the classes in that package folder.
- All classes of the package which we wish to access outside the package must be declared public.
- All classes within the package must have the package statement as its first line.
- All classes of the package must be compiled before use.

IMPORT KEYWORD:

import keyword is used to "access" other package members from this package class. Actually, it does not import other members into the package,instead it shows the path of the other package members to the compiler and JVM.

PACKAGE KEYWORD:

Package is a Java keyword. It declares a 'namespace' for the Java class. It must be put at the top of the Java file, it should be the first Java statement line.Ensure that the package name will be unique.

SYNTAX:

To create the package:

```
package packagename;
public class classname
{
  …...Data members;
```

```
    public void method()   {
      ……….content ;
    }
}
```

To import:

```
import  package name.class;   // import a single class
import  package name.*;   // import the whole package
class Main
{
       public static void main(String args[])
      {
            class obj=new class();  //creating an object for the package class
            obj.method();  //calling the method in this package present inside the class
      }
}
```

How packages work ?

Package names and directory structure are closely related. For example if a package name is college.cse.staff, then there are three directories, college, staff and cse such that staff is present in cse and cse is present college. Also, the directory college is accessible through the CLASSPATH variable, i.e., the path of the parent directory of college is present in CLASSPATH. The idea is to make sure that classes are easy to locate.

Package naming conventions:

Packages are named in reverse order of domain name:

DOMAIN NAME	PACKAGE NAME PREFIX
hyphenated_name.example.org	org.example.hyphenated_name
example.int	int_.example
123name.example.com	com.example._123name

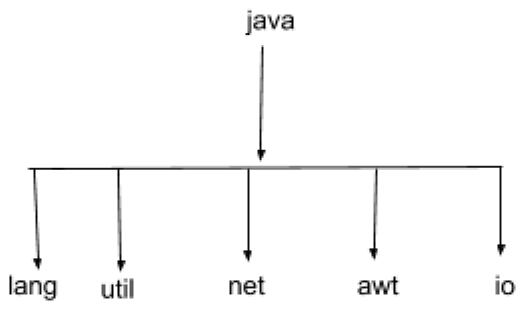
NOTE:

- Sequence of the program must be;
    - Package
    - Import

    ○ Class

BUILT-IN PACKAGES:

A well-known built-in package is Scanner which is used to get user input. Here Scanner is not a package, it is the class of the particular package called java.util. Some examples are,



EXAMPLE:

```
import  java.util.Scanner;

class Main

{

        public static void main(String args[])

        {

        Scanner obj=new Scanner(System.in);

        System.out.println("Enter username:");

        String name=obj.nextLine();

        System.out.println("Username:"+name);

        }
}
```

Output:

```
Enter the username:

Student.

Username:Student.
```

Explanation:
In the example above, java.util is a package, while Scanner is a class of the java.util package.To use the Scanner class, create an object of the class and use any of the available methods found in the Scanner class documentation. In our example, we will use the nextLine() method, which is used to read a complete line.

USER-DEFINED PACKAGE:

To create our own package,we need to understand that Java uses a file system directory to store them. Just like folders on the computer.

EXAMPLE:

Single package with one method and import into the main:

```
package pack;
class Base
{
  String data="data member";
  public void method1()
  {
      System.out.println("pack-base-method");
  }
}


import pack.Base;
class Main
{
   public static void main (String[] args) {

      base b=new base();
      b.method1();
      System.out.println(b.data);
   }
}
```

Output:

```
pack-base-method
data member
```

Explanation :
In this Example,one package is created with the name pack.In that package a Base class is created. Then that package pack is imported to the main class.By creating objects
To the base class,so we accessing data member and member function of the base class

Package with multiple methods:

```
package pack;
class Base
{
   String data="data member";
   public void method1()
   {
      System.out.println("pack-base-method1");
   }
   public void method2()
  {
      System.out.println("pack-base-method2");
  }
  public void method3()
  {
        System.out.println("pack-base-method3");
  }
}

import pack.Base;
class Main
{
   public static void main (String[] args) {

      base b=new base();
      b.method1();
      b.method2();
      b.method3();
      System.out.println(b.data);
   }
}
```

Output:

```
pack-base-method1
pack-base-method2
pack-base-method3
data member
```

Explanation :
In this Example,one package is created with the name pack.In that package a Base class is created. Then that package pack is imported to the main class.By creating objects
To the base class,so we accessing data member and member function of the base class .In this program additionally,in base class three methods are created

Multiple packages:

Case1:multiple packages with same class

```
package p1;

public class Base{
   public String data="data member1";
   public void method()
   {
        System.out.println("p1-base-method");
   }
}

package p2;
public class Base
{
   public String data="data member2";
   public void method()
   {
        System.out.println("p2-base-method");
   }
}

package p3;

public class Base {
   public String data="data member3";
   public void method()
   {
        System.out.println("p3-base-method");
   }
}

import p1.Base;
import p2.Base;
import p3.Base;
class Main
{
  public static void main(String args[])
 {
     p1.Base b1=new p1.Base();
     p2.Base b2=new p2.Base();
     p3.Base b3=new p3.Base();
     b1.method();
     System.out.println(b1.data);
      b2.method();
      System.out.println(b2.data);
      b3.method();
      System.out.println(b3.data);
   }
}
```

Output:

```
p1-base-method
data member1
```

```
p2-base-method
data member2
p3-base-method
data member3
```

Explanation:
In this Program,Three packages are created Named p1,p2 and p3.Then Importing all Three packages with their respective Base class(i.e, BaseA,BaseB and BaseC).In main class,Creating Objects to the Package Classes(b1,b2 and b3).By using those object ,Operating Data member and Member Functions of those Base class of the Packages.In additionally creating object using packages names(i.e,p1.Base b1 =
Note :
If the classes in these packages are same,thus using fully qualified name (ie.p1.Base)

Case2: multiple packages with different classes

```
package p1;
public class BaseA
{
   public String data="data member1";
   public void method()
   {
        System.out.println("p1-baseA-method");
   }
}

package p2;
public class BaseB
{
   public String data="data member2";
   public void method()
   {
        System.out.println("p2-baseB-method");
   }
}

package p3;
```

```
public class BaseC
{
   public String data="data member3";
   public void method()
   {
        System.out.println("p3-baseC-method");
   }
}

import p1.BaseA;
import p2.BaseB;
import p3.BaseC;
class Main
{
  public static void main(String args[])
 {
     BaseA b1=new BaseA();
     BaseB b2=new BaseB();
     BaseC b3=new BaseC();
     b1.method();
     System.out.println(b1.data);
      b2.method();
      System.out.println(b2.data);
      b3.method();
      System.out.println(b3.data);
   }
}
```

Output:

```
p1-baseA-method
data member1
p2-baseB-method
data member2
p3-baseC-method
data member3
```

Explanation:

In this Program,Three packages are created Named p1,p2 and p3.Then Importing all Three packages with their respective Base class(i.e, BaseA,BaseB and BaseC).In main class,Creating Objects to the Package Classes(b1,b2 and b3).By using those object ,Operating Data member and Member

Functions of those Base class of the Packages

Case3 : packages with different classes using inheritance(same method())

```
package pack;
public class Base
{
  public  String data="data member";
   public void method()
   {
       System.out.println("pack-base-method "+data);
   }
}

package pack;
public class Derived extends Base
{
    public void method()
   {
       System.out.println("pack-derived-method "+data);
   }
}

package pack;
public class C
{
   Base b=new Base();
   public void method()
   {
       System.out.println("pack-C-method "+b.data);
   }
}

import pack.Base;
import pack.Derived;
import pack.C;
class Main
{
   public static void main (String[] args) {
      Base b1=new Base();
```

```
     b1.method();
     Derived d=new Derived();
     d.method();
    C c=new C();
    c.method();
  }
}
```

Output:

```
pack-base-method data member
pack-derived-method data member
pack-C-method data member
```

Explanation:
In this Program,Three Packages were Created with the same name pack.In main class creating objects to all the Class Base,Derived and C.By using Those objects operating the member functions of the Class(i.e, Base,Derived,C).
Note:
Derived class is inherited using extends Keyword.It is Derived Class for Base class.So Creating objects to Derived class can Access the Member function of the Base class

Case4 : packages with different classes using inheritance(different method())

```
package pack;
public class Base
{
  public  String data="data member";
   public void method()
   {
        System.out.println("pack-base-method "+data);
   }
}

package pack;
public class Derived extends Base
{

   public void method1()
  {
      System.out.println("pack-derived-method1 "+data);
  }
}

package pack;
public class C extends Derived
{
 // Base b=new Base();
  public void method2()
  {
      System.out.println("pack-C-method2 "+data);
  }
}


import pack.Base;
import pack.Derived;
import pack.C;
class Main
{
  public static void main (String[] args) {
    //Base b1=new Base();
    C c=new C();
    c.method();
    //Derived d=new Derived();
    c.method1();
    c.method2();
  }
}
```

Output:

```
pack-base-method data member
pack-derived-method1 data member
pack-C-method2 data member
```

Explanation:

In this Example also Three packages with same name pack are created.In 1st pack we created a Base class,in 2nd pack Derived class is created for the Base class using Extends Keyword and in 3rd pack Class C is Created as a Derived class for Derived class.Therefore By Creating Object to Class C can Operate all the Member functions of the Base class and Derived class.

Case5: Packages using constructor (inheritance)

```
package pack;
public class Base
{
  public  String data="data member";
   Base(String data)
   {
       this.data= data;
       System.out.println("pack-base-method "+data);
   }
}

package pack;
public class Derived extends Base
{
   String data;
   Derived(String a,String b)
   {
       super(a);
       this.data=b;
       System.out.println("pack-derived-method "+data);
   }
}

package pack;
public class C extends Derived
{
   String data;
   C(String a,String b,String c)
   {
        super(a,b);
        this.data=c;
       System.out.println("pack-C-method "+data);
   }
}


import pack.Base;
import pack.Derived;
import pack.C;
class Main
```

```
{
   public static void main (String[] args) {
      C c=new C("Base","Derived","C");
   }
}
```

Output:

```
pack-base-method Base
pack-derived-method Derived
pack-C-method C
```

Explanation:
In this example, three packages called pack were created with classes Base,Derived and C , in all those package classes were created with Parametrized constructor. Base class is inherited to package 2 using extends keyword(i.e,derived class is derived form base class). Derived class is inherited to package 3 using extends keyword(i.e, C class is derived from derived class). Therefore Multilevel inheritance. Then all packages are imported to main class. Then object c is created to class C. While creating a object to constructor it automatically call the class. In all these class super keyword is used(it points to the immediate Base/super class). So while creating object to class C we can access the Base class.

ACCESS PROTECTION:

TYPES	PUBLIC	PROTECTED	PRIVATE	NO MODIFIER
SAME CLASS	YES	YES	YES	YES
SAME PACKAGE SUBCLASS	YES	YES	NO	YES
SAME PACKAGE AND NON-SUBCLASS	YES	YES	NO	YES
DIFF. PACKAGE SUBCLASS	YES	YES	NO	NO
DIFF. PACKAGE NON-SUBCLASS	YES	NO	NO	NO

TYPE 1:

```
package p1;
public class Base
{
        public int a=10;
        int b=20;
        protected int c=30;
        private int d=40;
        public void show()
        {
                System.out.println(a+" "+b+" "+c+" "+d);

        }
}

import p1.Base;
class Main
{
        public static void main(String args[])
        {
                Base b=new Base();
                b.show();
        }
}
```

output:

```
10 20 30 40
```

Explanation:
The above file Base belongs to package p1, and contains data members with all access modifiers. Note that,all the data members within the same class can be accessed.

TYPE 2:

```
package p1;
public class Base
{
	public int a=10;
	int b=20;
	protected int c=30;
	private int d=40;
	 public void show()
	{
		System.out.println(a+" "+b+" "+c+" "+d);
	}
}

package p1;
public class Derived extends Base
{
	public void show()
	{
		System.out.println(a);
		System.out.println(b);
		System.out.println(c);
		System.out.println(d);

	}
}

import p1.Base;
import p1.Derived;
class Main
{
	public static void main(String args[])
	{
		Base b=new Base();
		b.show();
		Derived d=new Derived();
		d.show();
	}
}
```

Output:

```
10 20 30 40
Exception in thread "main" java.lang.Error: Unresolved
compilation problem:
	The field Base.d is not visible
	at p1.Derived.show(Derived.java:10)
	at Main.main(Main.java:10)
```

Explanation:
The above file Base belongs to package p1, and contains data members with all access modifiers. Note that, all the data members within the same class can be accessed. And ,the above file Derived belongs to package p1, and extends Base, which belongs to the same package. Note that,private datamember can't be used outside the class. By ignoring that,

Output is :

```
10 20 30 40
10
20
30
```

TYPE 3:

```
package p1;
public class Base
{
        public int a=10;
        int b=20;
        protected int c=30;
        private int d=40;
        public void show()
        {
                System.out.println(a+" "+b+" "+c+" "+d);
        }
}

package p1;
public class Derived
{
        Base b1=new Base();
        public void show()
        {
                System.out.println(b1.a);
                System.out.println(b1.b);
                System.out.println(b1.c);
                System.out.println(b1.d);

        }
}
```

```
import p1.Base;
import p1.Derived;
class Main
{
        public static void main(String args[])
        {
                Base b=new Base();
                b.show();
                Derived d=new Derived();
                d.show();
        }
}
```

Output:

```
10 20 30 40
Exception in thread "main" java.lang.Error: Unresolved
compilation problem:
        The field Base.d is not visible
        at p1.Derived.show(Derived.java:10)
        at Main.main(Main.java:10)
```

Explanation:

The above file Base belongs to package p1, and contains data members with all access modifiers. Note that, all the data members within the same class can be accessed. And the above file Derived belongs to package p1, and having an instance of Base. by this instance , the implementation of the file Base data members into the Derived could be done. Note that,private datamember can't be able to access in Derived . By ignoring that,

Output is :

```
10 20 30 40
10
20
30
```

TYPE 4:

```
package p1;
class Base
{
        public int a=10;
        int b=20;
        protected int c=30;
        private int d=40;
        public void show()
        {
                System.out.println(a+" "+b+" "+c+" "+d);

        }
}

package p2;
public class Derived extends p1.Base
{
        //Base b1=new Base();
        public void show()
        {
                System.out.println(a);
                System.out.println(b);
                System.out.println(c);
                System.out.println(d);
        }
}

import p1.Base;
import p2.Derived;
class Main
{
        public static void main(String args[])
        {
                Base b=new Base();
                b.show();
                Derived d=new Derived();
                d.show();
        }
}
```

Output:

```
10 20 30 40
Exception in thread "main" java.lang.Error: Unresolved
compilation problems:
        The field Base.b is not visible
        The field Base.d is not visible
        at p2.Derived.show(Derived.java:8)
        at Main.main(Main.java:10)
```

Explanation:
The above file Base belongs to package p1, and contains data members with all access modifiers. Note that, all the data members within the same class can be accessed. And the above file Derived belongs to package p2. extends Base, which belongs to p1, and it is trying to access data members of the class Base. But, the data members with no-modifier access and private access can't be accessed. By ignoring that,

Output is:

```
10 20 30 40
10
30
```

TYPE 5:

```
package p1;
public class Base
{
        public int a=10;
        int b=20;
        protected int c=30;
        private int d=40;
        public void show()
        {
                System.out.println(a+" "+b+" "+c+" "+d);

        }
}

package p2;
public class Derived
{
        p1.Base b1=new p1.Base();
        public void show()
        {
                System.out.println(b1.a);
                System.out.println(b1.b);
                System.out.println(b1.c);
                System.out.println(b1.d);
        }
}
```

```
import p1.Base;
import p2.Derived;
class Main
{
        public static void main(String args[])
        {
                Base b=new Base();
                b.show();
                Derived d=new Derived();
                d.show();
        }
}
```

Output:

```
10 20 30 40
Exception in thread "main" java.lang.Error: Unresolved
compilation problems:
        The field Base.b is not visible
        The field Base.c is not visible
        The field Base.d is not visible
        at p2.Derived.show(Derived.java:9)
        at Main.main(Main.java:10)
```

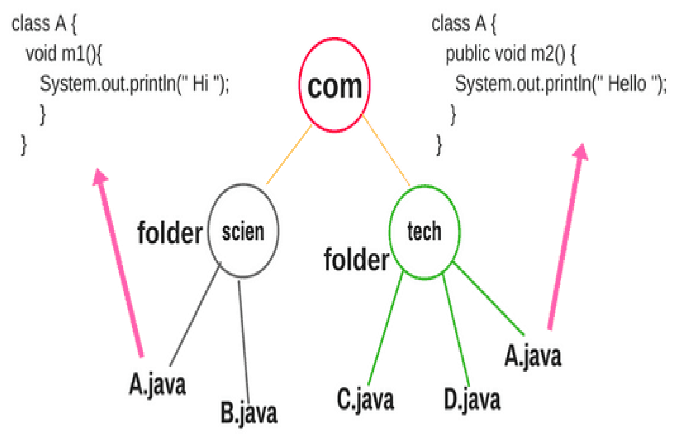
Explanation:

The above file Base belongs to package p1, and contains data members with all access modifiers. Note that, all the data members within the same class can be accessed. And the above file Derived belongs to package p2, and having an instance of Base of package p1, trying to access its some data members. Only public access can only be able to access. By ignoring that,

Output is:

```
10 20 30 40
10
```

APPLICATION TYPE PROGRAMS:



My requirement to call m1 of class A of sub-package scien and m2 of class A of sub-package tech form class B of sub-package scien.

Q1. How will you call m1 of class A of sub-package scien and m2 of class A of sub-package tech from class B of sub-package scien?

Let's create a program using the first approach to call the following requirement.

```
 package com.scien;
import com.tech.A;
 class B
 {
   void m3()
   {
     System.out.println("Hello Java");
   }
public static void main(String[] args)
 {
   A a = new A();
   a.m1();

   A a1 = new A();
    a1.m2;
   B b = new B();
    b.m3();
 }
}
```

Will the above code compile?

1. No: because the statement A a = new A(); does not say anything about class A from which sub-packages (scien or tech) it is referring.
2. No: because the statement A a1 = new A(); is also not saying anything about class A of which sub-packages (scien or tech) it is referring.
3. No: because a.m1() and a1.m2() will get confused to call the method of which package's class. Here, the compiler will be also confused.

In this case, the import is not working. So, we remove the import statement and use the fully qualified name.

```
package com.scien;
 class B
 {
  void m3()
  {
    System.out.println("Hello Java");
  }
public static void main(String[] args)
 {
   A a = new A();       // keep as it is because it is from same package "scien".
   a.m1();
    com.tech.A a1 = new com.tech.A();       // It will direct go to tech package and call the method m2.
   a1.m2;
   B b = new B();
  b.m3();
  }
```

Output:

```
Hi
Hello
Hello Java
```

Explanation:
Suppose you are not using public with m2() method in the above program then it will give error " The method m2() from the type A is not visible" because it is a default and default access modifier cannot be accessed from outside the package.

Q.Write a program creating three classes to get employee details, calculate employee salary, get company details. Create these classes with different packages. In this, inherent employee details and calculating employee salary class. Get the employee details like name, salary, working hours in a separate method. Get the company details in a separate method. While calculating salary i) addSalary() --> increments salary to 500 if the salary is above 5000
ii) addWork() --> increments salary to 250 if hours is above 6.

```java
package Emp;

public class Empdetails {
	public String Ename;
	protected double sal,hrs;
	public Empdetails(String n,double s,double h)
	{
		Ename=n;
		sal=s;
		hrs=h;
		System.out.println(Ename+" "+sal+" "+hrs);
	}
}

package Calc;

public class EmpCalculation extends Emp.Empdetails
{
	public EmpCalculation(String n,double s,double h) {
			super(n,s,h);
		}
		public void addSal()
		{
			if(sal<5000)
			{
				sal=sal+500;
			}

			System.out.println("Increment for "+Ename+" is:"+sal);
		}
		public void addWork()
		{
			if(hrs>6)
			{
				sal=sal+250;
			}
			System.out.println("Increment for "+Ename+" is:"+sal);
		}

}


package Comp;

public class CompDetail {
		public String cname,place;
		public CompDetail()
		{
			cname="";
			place="";
		}
		public void getCompInfo(String n,String p)
		{
			cname=n;
			place=p;
			System.out.println("Company:"+cname+" at "+place);
		}
}


import Emp.Empdetails;
import Calc.EmpCalculation;
import Comp.CompDetail;
import java.util.Scanner;
class Main
{
   public static void main (String[] args) {
            String name;
            double salary,hour,s1=0,s2=0;
            CompDetail c=new CompDetail();
            c.getCompInfo("INFOSYS","Mysore");
            Scanner s=new Scanner(System.in);
            System.out.println("Enter the employee name:");
            name=s.nextLine();
            System.out.println("Enter the employee salary and their working
hours:");
            salary=s.nextFloat();
            hour=s.nextFloat();
            EmpCalculation cal=new EmpCalculation(name,salary,hour);
            cal.addSal();
            cal.addWork();
   }

}
```

Output:

```
Company:INFOSYS at Mysore
Enter the employee name:
john
Enter the employee salary and their working hours:
1000
10
john 1000.0 10.0
Increment for john is:1500.0
Increment for john is:1750.0
```

Explanation:
In this program, Three packages named, Emp, Calc,Comp.in those packages required variables
and Constructors are created. Here Super keyword(which call the immediate Base class) is used in
All the classes. (Empdetails is Base class for EmpCalculation, EmpCalculation is Base class for Compdetails)
Then all packages are imported to main class. Then by creating a object to derived class Compdetials
Access the data member and member function of the package classes