

Batch: B2 Roll No.: 16010124107

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 the following static methods.

i) power (x, y) – to compute x^y

ii) fact (x) – to compute $x!$

Write a program to find the following series.

$\cos(x) = 1 - (x^2/2!) + (x^4/4!) - (x^6/6!) + \dots$ 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.
3.    public class MyClass
4.    {
5.    public void getNames(String name)
6.    {
7.    System.out.println(name);
8.    }
9.
10. }
```

```
1.    package First;
2.    import First.MyClass;
3.    public class MyClass1 {
4.    public static void main(String args[])
5.    {
6.    // Initializing the String variable with a value
7.    String name = "Welcome";
8.    // Creating an instance of class MyClass in the package.
9.    MyClass obj = new MyClass();
10.   obj.getNames(name);
11.   }
12.   }
```

Class Diagram:

myPackage ->myMath
public static double power(double a, int b) public static double fact(int x)
double result;

myClass
public static void main()
int x, double n, double ans

Algorithm:

1. Start
2. Create a folder and inside it create the class of the package
3. Create two functions power and fact and define them with their functionality
4. Create a main class outside the package folder
5. Create a main function inside this class
6. Import scanner and take all inputs
7. Run a loop for N terms and for each time, calculate the term to be added
8. Output the value of cos x
9. End

Implementation details:

```
package myPackage;

public class MyMath{

    public static double power(double a, int b) {

        double result = 1;

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

            result *= a;

        }

        return result;

    }

    public static double fact(int x) {

        double result = 1;

        for(int i = 2; i <= x; i++) {
```

```
        result *= i;

    }

    return result;

}

}
```

```
import java.util.*;

import myPackage.MyMath;

class myClass{

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter value of x.");

        double x = sc.nextDouble();

        System.out.println("Enter number of terms.");

        int n = sc.nextInt();

        int counter=2;

        double ans=1;

        n--;

        while(n > 0){

            int sign = ( (counter/2) % 2 == 0 ) ? -1 : 1;

            // System.out.println("Power term is " +
MyMath.power(x,counter));
```

K. J. Somaiya School of Engineering, Mumbai-77
(A Constituent College of Somaiya Vidyavihar University)
Department of Computer Engineering

```
        // System.out.println("Factorial is " +  
MyMath.fact(counter));  
  
        double term = (double)MyMath.power(x,counter) /  
(double)MyMath.fact(counter);  
  
        ans += (term*sign);  
  
        counter+=2;  
  
        n--;  
  
    }  
  
    System.out.println("cos(" + x + ") = " + ans);  
  
}  
  
}
```

Output:

```
● PS C:\Users\syeda\OneDrive\Desktop\personal> javac myClass.java  
● PS C:\Users\syeda\OneDrive\Desktop\personal> java myClass  
Enter value of x.  
5  
Enter number of terms.  
10  
cos(5.0) = 1.716374984910827  
❖ PS C:\Users\syeda\OneDrive\Desktop\personal>   
Loading 46 secs Debugging User program finished Java: Ready  
cos(5.0) = 1.716374984910827  
● PS C:\Users\syeda\OneDrive\Desktop\personal> java myClass  
Enter value of x.  
0  
Enter number of terms.  
10  
cos(0.0) = 1.0  
❖ PS C:\Users\syeda\OneDrive\Desktop\personal>
```

Conclusion:

The program successfully demonstrates implementation of user defined packages in Java by separating reusable mathematical functions from the main class. Using the Taylor series expansion, cosine of a number in radians is calculated accurately. The code structure improves modularity, readability and reusability for future programs.

Date: 27/10/2025

Signature of faculty in-charge

Post Lab Descriptive Questions

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

Java packages are a way to group related classes, interfaces, and sub-packages together. They help organize large programs, avoid name conflicts, and improve code maintenance. Packages support modular programming by allowing reusable code and provide access control and better project structure.

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

No, importing a package does not import its sub-packages in Java. Each sub-package must be imported separately using its full package name. For example, importing `java.util.*` does not include `java.util.regex.*`; it must be imported explicitly.

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.)

```
package myPackage;  
  
public class Trigonometry {  
    public static double Sin(int x){  
        switch(x){
```



```
        case 0: return 0;

        case 30: return 0.5;

        case 45: return 0.7071;

        case 60: return 0.8660;

    }
    return -1;
}

public static double Cos(int x){
    switch(x){
        case 0: return 1;

        case 30: return 0.8660;

        case 45: return 0.7071;

        case 60: return 0.5;

    }
    return -1;
}

public static double Tan(int x){
    switch(x){
        case 0: return 0;

        case 30: return 0.5774;

        case 45: return 1;

        case 60: return 1.7321;

    }
    return -1;
}

public static double Cot(int x){
```

```
        switch(x){
            case 0: return Double.POSITIVE_INFINITY;

            case 30: return 1.7321;

            case 45: return 1;

            case 60: return 0.5774;

        }
        return -1;
    }
    public static double Csc(int x){
        switch(x){
            case 0: return Double.POSITIVE_INFINITY;

            case 30: return 2;

            case 45: return 1.4142;

            case 60: return 1.1547;

        }
        return -1;
    }
    public static double Sec(int x){
        switch(x){
            case 0: return 1;

            case 30: return 1.1547;

            case 45: return 1.4142;

            case 60: return 2;

        }
        return -1;
    }
}
```

```
}
```

```
import java.util.Scanner;
import myPackage.Trigonometry;
public class myPack {
    public static void main(String[] args) {
        System.out.println("Enter angle in degrees among 0, 30, 45,
60:");
        Scanner sc = new Scanner(System.in);
        int angle = sc.nextInt();
        System.out.println("Sin(" + angle + ") = " +
Trigonometry.Sin(angle));
        System.out.println("Cos(" + angle + ") = " +
Trigonometry.Cos(angle));
        System.out.println("Tan(" + angle + ") = " +
Trigonometry.Tan(angle));
        System.out.println("Cot(" + angle + ") = " +
Trigonometry.Cot(angle));
        System.out.println("Csc(" + angle + ") = " +
Trigonometry.Csc(angle));
        System.out.println("Sec(" + angle + ") = " +
Trigonometry.Sec(angle));
    }
}
```

```
PS C:\Users\syeda\OneDrive\Desktop\personal> java myPack
Enter angle in degrees among 0, 30, 45, 60:
0
Sin(0) = 0.0
Cos(0) = 1.0
Tan(0) = 0.0
Cot(0) = Infinity
Csc(0) = Infinity
Sec(0) = 1.0
PS C:\Users\syeda\OneDrive\Desktop\personal> java myPack
Enter angle in degrees among 0, 30, 45, 60:
30
Sin(30) = 0.5
Cos(30) = 0.866
Tan(30) = 0.5774
Cot(30) = 1.7321
Csc(30) = 2.0
Sec(30) = 1.1547
PS C:\Users\syeda\OneDrive\Desktop\personal> █
```