

Package

Q What is the Package?

Package is collection of classes and interfaces in java

Q Why Use Package or what is the benefit of package?

The major use of package is reuse the bundle of classes as well as it is used for create the distribution or reusable component using java.

Q how to create the package and Reuse the Package using java?

If we want to create the package in java we have the some important steps.

i) Declare the package: if we want to declare the package in java we have the package keyword.

syntax:

package packagename;

e.g package org.techhub;

ii) Declare the classes under the package:

Once we declare the package we can declare the number of classes in package as well as declare the interface as member under the package.

syntax:

```
_____
package packagename;
access specifier class classname
{
    access specifier returntype functionname(datatype variablename)
    {
    }
}
```

```
package org.techhub;
public class Table
{
    public void showTable(int x)
    {
        for(int i=1; i<=10; i++)
        {
            System.out.printf("%d X %d = %d\n",i,x,i*x);
        }
    }
}
```

Once we create package and declare the classes in package then we can compile the package

If we want to compile class in which we declare the package we have the command like as

javac -d . filename.java

-d means create the folder according to package name mention in file and store the all .class files in it.

Note: when we want to compile your package or java program from any another location except the jdk\bin then we need to set the jdk\bin folder path in my computer enviourmental variable.

How to set the jdk\bin folder in My Computer Enviourmental variable

If we want to set the path of jdk\bin folder in mycomputer enviourmental variable we have the some steps.

Right click on my computer → click on advanced system setting → choose advance option → click on enviourmental variable → system variable → select path variable → click on edit button → click on new button → paste the jdk\bin folde where it is present.

Once we set the jdk\bin folder path then we can compile the java program from any location in your machine

3)Compile the package program: if we want to compile the package program then we have the command given below

javac -d . filename.java

e.g `javac -d . Table.java`

as per our example when we execute this command then java automatically create the package name as `org.techhub` and store them all `.class` file in it.

Once we create the package then we can reuse the package more than one time.

How to Reuse the Package

If we want to reuse the package in java we have the some important steps.

i) **import the created package:** if we want to import package we have the `import` keyword.

syntax

`import` packagename;

e.g `import org.techhub;`

ii) **Create the object of class which we want use:**

Example

```
import org.techhub.Table;
public class UseTableApplication
{
    public static void main(String x[])
    {
        Table t = new Table();
    }
}
```

iii) Call the member from package which we want to use

```
import org.techhub.Table;
public class UseTableApplication
{
    public static void main(String x[])
    {
        Table t = new Table();

        t.showTable(5);

    }
}
```

```
D:\batchdevyani>javac TableApplication.java
```

```
D:\batchdevyani>java TableApplication
```

```
1 X 5   = 5
2 X 5   = 10
3 X 5   = 15
4 X 5   = 20
5 X 5   = 25
6 X 5   = 30
7 X 5   = 35
8 X 5   = 40
9 X 5   = 45
10 X 5  = 50
```

Rules of Access Specifier With package

public: if we declare the member as public in package then member can access inside of package as well as outside of package.

default: if we not specify any access specifier with member then java use the default access specifier and if declare any member as default then member can access only side of package not outside of package.

private : if we declare any member as private then member cannot access outside class as well as outside of package also.

protected : protected member can access within package as well as outside of package but only within child class.

Example

```
package org.techhub;  
public class Table  
{  
    protected void showTable(int x)  
    {  
        for(int i=1; i<=10; i++)  
        {  
            System.out.printf("%d X %d = %d\n",i,x,i*x);  
        }  
    }  
}
```

```
import org.techhub.Table;
class MyTable extends Table
{
    void display()
    {
        showTable(5);
    }
}
public class TableApplication
{
    public static void main(String x[])
    {
        MyTable m = new MyTable();
        m.display();
    }
}
```


