|  |
| --- |
| Experiment No. 5 |
| Implement a program on Packages. |
| Date of Performance: |
| Date of Submission: |

**Aim:** To use packages in java.

**Objective:** To use packages in java to use readymade classes available in them using square root method in math class.

## Theory:

A java package is a group of similar types of classes, interfaces and sub-packages. Packages are used in Java in order to prevent naming conflicts, to control access, to make searching/locating and usage of classes, interfaces, enumerations and annotations easier, etc.

There are two types of packages-

1. Built-in package: The already defined package like java.io.\*, java.lang.\* etc are known as built-in packages.
2. User defined package: The package we create for is called user-defined package.

Programmers can define their own packages to bundle group of classes/interfaces, etc. While creating a package, the user should choose a name for the package and include a package statement along with that name at the top of every source file that contains the classes, interfaces, enumerations, and annotation types that you want to include in the package. If a package statement is not used then the class, interfaces, enumerations, and annotation types will be placed in the current default package.

**Code:**

import java.util.\*;

// Main Class

class demo1 {

// Main driver method

public static void main(String[] args)

{

// Scanner to take input from the user object

Scanner myObj = new Scanner(System.in);

String userName;

// Display message

// Enter Your Name And Press Enter

System.out.println("Enter Your Name:");

// Reading the integer age entered using

// nextInt() method

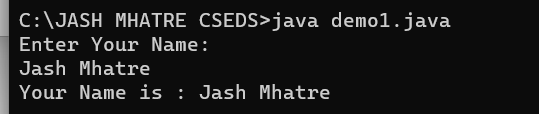
userName = myObj.nextLine();

// Print and display

System.out.println("Your Name is : " + userName);

}

}



## Conclusion:

packages in Java provide a way to structure and organize your code, manage namespaces, and control access to classes and members. They are a fundamental part of Java's modularity and encapsulation features, making your code more organized, maintainable, and reusable.