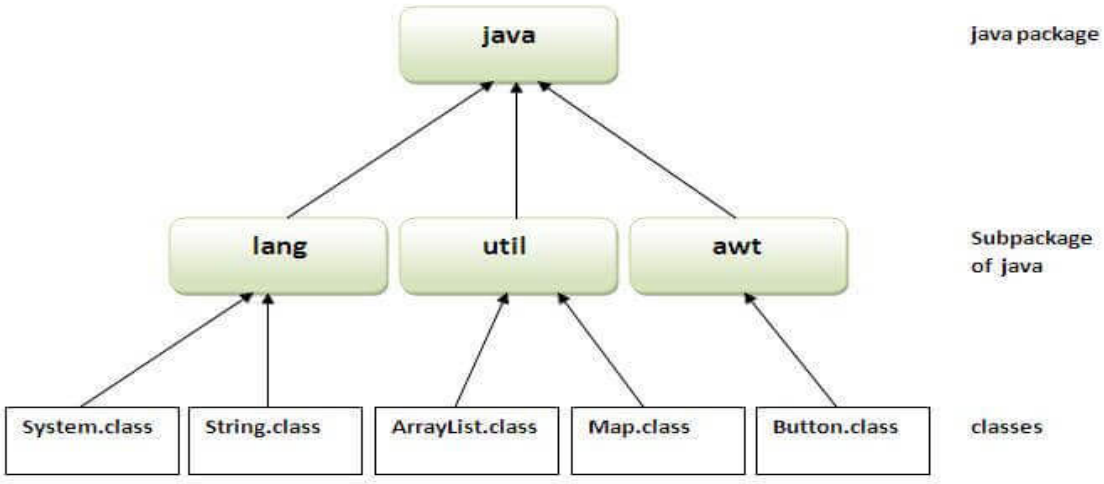
Package

- 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.

Built in packages in java

java.lang	General support package, it is automatically imported into all java programs
java.awt	This package contains classes which are required to create GUI application.
java.applet	This package contains classes which are required to create Applet application.
javax.swing	It is an advanced package of awt.
java.net	This package contains classes which are required to create network application
java.sql	This package contains classes which are required to create database application



Creating a user defined package

The "**package**" keyword is used to create a package in java.

We include a "package" keyword as the firs statement in a java source file.

Any classes declared within that file will belong to the specified package. The package statement defines a name space in which classes are stored. If you omit the package statement, the class names are put into the default package, which has no name and is called un-named package.

Syntax:

```
package   pakage_name;
```

Eg:

package MyPack;

Java uses file system directories to store packages. Remember that it is case significant and directory name must match the package name exactly. More than one file can include the same package statement.

You can create a hierarchy of packages. To do so, simply separate each package name from the one above it by use of a period or dot operator.

The general syntax of a multi-leveled package statement is:

```
package pkg_name1 . pkg_name2 . pkg_name3;
```

Eg

package   mypack1 . mypack2 . mypack3;

To create a User defined package

```
package p1;
import java.util.*;
public class Person
{
        int id;
        String name;
        Scanner scan=new Scanner(System.in);
        public void set_data()
        {
                System.out.println("Enter the Id:");
                id=scan.nextInt();
                System.out.println("Enter the Name:");
                name=scan.next();
        }
        public void display()
        {
                System.out.println(id+"\t"+name);
        }
}
```

Compile java package

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

```
javac -d . javafilename
```

Eg:

```
Javac –d . Person.java
```

Importing Package / How to access package from another package?

Eg 1:

```
import p1.Person;
 class Student
 {
 public static void main(String args[])
        {
                Person p=new Person();
                p.set_data();
                p.display();
        }
 }
```

Eg2.

```
class Student
	{
		public static void main(String args[])
		{
			p1.Person p=new p1.Person();
			p.set_data();
			p.display();
		}
	}
```

Create a hierarchy of packages / Sub-package in java

Package inside the another package is called the sub-package. It should be created to categorize the package further.

```
	package mypack.p1;
	import java.util.*;
	public class Person
	{
		int id;
		String name;
		Scanner scan=new Scanner(System.in);
		public void set_data()
		{
```

```
                System.out.println("Enter the Id:");
                id=scan.nextInt();
                System.out.println("Enter the Name:");
                name=scan.next();
        }
        public void display()
        {
                System.out.println(id+"\t"+name);
        }
}
```

Importing Package

```
import mypack.p1.Person;
 class Student
 {
 public static void main(String args[])
        {
                Person p=new Person();
                p.set_data();
                p.display();
        }
}
```

// Demonstrate the use of package in inheritance

```
class Employee extends mypack.p1.Person
{

  Public static void main(String args[])
  {
       Employee emp=new Employee();
        emp.set_data();
        emp.display();
  }

}
```

Advantage of Java Package

1] Make easy searching or locating of classes and interfaces.
2] Java package is used to categorize the classes and interfaces so that they can be easily maintained.

3] Java package provides access protection.

4] Java package removes naming collision.

5] Reuse the classes contained in the packages of other programs.

6] Implement data encapsulation (or data-hiding).

Q: created a class Calculator with the following function i.e add(), subs(), multi() and div() inside a package name letmecalculate. Create a menu driven program in Test class and use the class Calculator in the Test class.