

Experiment No. 5

Title: Create and use user defined package

Batch:**Roll No.:****Experiment No.:**

Aim: Create a package 'myPackage' which contains a class myMath. The class contains following static methods.

i) power (x, y) – to compute x^y

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.)

Resources needed: java

Theory:

Packages:

A package is a collection of related classes. It helps to organize your classes into a folder structure and makes it easy to locate and use them. More importantly, it helps improve re-usability.

Syntax:- package <package_name>;

Steps to create a package:

Step 1) Create a package named "p1" and declare a class named "c1" in it as shown below

```
package p1;
class c1{
public void m1(){
System.out.println("Method m1 of Class c1");
}
public static void main(String args[]){
c1 obj = new c1();
obj.m1();
}
}
```

Step 2) Save the file as c1.java.

Step 3) Compile the file as,

javac -d . c1.java

where

-d – it is used to specify the location where the .class file will be saved after compilation

.(dot) – It refers to the current folder.

-d . – It is used to save the .class file in the current folder.

Step 3) Run the file as

java p1.c1

Steps to create a sub-package

Step1) Create a package named “p1.p2” and declare a class named “c2” in it as shown below

```
package p1.p2;
class c2{
public void m2(){
System.out.println("Method m2 of Class c2");
}
public static void main(String args[]){
c2 obj = new c2();
obj.m2();
}
}
```

Step 2) Save the file as c2.java.

Step 3) Compile the file as,

```
javac -d . c2.java
```

Step 4) Run the file as,

```
java p1.p2.c2
```

Results: (Program with output)

Code :

```
class myMath {
// Method to calculate power (x^y)
public static double power(double x, int y) {
double result = 1;
for (int i = 1; i <= y; i++) {
result *= x;
}
return result;
}

// Method to calculate factorial
public static double factorial(int n) {
double fact = 1;
for (int i = 1; i <= n; i++) {
fact *= i;
}
return fact;
}
}

public class Main {
public static void main(String[] args) {
java.util.Scanner scanner = new java.util.Scanner(System.in);
```

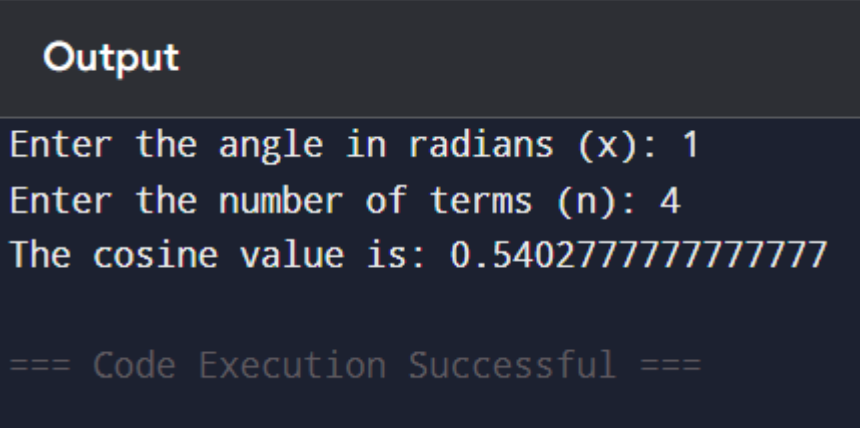
```
System.out.print("Enter the angle in radians (x): ");
double x = scanner.nextDouble();

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

double cosValue = 0;
for (int i = 0; i < n; i++) {
    int power = 2 * i;
    double term = myMath.power(x, power) / myMath.factorial(power);
    cosValue += (i % 2 == 0) ? term : -term;
}

System.out.println("The cosine value is: " + cosValue);
}
```

Output :



Output

```
Enter the angle in radians (x): 1
Enter the number of terms (n): 4
The cosine value is: 0.5402777777777777

=== Code Execution Successful ===
```

Questions:

1. List any five inbuilt packages and its usage?

java.util: Provides utility classes such as Scanner, ArrayList, HashMap, and date/time utilities.

java.lang: Contains fundamental classes like String, Math, and System. Automatically imported.

java.io: Supports input/output operations, including file handling with classes like File, BufferedReader, and PrintWriter.

java.net: Provides classes for networking, such as Socket, ServerSocket, and URL.

javax.swing: Offers GUI components like JFrame, JButton, and JPanel for creating desktop applications.

2. Demonstrate the visibility of a class member having private and default access specifier with respect to packages with the help of an appropriate java program.

```
// Main Java file simulating the demo of private and default access
```

```
public class Main {
```

```
    // DemoClass simulating private and default access
```

```
    static class DemoClass {
```

```
        private int privateValue = 10; // Private access
```

```
        int defaultValue = 20; // Default access
```

```
        public void showValues() {
```

```
            System.out.println("Private Value: " + privateValue);
```

```
            System.out.println("Default Value: " + defaultValue);
```

```
        }
```

```
    }
```

```
    public static void main(String[] args) {
```

```
        // Create an object of DemoClass
```

```
        DemoClass obj = new DemoClass();
```

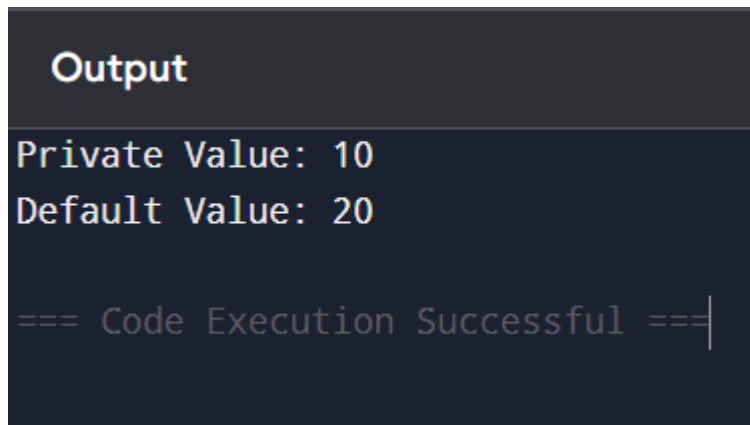
```
        // Call the public method to display the values
```

```
        obj.showValues();
```

```
        // Uncommenting below lines will cause errors
```

```
        // System.out.println(obj.privateValue); // Error: privateValue is not accessible outside the class
```

```
// System.out.println(obj.defaultValue); // Error: defaultValue is not accessible outside the package  
  
}  
  
}
```



```
Output  
Private Value: 10  
Default Value: 20  
  
=== Code Execution Successful ===|
```

Outcomes:

CO3: Demonstrate the concept of packages, multithreading and exception handling in java

Conclusion: (Conclusion to be based on the outcomes achieved)

From this article I learned how to create custom classes with static methods and simulate packages in Java. I understood the importance of access specifiers and how they control the visibility of class members. Additionally, I practiced implementing mathematical series without relying on built-in functions.

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

Signature of faculty in-charge with date

References:

Books/ Journals/ Websites:

1. Herbert Schildt, “Java: The Complete Reference” Tata McGrawHill Publishing Company Limited Tenth Edition, 2017
2. Sachin Malhotra, Saurab Choudhary, “Programming in Java” Oxford University Press Second Edition, 2018 3.
3. D.T. Editorial Services, “Java 8 Programming Black Book” Dream tech Press Edition 2015