# PRACTICAL 4

**Aim**: To understand packages and access modifiers inJava programming Language

**Prerequisite:**

* Knowledge of basic constructs in Java programming language.

**Outcome:** After successful completion of this experiment students will be able to,

* Create packages and use them within other classes
* Understand the usage of Public, Default and Private modifiers

**Theory:**

A set of classes and interfaces grouped together are known as Packages in JAVA. Every class is a part of a certain package. When you need to use an existing class, you need to add the package within the Java program.

The benefits of using Packages are as follows:

* The packages organize the group of classes into a single unit
* It will control the naming conflicts
* The access protection will be easier. Protected and default are the access level control to the package
* Easy to locate the related classes
* Reuse the existing classes in packages

In Java, it divides packages into two types: Built-in Packages (packages from the Java API) and User-defined Packages are the two types of packages (create your own packages).

When we install Java on a personal computer or laptop, many packages are automatically installed. Each of these packages is unique and capable of handling various tasks. This eliminates the need to build everything from scratch. Here are some examples of built-in packages:

java.lang

java.io

java.util

**Using built in packages:**

import java.lang.\*;

public class Example {

public static void main(String[] args) {

double radius = 5.0;

double area = Math.PI \* Math.pow(radius, 2);

System.out.println("Area: " + area);

String message = "Hello, World!";

int length = message.length();

System.out.println("Length: " + length);

}

}

User-defined packages are those that developers create to incorporate different needs of applications. In simple terms, User-defined packages are those that the users define. Inside a package, you can have Java files like classes, interfaces, and a package as well (called a sub-package).

Sub-package - **A package defined inside a package is called a sub-package.** It’s used to make the structure of the package more generic. It lets users arrange their Java files into their corresponding packages.

**Using User defined packages:**

package packageA;

class Demo {

public void display() {

System.out.println("My first package");

}

}

package packageB;

import packageA.\*;

class B {

public static void main(String args[]) {

// display class is present inside packageA

Demo d=new Demo();

d.display();

}}

**Access Modifiers:**

Access modifiers are used for setting the access level to classes, variables, methods, and constructors. The **four access modifiers are** *public*, *private*, *protected* and *default* (no keyword).

**Default:**When we don’t use any keyword explicitly, Java will set a default access to a given class, method or property. The default access modifier is also called package-private, which means that all members are visible within the same package but aren’t accessible from other packages:

package com.baeldung.accessmodifiers;

public class SuperPublic {

static void defaultMethod() {

...

}

}

defaultMethod() is accessible in another class of the same package:

**Public:**If we add the public keyword to a class, method or property then we’re making it available to the whole world, i.e. all other classes in all packages will be able to use it. This is the least restrictive access modifier:

package com.baeldung.accessmodifiers;

public class SuperPublic {

public static void publicMethod() {

...

}

}

publicMethod() is available in another package:

**Private:**Any method, property or constructor with the private keyword is accessible from the same class only. This is the most restrictive access modifier and is core to the concept of encapsulation. All data will be hidden from the outside world:

package com.baeldung.accessmodifiers;

public class SuperPublic {

static private void privateMethod() {

...

}

private void anotherPrivateMethod() {

privateMethod(); // available in the same class only.

}

}

(TO BE COMPLETED BY STUDENTS)

|  |  |
| --- | --- |
| Roll No. K005 | Name: Jal Bafana |
| Class: Btech. Cyber Security (Sem-4) | Batch: K1 |
| Date of Practical: 17.02.2025 | Date of Submission: 17.02.2025 |
| Grade: |  |

1. Create a package and implement a class within it having appropriate constructors and one function to add 2 numbers and one more function to display the result.

package newproject;

public class Newproject {

private int num1;

private int num2;

private int result;

public Newproject(int n1, int n2)

{

num1 = n1;

num2 = n2;

}

public void add()

{

result = num1 + num2;

}

public void displayResult()

{

System.out.println("The sum of " + num1 + " and " + num2 + " is " + result);

}

}

package newpackage2;

import newproject.Newproject;

public classNewClass2

{

public static void main(String[] args)

{

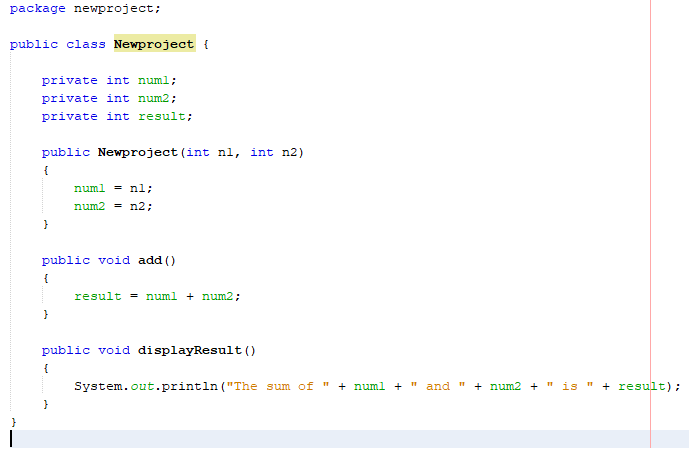
Newprojectpgm = new Newproject(10, 20);

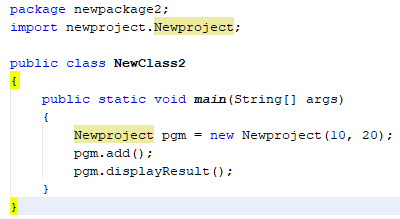
pgm.add();

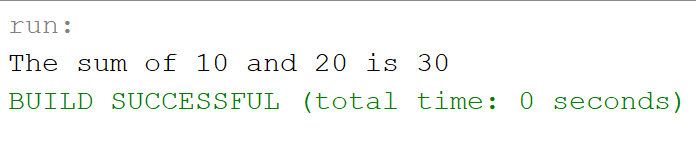
pgm.displayResult();

}

}







1. Create a new package having a class to access the add and display functions created earlier.

Change the access modifiers in the previous question and demonstrate the usage of “public”, private” and “default” modifiers.

package newproject;

public class Newproject {

private int num1;

private int num2;

private int result;

public Newproject(int n1, int n2)

{

num1 = n1;

num2 = n2;

}

public void add()

{

result = num1 + num2;

}

void displayResult()

{

System.out.println("The sum of " + num1 + " and " + num2 + " is " + result);

}

}

package newpackage2;

import newproject.Newproject;

public classNewClass2

{

public static void main(String[] args)

{

Newprojectpgm = new Newproject(10, 20);

pgm.add();

pgm.displayResult();

}

}

1. Create a class with minimum of 2 member functions and 2 data members. Create one more class within the same file to access the functions defined in the previous class. Demonstrate the usage of public, default and private access specifier.

