

Implementation of Packages in Java

Aim:

Write a Java program to implement built-in, user-defined packages and accessing all classes in a package.

PRE LAB EXERCISE

QUESTIONS

1. What is java.util package and what collection framework does it contain?
The java.util package provides utility classes for data structures, date/time, and more. It contains the Java Collection Framework (JCF) for storing and manipulating data. JCF includes interfaces like List, Set, Queue and classes like ArrayList, HashMap. It makes programming easier by offering reusable data structures and algorithms.
2. What are the two types of packages in Java?

Java has built-in packages provided by the language, like java.util and java.io. It also has user-defined packages created by programmers for organizing code. Packages help avoid naming conflicts and improve code modularity. They make code easier to maintain and reuse across programs.

IN LAB EXERCISE

Objective

To understand and implement the concepts of built-in, user-defined packages and accessing all classes in a package in Java.

Built-in Packages comprise a large number of classes that are part of the Java API. Some of the commonly used built-in packages are:

- **java.lang:** Contains language support classes(e.g, classes that define primitive data types, math operations). This package is automatically imported.
- **java.io:** Contains classes for supporting input/output operations.
- **java.util:** Contains utility classes that implement data structures such as Linked Lists and Dictionaries, as well as support for date and time operations.
- **java.applet:** Contains classes for creating Applets.
- **java.awt:** Contains classes for implementing the components for graphical user interfaces (like buttons, menus, etc).

Source Code

```

import java.util.Random; // built-in package

public class Sample{

    public static void main(String[] args) {
        // using Random class

        Random rand = new Random();

        // generates a number between 0–99
        int number = rand.nextInt(100);

        System.out.println("Random number: " + number);

    }
}

```

Output

Random number: 59

The screenshot shows the Eclipse IDE interface. At the top, there are tabs for 'DisplayInfo.java', 'Main.java', and 'Sample.java'. The 'Sample.java' tab is active. Below the tabs is the Java code for the 'Sample' class. In the code editor, line 3 ('import java.util.Random;') has a blue vertical bar under it, indicating it's currently selected or being edited. The code itself is as follows:

```

1 package programming.java.com;
2
3 import java.util.Random;
4 public class Sample{
5     public static void main(String[] args) {
6         // using Random class
7         Random rand = new Random();
8         // generates a number between 0–99
9         int number = rand.nextInt(100);
10        System.out.println("Random number: " + number);
11    }
12 }

```

Below the code editor is the Eclipse IDE toolbar with icons for Problems, Javadoc, Declaration, Console, and Eclipse IDE for. The 'Console' tab is active. The console output window shows the message '<terminated> Sample [Java Application] D:\eclipse\plugins\org.eclipse.justj.open'. Underneath this, the text 'Random number: 59' is displayed, which is the output of the program.

User-defined Packages are the packages that are defined by the user.

Source code

```

package com.myapp;

public class Helper {

    public static void show() {
        System.out.println("Hello from Helper!");
    }
}

```

```
}
```

==To use this in another class==

```
import com.myapp.Helper;  
public class Test {  
    public static void main(String[] args) {  
        Helper.show();  
    }  
}
```

Output:

Hello from Helper!

The screenshot shows the Eclipse IDE interface. At the top, there are tabs for 'DisplayInfo.java', 'Main.java', 'Sample.java', and 'Test.java' (which is currently selected). Below the tabs is a code editor with the following content:

```
1 package programming.java.com;  
2  
3  
4 public class Test {  
5     public static void main(String[] args) {  
6         Helper.show();  
7     }  
8 }  
9  
10
```

At the bottom of the screen, the 'Console' tab is active, showing the output of the program:

```
<terminated> Test [Java Application] D:\eclipse\plugins\org.eclipse.jdt.core\src\com\myapp\Helper.java  
Hello from Helper!
```

//Importing all classes from a package.

Source code

```
import java.util.Vector;  
  
public class Coders {  
    public Coders() {  
        // java.util.Vector is imported, We are able to access it directly in our code.  
        Vector v = new Vector();  
        java.util.ArrayList l = new java.util.ArrayList();  
        l.add(3);
```

```

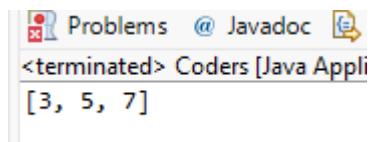
        l.add(5);
        l.add(7);
        System.out.println(l);
    }

    public static void main(String[] args) {
        new Coders();
    }
}

```

Output

[3,5,7]



POST LAB EXERCISE

- What will happen if two classes in different packages have the same name and are imported in a Java file?
The compiler will get confused if both classes are imported using * and have the same name. This will cause a naming conflict and a compilation error.
To resolve it, you must use the fully qualified class name with the package.
Example: java.util.Date d; java.sql.Date s;
- What is the purpose of using packages in Java?
Packages organize classes and interfaces into namespaces for better structure.
They avoid naming conflicts between classes with the same name.
Packages make code modular, reusable, and easier to maintain.
They also provide access control with public, private, and protected modifiers.
- Which built-in Java package would you use if you want to create a GUI window and display a message?
A. java.util
B. java.sql
C. java.awt
D. java.net

C. java.awt

The java.awt package provides classes for creating GUI components like windows, buttons, and dialogs.

ASSESSMENT

Description	Max Marks	Marks Awarded
Pre Lab Exercise	5	
In Lab Exercise	10	
Post Lab Exercise	5	
Viva	10	
Total	30	
Faculty Signature		