Experiment Number : 8 **Date: 25/02/2026**

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?
   java.util is a built-in Java package that provides utility classes and the **Collection Framework** for storing and manipulating data.
   It contains:

- **List** → ArrayList, LinkedList
- **Set** → HashSet, TreeSet
- **Queue** → PriorityQueue
- **Map** → HashMap, TreeMap

2. What are the two types of packages in Java?

   The two types of packages in Java are:

1. **Built-in Packages**
2. **User-defined Packages**

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.

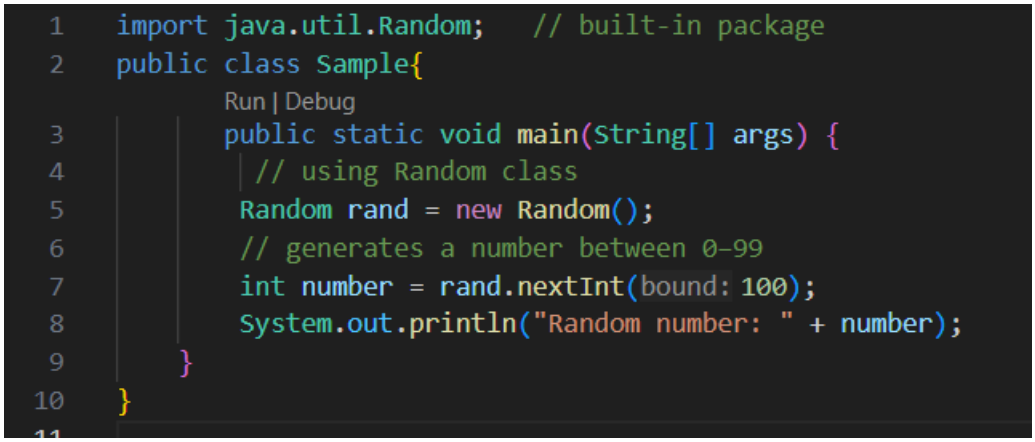
Built-in Packages comprise a large number of classes that are part of the Java API. Some of the commonly used built-in packages are:

- 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: 49

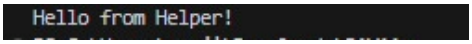
```
Random number: 88
```

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:



//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();
   }
}
```

```java
import java.util.Vector;
public class Coders {
    public Coders() {
        // java.util.Vector is imported, We are able to acce
        Vector v = new Vector();
        java.util.ArrayList l = new java.util.ArrayList();
        l.add(e: 3);
        l.add(e: 5);
        l.add(e: 7);
        System.out.println(l);
    }
    Run | Debug
    public static void main(String[] args) {
        new Coders();
    }
}
```

Output

```
[3, 5, 7]
```

POST LAB EXERCISE

1. What will happen if two classes in different packages have the same name and are imported in a Java file?

   If two classes with the **same name** from different packages are imported, Java will show a **compile-time ambiguity error** because it cannot decide which class to use.
   Solution:
   Use the **fully qualified class name** (complete package path).

2. What is the purpose of using packages in Java?

   Packages are used to:

- Organize large programs properly
- Avoid class name conflicts
- Improve code reusability
- Provide security (access control)
- Make maintenance easier

3. 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
Reason (brief):
java.awt provides classes to create GUI components like windows, buttons, labels, 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		