Batch: A1 **Roll No.: 16010123012**

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.
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. }
```

Algorithm:

Method power(x, y):
Initialize result to 1.
For each i from 0 to y - 1, multiply result by x.
Return result, which is now x^y

Method fact(x):
Initialize factorial to 1.
For each integer from 2 up to xxx, multiply factorial by the integer.
Return factorial, which is now x!

Define calculateCosine(x, n):
Initialize result to 0 to store the cosine series result.
For each i from 0 to n - 1:
Calculate 2 * i (for the even powers in the cosine series).
Calculate the term using mymath.power(x, 2 * i) / mymath.fact(2 * i).
If i is even, add the term to result; if i is odd, subtract the term from result.
Return result as the approximate value of cos(x).

Main Method (Execution Flow):
Create a Scanner object for user input.
Prompt the user to enter x (in radians).
Prompt the user to enter the number of terms n.
Call calculateCosine(x, n) and store the result in cosineValue.
Print the result as the approximate value of cos(x).
Close the Scanner

Implementation details:

```java
package myPackage;

public class mymath {

    public static double power(double x, int y) {
        double result = 1;
        for (int i = 0; i < y; i++) {
            result *= x;
        }
        return result;
    }

    public static long fact(int x) {
        long factorial = 1;
        for (int i = 2; i <= x; i++) {
            factorial *= i;
        }
        return factorial;
    }
}
```

```java
import java.util.*;
import myPackage.mymath;

public class CosineCalculator {

    public static double calculateCosine(double x, int n) {
        double result = 0;

        for (int i = 0; i < n; i++) {
            double term = mymath.power(x, 2 * i) / mymath.fact(2 * i);
            if (i % 2 == 0) {
                result += term;
            } else {
                result -= term;
            }
        }

        return result;
```

```
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the value of x (in radians): ");
        double x = sc.nextDouble();

        System.out.print("Enter the number of terms for approximation:
");
        int n = sc.nextInt();

        double cosineValue = calculateCosine(x, n);

        System.out.printf("Approximate value of cos(%.2f) using %d terms:
%.8f\n", x, n, cosineValue);

        sc.close();
    }
}
```

Output:

```
PS D:\KJSCE\SY\OOPS\Program\package> javac -d build1 -cp build1 CosineCalculator.java
PS D:\KJSCE\SY\OOPS\Program\package> java -cp build1 CosineCalculator
Enter the value of x (in radians): 35
Enter the number of terms for approximation: 4
Approximate value of cos(35.00) using 4 terms: -2491232.15972222
```

Conclusion:

We learnt about user defined packages in java and how we can implement it for various applications thus successfully implemented the experiment.

Date: 05 / 11 / 2024 **Signature of faculty in-charge**

Post Lab Descriptive Questions

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

A package in Java is a namespace that groups related classes and interfaces. It helps in organizing the classes in a structured manner.

Significance:

Organization: Packages help in organizing large codebases by grouping related classes and interfaces, making it easier to manage and navigate the code.
Namespace Management: Packages prevent naming conflicts. Classes in different packages can have the same name without causing a collision. For example, java.util.List and java.awt.List can coexist.
Access Control: Packages provide a way to control access to classes and members. For example, classes that are declared as package-private (without an access modifier) are only accessible within the same package.
Reusability: Well-defined packages can be reused across different projects. Java's standard libraries are organized into packages, allowing developers to easily include and use existing functionalities.
Modularity: Packages contribute to modular programming, where classes can be developed, tested, and maintained independently. This modularity is beneficial for teamwork and code maintenance.
Easier Maintenance: By grouping related classes, packages make it easier to locate and update code, improving maintainability.

Q.2 Does Importing a package imports its sub-packages as well in Java?
No, importing a package in Java does not automatically import its sub-packages. When you import a package, only the classes and interfaces in that specific package are available to use. If you want to use classes from sub-packages, you need to import each sub-package 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 sine(int degree) {
        return switch (degree) {
            case 0 ->
                0;
            case 30 ->
                0.5;
            case 60 ->
                Math.sqrt(3) / 2;
            case 90 ->
                1;
            default ->
                throw new IllegalArgumentException("Invalid degree: " +
degree);
        };
    }

    public static double cos(int degree) {
        return switch (degree) {
            case 0 ->
                1;
            case 30 ->
                Math.sqrt(3) / 2;
            case 60 ->
                0.5;
            case 90 ->
                0;
            default ->
                throw new IllegalArgumentException("Invalid degree: " +
degree);
        };
    }

    public static double tan(int degree) {
        return sine(degree) / cos(degree);
    }

    public static double cot(int degree) {
        return 1 / tan(degree);
    }

    public static double cosec(int degree) {
        return 1 / sine(degree);
    }
```

```
    public static double sec(int degree) {
        return 1 / cos(degree);
    }
}
```

```
import myPackage.Trigonometry;
import java.util.*;

public class PostLab {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the degree (0, 30, 60, 90): ");
        int degree = sc.nextInt();

        if (degree != 0 && degree != 30 && degree != 60 && degree != 90)
{
            System.out.println("Invalid degree. Please enter 0, 30, 60,
or 90.");
        } else {
            System.out.println("Sine of " + degree + " degrees: " +
Trigonometry.sine(degree));
            System.out.println("Cosine of " + degree + " degrees: " +
Trigonometry.cos(degree));
            System.out.println("Tangent of " + degree + " degrees: " +
Trigonometry.tan(degree));
            System.out.println("Cotangent of " + degree + " degrees: " +
Trigonometry.cot(degree));
            System.out.println("Cosecant of " + degree + " degrees: " +
Trigonometry.cosec(degree));
            System.out.println("Secant of " + degree + " degrees: " +
Trigonometry.sec(degree));
        }
        sc.close();
    }
}
```

```
PS D:\KJSCE\SY\OOPS\Program\package> javac -d build0 -cp build0 PostLab.java
PS D:\KJSCE\SY\OOPS\Program\package> java -cp build0 PostLab
Enter the degree (0, 30, 60, 90): 60
Sine of 60 degrees: 0.8660254037844386
Cosine of 60 degrees: 0.5
Tangent of 60 degrees: 1.7320508075688772
Cotangent of 60 degrees: 0.5773502691896258
Cosecant of 60 degrees: 1.1547005383792517
Secant of 60 degrees: 2.0
```