Java Packages

Description

A **java package** is

- Collection of alike types of classes, interfaces and sub-packages.
- Classified into two types, java built-in packages and user-defined packages.
- Java built-in packages samples – java.lang, java.awt, java.swing, java.net, etc.

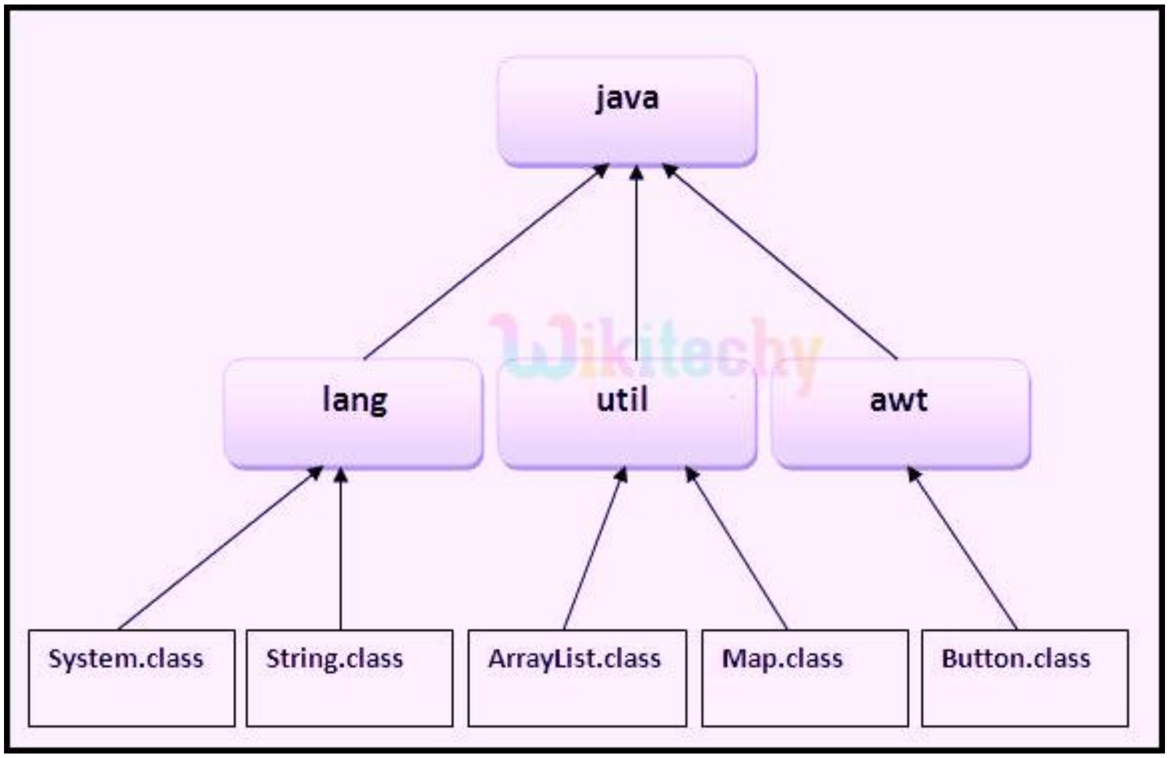


Fig1. java in-built packages

Advantage of Java Package:

- Classifies the classes and interfaces for its perfect maintenance.
- Access protection is highly available in java package.
- Eradicates naming conflict in a well-trained way.

Using Existing Package:

- To use an existing package in a java code, package declaration must be implemented as follows: Below are the steps to perform the same perfectly:

  - Use import keyword as the first word in the java program.
  - Call the package with fully qualified name

import java.awt.event;

- The above package import statement becomes the first line of code in the java program that uses it. Here, it imports all classes from the **java.awt.event package**.
- When there is a need for a single class within the package, mention it explicitly:

import java.awt.event.ActionEvent;

- The above code imports only the ActionEvent class from the **java.awt.event** package.

- When complete package content is required, import all the classes

  import java.awt.*;

- When there is a need for static members of a package class, imply static import as follows:

  import static java.lang.Math.sqrt;

Creating a package:

- Creating a java package implements the keyword package as the first word in the java code as follows:

Syntax:

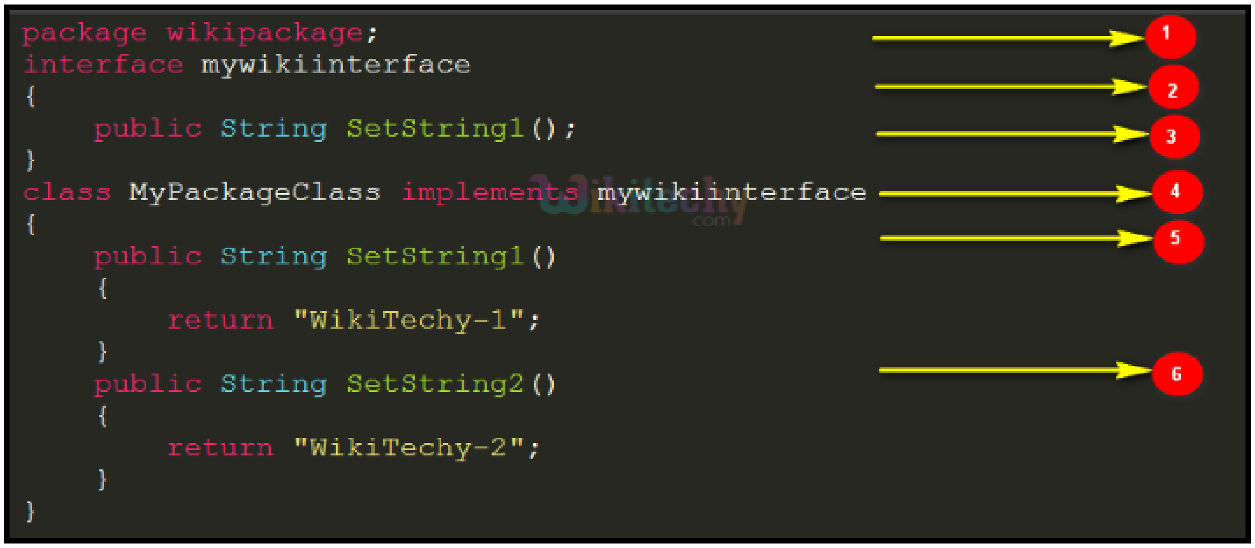
```
package [package name];
interface [interface name]
{
        \\ variables and methods optional
}
class [class name1] implements [package name]
{       }

\\ Package call
class [class name2]       {
        public static void main(String[] args)       {
        \\ create object for the interface
         [interface name] [object name] = new [class name1]
        \\ call interface variables and objects using the method
         [class name1] [object name] = new [class name1]

        \\ call class name1's variables and objects using the method
    }
```

Sample Code – Create Package:

```
package wikipackage;
interface mywikiinterface
{
        public String SetString1();
}

class MyPackageClass implements mywikiinterface
{
        public String SetString1()
        {
                return "WikiTechy-1";
        }
        public String SetString2()
        {
                return "WikiTechy-2";
        }
}
```

```
public class HelloWorldPackage
{
    public static void main(String[] args)
    {
        System.out.println("\n\n WikiTechy - Welcome to Java Packages");

        mywikiinterface objIntrfc = new MyPackageClass();
        System.out.println("\n" + objIntrfc.SetString1());

        wikipackage.MyPackageClass  objHello  =  new
        wikipackage.MyPackageClass();
        System.out.println("\n" + objHello.SetString2());
    }
}
```

Code Explanation:



```
package wikipackage;
interface mywikiinterface
{
    public String SetString1();
}
class MyPackageClass implements mywikiinterface
{
    public String SetString1()
    {
        return "WikiTechy-1";
    }
    public String SetString2()
    {
        return "WikiTechy-2";
    }
}
```

1. A package is created with a package name as package wikipackage;

2. An interface is created under the package as interface mywikiinterface.

3 A method signature is defined under the interface as public String SetString1(); Class which implements the interface can present a definition for this method.

4 Now a class is defined under the package as class MyPackageClass implements mywikiinterface

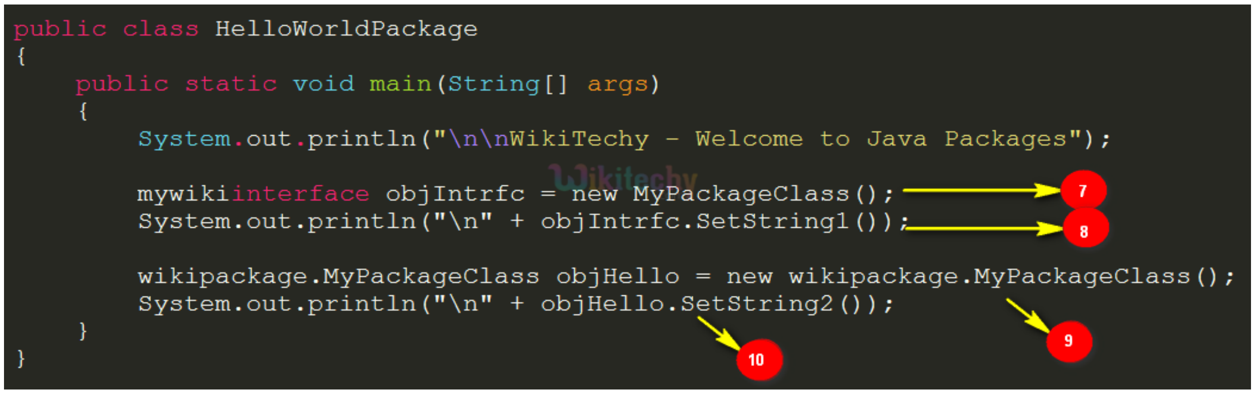
5 Interface method is implemented in this class with the code

```
public String SetString1()
{
        return "WikiTechy-1";
}
```

6 Again a new public method is defined in the class that implements

this interface. Below follows the code:

```
public String SetString2()
{
        return "WikiTechy-2";
}
```



```
public class HelloWorldPackage
{
    public static void main(String[] args)
    {
        System.out.println("\n\nWikiTechy - Welcome to Java Packages");

        mywikiinterface objIntrfc = new MyPackageClass();
        System.out.println("\n" + objIntrfc.SetString1());

        wikipackage.MyPackageClass objHello = new wikipackage.MyPackageClass();
        System.out.println("\n" + objHello.SetString2());
    }
}
```

7 Now in the class that contains main method, an object is created for the interface of type “MyPackageClass” as follows:

```
mywikiinterface objIntrfc = new MyPackageClass();
```

8 A call is made to the method “SetString1()” as `System.out.println("\n" + objIntrfc.SetString1());`

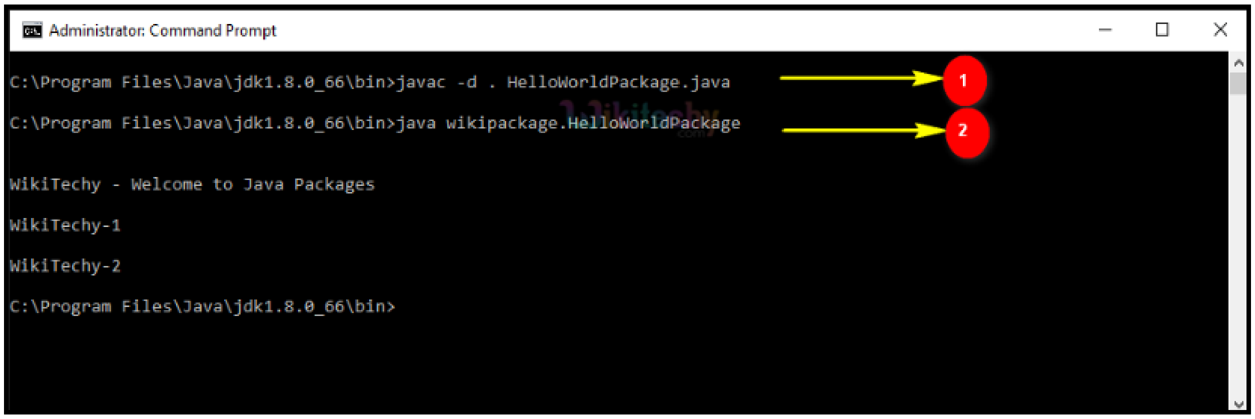
9 An object is created for the class “MyPackageClass”.

```
wikipackage.MyPackageClass objHello = new wikipackage.MyPackageClass();
```

10 A call is made to the method “SetString2()” as

```
System.out.println("\n" + objHello.SetString2());
```

Output:



```
C:\Program Files\Java\jdk1.8.0_66\bin>javac -d . HelloWorldPackage.java

C:\Program Files\Java\jdk1.8.0_66\bin>java wikipackage.HelloWorldPackage

WikiTechy - Welcome to Java Packages

WikiTechy-1

WikiTechy-2

C:\Program Files\Java\jdk1.8.0_66\bin>
```

1 Compiling java package:

Syntax:

javac -d [directory] [javafilename]

Eg. javac -d . Simple.java

- The -d mentions the target place for the class file. Any folder name like D:/aa can be used. When the class file needs to be in the same folder,"." (dot) is used as shown above.

Hence to Compile: **javac -d . HelloWorldPackage.java**

- If in case to store the class in another folder, use the following way:

```
javac -d C:\aa HelloWorldPackage.java
```

Classpath:

- An environmental variable that contains the path for the default-working directory of java program is called classpath (.). On mention of classpath java compiler will believe it as the root of that package hierarchy.

- And hence before running this class file, follow the below way to set the classpath for the class file in command prompt as:

```
set classpath = C:\aa
(or directly mention classfile path is running prompt)
  java -classpath c:\aa wikipackage.HelloWorldPackage
```

2 To Run use the following syntax:

```
 java [package name]. [class file name]

Eg. java wikipackage. HelloWorldPackage
```

Sample Code – Import Package:

1. File: MyDemo.java

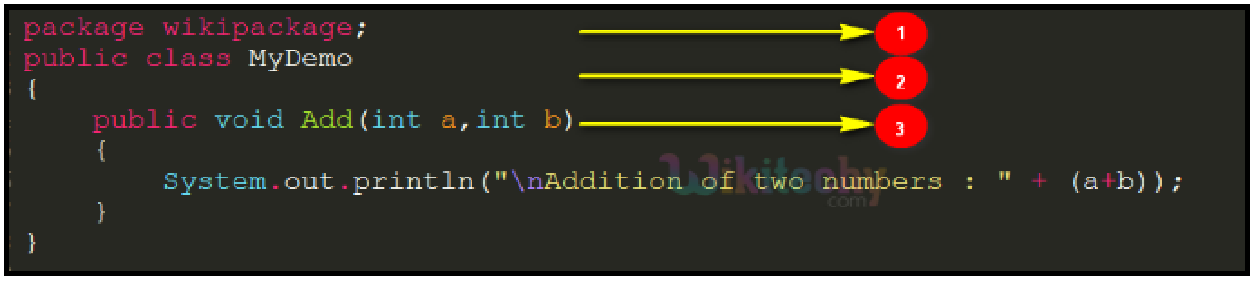
```
package wikipackage;
public class MyDemo
{
        public void Add(int a, int b)
        {
        System.out.println("\nAddition of two numbers: "
+ (a+b));
        }
}
```

2. File: wikiclass.java

```
import wikipackage.MyDemo;
import java.lang.Math;

class wikiclass extends MyDemo
{
        public static void main (String args[])
        {
System.out.println("\n\n WikiTechy - Working With Package
     Import\n");
                int a = 20, b = 10, c = 144;
                System.out.println("Input Values: a=20, b=10");
                wikiclass obj=new wikiclass();
                obj.Add(a, b);
                System.out.println("\nSquare Root of c=144: " +
Math.sqrt(c));
        }
}
```

Code Explanation:

✹ File: MyDemo.java

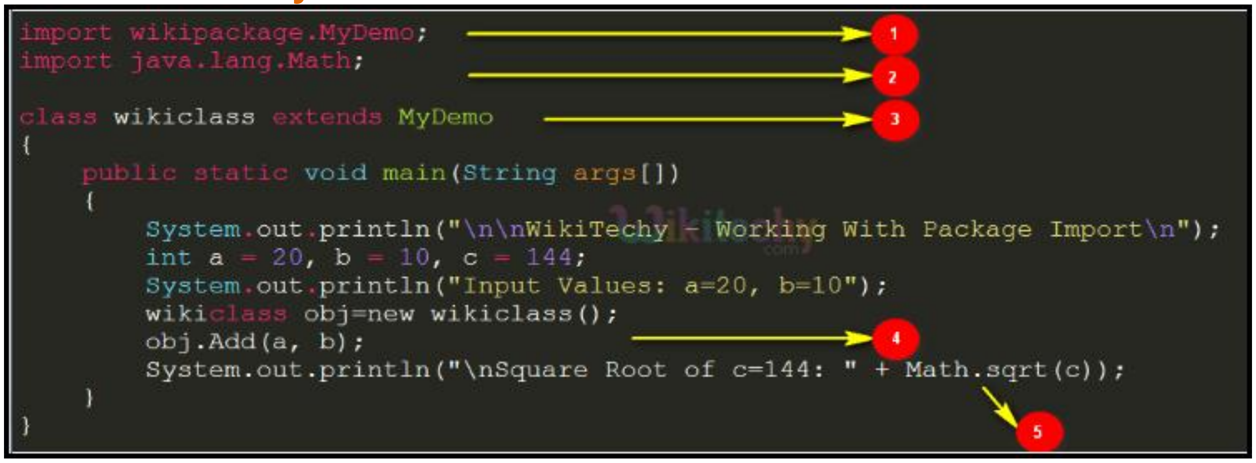


```
package wikipackage;
public class MyDemo
{
     public void Add(int a,int b)
     {
          System.out.println("\nAddition of two numbers : " + (a+b));
     }
}
```

1. Package created with the name wikipackage.
2. A class is created under the package named “MyDemo”.
3. The class defines a public method “Add” to add two integers as

```
public void Add(int a, int b)
{
        System.out.println("\nAddition of two numbers: " + (a+b));
}
```

File: wikiclass.java



```
import wikipackage.MyDemo;
import java.lang.Math;

class wikiclass extends MyDemo
{
    public static void main(String args[])
    {
        System.out.println("\n\nWikiTechy - Working With Package Import\n");
        int a = 20, b = 10, c = 144;
        System.out.println("Input Values: a=20, b=10");
        wikiclass obj=new wikiclass();
        obj.Add(a, b);
        System.out.println("\nSquare Root of c=144: " + Math.sqrt(c));
    }
}
```

1 Import of user-defined package “wikipackage.MyDemo”.

2 Import of java built-in package “java.lang.Math”.

3 Extending properties from user-defined package class via the code:

class wikiclass extends MyDemo

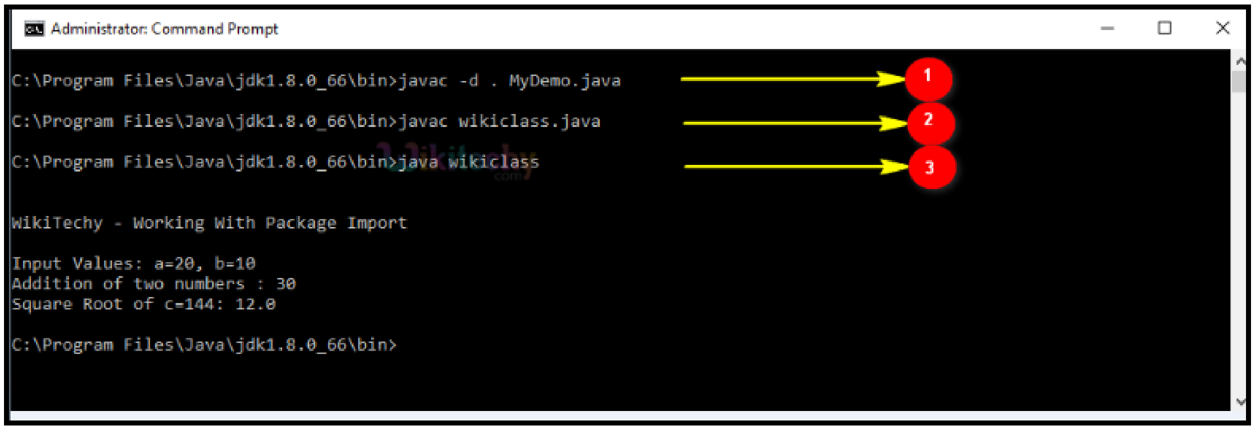
4 Creating object to call the method in package class.

wikiclass obj=new wikiclass();
obj.Add(a, b);

5 Calling method from java in-built package.

System.out.println("\nSquare Root of c=144: " + Math.sqrt(c));

Output:



```
C:\Program Files\Java\jdk1.8.0_66\bin>javac -d . MyDemo.java

C:\Program Files\Java\jdk1.8.0_66\bin>javac wikiclass.java

C:\Program Files\Java\jdk1.8.0_66\bin>java wikiclass


WikiTechy - Working With Package Import

Input Values: a=20, b=10
Addition of two numbers : 30
Square Root of c=144: 12.0

C:\Program Files\Java\jdk1.8.0_66\bin>
```

1. Compiling java package class file as javac –d . MyDemo.java. As discussed earlier, "-d ." eases java compiler to place the class file of package class in the same folder, so that it can be used by the java programs that imports the package, which as well exist in the same folder.

2. Compiling the class that imports the java package class as javac wikiclass.java.

3. Run the class file using java jre to display the output in the command prompt window as java wikiclass.