**Batch: C3 Roll No.: 16010123217**

**Experiment / assignment / tutorial No. 9**

**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of the Staff In-charge with date**

|  |
| --- |
| **TITLE :Java Packages** |

**AIM:** Create a package ‘myPackage’ which contains a class myMath. The class contains following static methods.

i)  power (x, y) – to compute xy

ii) fact (x) – to compute x!

Write a program to find the following series.

cos (x) = 1 – (x2/2!) + (x4/4!) – (x6/6!) + … upto n terms (n given by user).

(Do not make use of inbuilt functions. Use the functions of user defined class MyMath by importing mypackage.)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected OUTCOME of Experiment:**

**CO4:** Explore the interface, exceptions, multithreading, packages.

**Books/ Journals/ Websites referred:**

1. Ralph Bravaco , Shai Simoson , “Java Programming From the Group Up” Tata McGraw-Hill.

2.Grady Booch, Object Oriented Analysis and Design .

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Pre Lab/ Prior Concepts:**

**Java Packages:**

A package in Java is a group of similar types of classes, interfaces, and sub-packages. They can be categorized into two categories, the built-in package ( java, lang, util, awt, javax, swing, net, io, sql et), and user-defined package.

They are used for the following tasks –

* To prevent the naming conflicts which can occur between the classes.
* Make the searching and locating of classes or enumerations or annotations much easier.
* Provide access control to the classes.
* Used for data encapsulation.

**Advantages of Java Package:**

* A Java package is mainly used for the categorization of classes and interfaces so that we can maintain them easily.
* They always provide access protection
* Used to bundle classes and interfaces.
* With the help of packages, we can reuse the existing code
* By using the package, we can easily locate the classes related to it.
* Also, remove the naming collision.

**Built-in Packages in Java**

Built-in is a part of Java API and it offers a variety of packages are –

lang – Automatically imported and it contains language support classes.

io – Contains classes for input and output operations.

util – Contains utility classes for implementing data structures.

applet – This package contains classes that create applets.

awt – Contain classes that implement compounds for GUI.

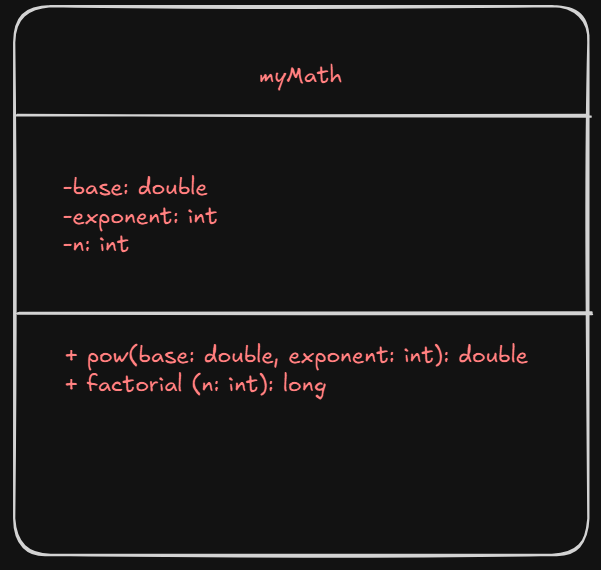
net – This package contains classes that support networking operations.

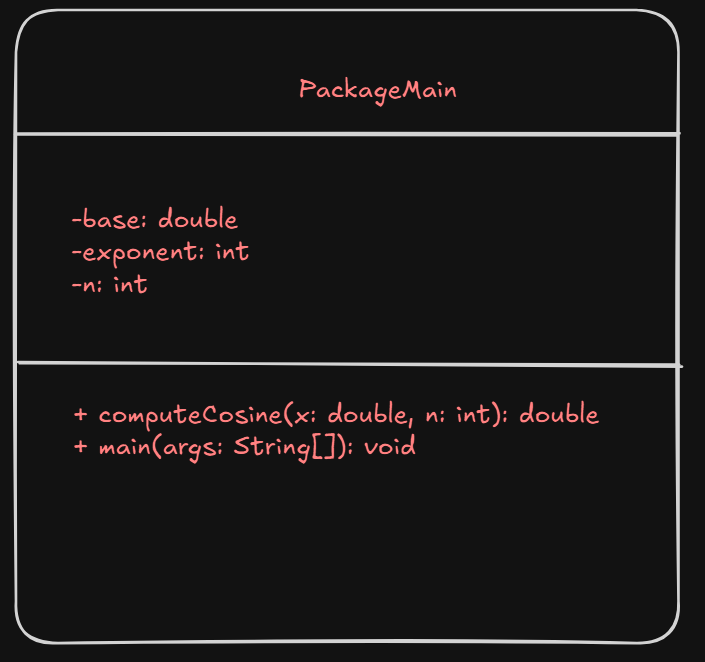
**User-defined Packages in Java**

1. package First;
2. public class MyClass
3. {
4. public void **getNames**(String name)
5. {
6. System.out.**println**(name);
7. }
8. }
9. package First;
10. import First.MyClass;
11. public class MyClass1 {
12. public static void **main**(String args[])
13. {
14. // Initializing the String variable with a value
15. String name = "Welcome";
16. // Creating an instance of class MyClass in the package.
17. MyClass obj = new **MyClass**();
18. obj.**getNames**(name);
19. }
20. }

.

**Class Diagram:**





**Algorithm:**

1. Start
2. Input the value of x(in degrees) and n
3. Convert the value of x in radians
4. Calculate the cosine series using computeCosine
5. Display the result
6. End

**Implementation details:**

myMath.java

package myPackage;

public class myMath {

    public static double  pow(double base, int exponent) {

        double result = 1;

        for (int i = 0; i < exponent; i++) {

            result \*= base;

        }

        return result;

    }

    public static long factorial(int n) {

        if (n == 0) {

            return 1;

        }

        else{

            return n \* factorial(n - 1);

        }

    }

}

PackageMain.java

import myPackage.myMath;

import java.util.Scanner;

public class PackageMain {

    public static double computeCosine(double x, int n) {

        double cos = 1;

        int sign = -1;

        for (int i = 1; i < n; i++) {

            int power = 2 \* i;

            double term = sign \* (myMath.pow(x, power) / myMath.factorial(power));

            cos += term;

            sign \*= -1;

        }

        return cos;

    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the value of x (in degrees): ");

        double xDegrees = sc.nextDouble();

        System.out.print("Enter the number of terms (n): ");

        int n = sc.nextInt();

        sc.close();

        double PI = 3.141592653589793;

        double rad = PI \* xDegrees / 180;

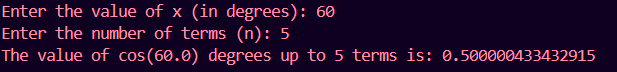
        double cosineValue = computeCosine(rad, n);

        System.out.println("The value of cos(" + xDegrees + ") degrees up to " + n + " terms is: " + cosineValue);

    }

}

**Output:**



**Conclusion:**

From this experiment we learnt about packages and how to make user defined packages

**Date: 10/10/24 Signature of faculty in-charge**

**Post Lab Descriptive Questions**

Q.1   What are Java Packages? What's the significance of packages?

Ans package: A collection of related classes.

Can also "contain" sub-packages.

Sub-packages can have similar names, but are not actually contained inside.

• Uses of Java packages:

group related classes together

as a namespace to avoid name collisions

provide a layer of access / protection

keep pieces of a project down to a manageable size

Q.2 Does Importing a package imports its sub-packages as well in Java?

Ans. No we will have to import the subpackages explicitly. Importing com.MyTest.\* will import classes in the package MyTest only. It will not import any class in any of it's subpackage.

Q.3 Write a program to create a package ‘myPack’ which contains a class Trigonometry. The

class contains following static methods.

i) sine() –accepts degree (0,30,60,90)

ii) cos() - accepts degree (0,30,60,90)

iii)tan()- accepts degree (0,30,60,90)

iv)cot()-- accepts degree (0,30,60,90)

v)cosec()-- accepts degree (0,30,60,90)

vi)sec()-- accepts degree (0,30,60,90)

(Do not make use of inbuilt functions. Use the functions of user defined class Trigonometry by

importing mypack.)

Ans:-

package mypack;

class Trignometry {

    static double sine(int deg){

        if(deg == 0)

            return 0;

        else if(deg == 30)

            return 0.5;

        else if(deg == 60)

            return 0.866;

        else if(deg == 90)

            return 1;

        return -1;

    }

    static double cos(int deg){

        if(deg == 0)

            return 1;

        else if(deg == 30)

            return 0.866;

        else if(deg == 60)

            return 0.5;

        else if(deg == 90)

            return 0;

        return -1;

    }

    static double tan(int deg){

        if(deg == 0)

            return 0;

        else if(deg == 30)

            return 0.577;

        else if(deg == 60)

            return 1.732;

        else if(deg == 90)

            return -1;

        return -1;

    }

    static double cot(int deg){

        if(deg == 0)

            return -1;

        else if(deg == 30)

            return 1.732;

        else if(deg == 60)

            return 0.577;

        else if(deg == 90)

            return 0;

        return -1;

    }

    static double sec(int deg){

        if(deg == 0)

            return 1;

        else if(deg == 30)

            return 1.155;

        else if(deg == 60)

            return 2;

        else if(deg == 90)

            return 0;

        return -1;

    }

    static double cosec(int deg){

        if(deg == 0)

            return 0;

        else if(deg == 30)

            return 2;

        else if(deg == 60)

            return 1.155;

        else if(deg == 90)

            return 1;

        return -1;

    }

}