

Website

<https://www.curiousinfotecholutions.com>

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



Java Lab

Java Programming Day-16



Email id:
info@curiousinfotecholutions.com

Syllabus Day-16

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



- Example of Java Package
- Access Package from Another Package
- Access Modifiers
- Justify Private Access Modifier
- Justify Default Access Modifier
- Justify Protected Access Modifier
- Justify Public Access Modifier



ExpPkg-1 Display *My Package*

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



- **Create** a sub-directory named
 - ExpPackage
- **Create** a Class Named as
 - Simple>notepad Simple.java
- **Save** Class Simple as Simple.java in sub-dir ExpPackage
- **Compile** Simple.java
 - javac -d . Simple.java
- **Check** sub-dir *mypack* and *Simple1.class*
- **Run** Package
 - java mypack.Simple1 MyPackage

- Step-1: //save as Simple.java

```
1. package mypack; //user defined pkg
2. public class Simple{
3.     public static void main(String args[]){
4.         System.out.println("My Package");
5.     }
6. }
```



ExpPkg-2 Display *My Package* through command-line argument

- **Create** a Class Named as
 - Simple1
- **Save** Class Simple as Simple.java in sub-dir ExpPackage
- **Compile** Simple1.java
 - javac -d . Simple1.java
- **Check** sub-dir mypack and Simple1.class
- **Run** Package
 - java mypack.Simple1 My Package

My
Package

- Step-1: //save as Simple.java
 1. **package** mypack; //user defined pkg
 2. **public class** Simple{
 3. **public static void** main(String args[]){
 4.
 1. for (String s: args)
 1. System.out.println(s); }
 - 5.}

ExpPkg-3

Read Command Line arguments as 10 20 30 and display argu and sum.

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



- **Create** a Class Named as
 - Simple2
- **Save** Class Simple as Simple2.java in sub-dir ExpPackage
- **Compile** Simple2.java
 - javac -d . Simple2.java
- **Check** sub-dir mypack and Simple2.class
- **Run** Package
 - java mypack.Simple2

```
• Step-1: //save as Simple.java
1. package mypack;//user defined pkg
2. public class Simple2{
3.     public static void main(String args[]){
4.         int sum=0;
5.         for (String s: args)
6.         {
7.             sum = sum + Integer.parseInt(s);
8.             System.out.println(s);
9.         }
10.        System.out.println("Sum="+sum);
11.    }
12.}
```

10
20
30
Sum = 60

ExpPkg-4: Access Package from Another Package



- **Create** a Class Named as **A**
- **Save** Class A as **A.java** in sub-dir ExpPackage
- **Compile** A.java as ***javac -d . A.java***
- **Check** sub-dir pack and A.class
- **Change** sub-dir from ExpPackage to pack
- **Create** a Class Named as **B**
- **Save** Class B as **B.java** in sub-dir pack
- **Again change** sub-dir from pack to ExpPackage
- **Compile** A.java as ***javac -d . Pack/B.java***
- **Now check** B.class in sub-dir mypack
- **Lastly, Run** B.class as from ExpPackage
 - *java mypack/B*

- Step-1: *//save by A.java*

```
1. package pack;  
2. public class A{  
3.     public void msg()  
4.     {System.out.println("Hello Hello"); }  
5. }
```

```
//Step-3 //save by B.java in pack Compile pkg mypack  
package mypack;  
import pack.*;  
class B{  
    public static void main(String args[]){  
        A obj1 = new A();  
        obj1.msg();  
    }  
}
```

ExpPackage/JavaLAB
A.java

Pack
A.class
B.java

mypack
B.class

ExpPkg-5: Access Package from Another Package

import the package. classname

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



- Create a Class Named as **A**
- Save Class A as **A.java** in sub-dir ExpPackage
- Compile A.java as **javac -d . A.java**
- Check sub-dir pack and A.class
- Run Package
 - java mypack.Simple1
- Change sub-dir from ExpPackage to pack
- Create a Class Named as **B**
- Save Class B as **B.java** in sub-dir pack
- Again change sub-dir from pack to ExpPackage
- Compile A.java as **javac -d . Pack/B.java**
- Now check B.class in sub-dir mypack
- Lastly, Run B.class as
 - java mypack/B



- Step-1: //save by A.java
 1. package pack;
 2. public class A{
 3. public void msg()
 4. {System.out.println("Hello Hello"); }
 5. }

```
//Step-3 //save by B.java in pack Compile pkg mypack
package mypack;
import pack.A; //<- Only Change is here
class B{
    public static void main(String args[]){
        A obj1 = new A();
        obj1.msg();
    }
}
```

ExpPkg-6: Access Package from Another Package

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

PDF

- **Create** a Class Named as **A** , **B**
- **Save** Class A as **A.java** and A1.java in sub-dir ExpPackage
- **Compile**
 - A.java as **javac -d . A.java**,
 - A1.java as **javac -d . A1.java**
- Check sub-dir pack and A.class and A1.class
- Change sub-dir from ExpPackage to pack
- Create a Class Named as **B**
- Save Class B as **B.java** in sub-dir pack
- Again change sub-dir from pack to ExpPackage
- Compile A.java as **javac -d . Pack/B.java**
- *Now check B.class in sub-dir mypack*
- *Lastly, Run B.class as*
 - **java mypack/B**

Hello Hello
Java Java



• Step-1: //save by A.java

```
1. package pack;  
2. public class A{  
3.     public void msg()  
4.     {System.out.println("Hello Hello"); }  
5. }
```

• Step-1: //save by A1.java

```
1. package pack;  
2. public class A1{  
3.     public void msg()  
4.     {System.out.println("Java Java"); }  
5. }
```

```
//Step-3 //save by B.java in pack Compile pkg mypack  
package mypack;  
import pack.*; //<- Only Change is here  
class B{  
    public static void main(String args[]){  
        A obj1 = new A();  
        A1 obj2 = new A1();  
        obj1.msg();//Hello Hello  
        obj2.msg();//Java Java  
    }  
}
```

ExpPkg-6: Access Package from Another Package

import the fully qualified name

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



- Create a Class Named as **A**
- Save Class A as **A.java** in sub-dir ExpPackage
- Compile A.java as **javac -d . A.java**
- Check sub-dir pack and A.class
- Run Package
 - java mypack.Simple1
- Change sub-dir from ExpPackage to pack
- Create a Class Named as **B**
- Save Class B as **B.java** in sub-dir pack
- Again change sub-dir from pack to ExpPackage
- Compile A.java as **javac -d . Pack/B.java**
- Now check B.class in sub-dir mypack
- Lastly, Run B.class as
 - java mypack/B

• Step-1: //save by A.java

```
1. package pack;  
2. public class A{  
3.     public void msg()  
4.     {System.out.println("Hello Hello"); }  
5. }
```

```
//Step-3 //save by B.java in pack Compile pkg mypack  
package mypack;  
//import pack.A; //<- First Change is here  
class B{  
    public static void main(String args[]){  
pack.A obj1 = new pack.A(); //<- Second Change is here  
        obj1.msg();  
    }  
}
```

Understanding Java Access Modifiers

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



Access Modifier	within class	within package	outside package by subclass only	outside package
Private	Y	N	N	N
Default	Y	Y	N	N
Protected	Y	Y	Y	N
Public	Y	Y	Y	Y