Batch-A2 Roll No : 16010123032

Expirement NO: 9

**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 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.

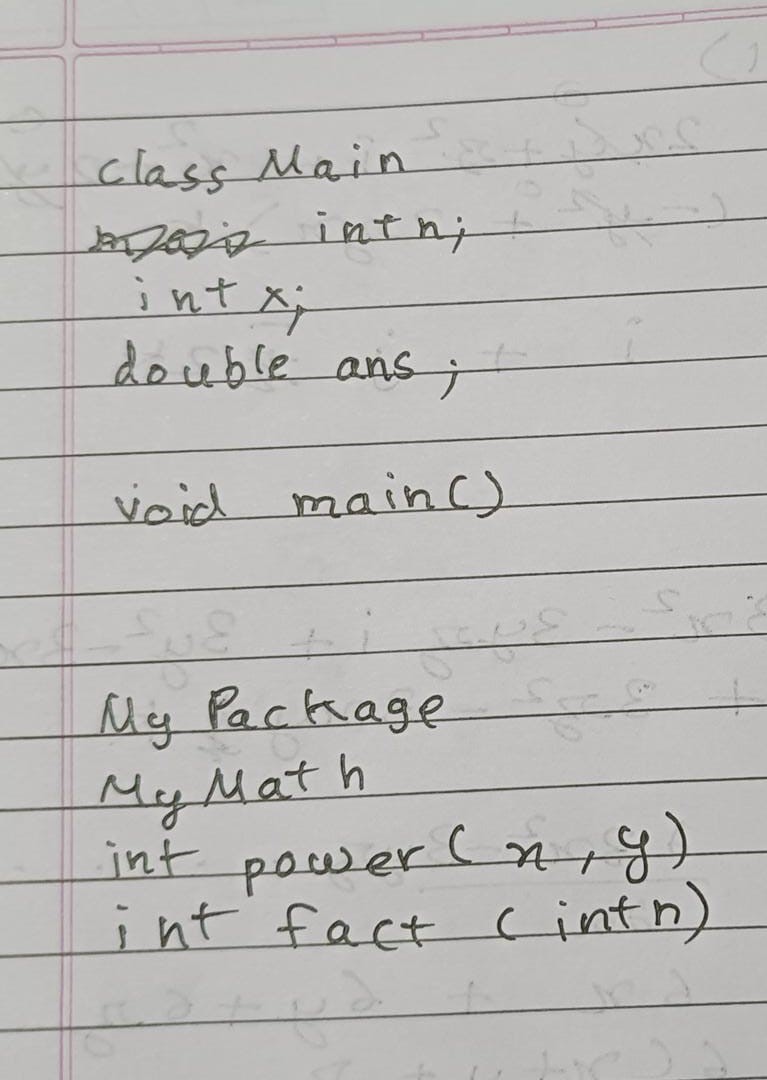
util – Contains utility classes for implementing data structures. applet – This package contains classes that create applets. awt – Contain classes that implement compounds for GUI. net – This package contains classes that support networking operations.

**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. | } |

.

**Class Diagram:**



**Algorithm:**

**Calculate factorial**

* The fact function calculates the factorial of a non-negative integer n. It uses a loop to iterate from 1 to n, multiplying the factorial by the current iterator value at each step.

**Calculate power**

* The power function calculates the power of a number x raised to the power of y.

It uses a loop to iterate y times, multiplying the result by x at each step.

**Calculate cosine series**

* The cosine\_series function calculates an approximation of the cosine of x using the first n terms of the cosine series.

○ It initializes answer to 1, term to 2, and sign to 1.

○ It iterates n times:

■ Calculates the current term of the cosine series using power(x, term) / factorial(term).

■ Alternates the sign of the term using if sign = -1 if i is odd and vice versa.

■ Adds the current term to answer

○ It returns the answer

**Main function**

* The main function prompts the user to enter the number of terms (n) and the value of x.
* It creates a Scanner object to read user input.
* It calls the cosine\_series function to calculate the cosine series approximation for the given x and n.
* It prints the cosine series approximation to the console.

**Implementation details:**

**Package :** package MyPackage; public class MyMath

{ public static int power(int x, int y)

{

int i; int pro=1; for(i=1;i<=y;i++)

{ pro=pro\*x; }

return pro; }

public static int fact(int n)

{

int i; int f=1; for(i=1;i<=n;i++)

{

f=f\*i; }

return f;

}

}

**Main:**

import java.util.\*; import MyPackage.\*; public class Main

{ public static void main(String[]args)

{

Scanner sc= new Scanner(System.in);

MyMath M= new MyMath();

System.out.println("Enter maximum terms to display "); int n=sc.nextInt();

System.out.println("Enter variable x value: "); int x=sc.nextInt();

System.out.print(" cos("+x+") = 1 ");

double ans=1; int j=2;

for(int i=1;i<=n;i++)

{

if(i%2!=0)

{

System.out.print(" - "); ans=ans-(M.power(x,j)/M.fact(j));

}

else

{

System.out.print(" + "); ans=ans+(M.power(x,j)/M.fact(j));

}

System.out.print(+x+"^"+j+"/"+j+"!"); j+=2;

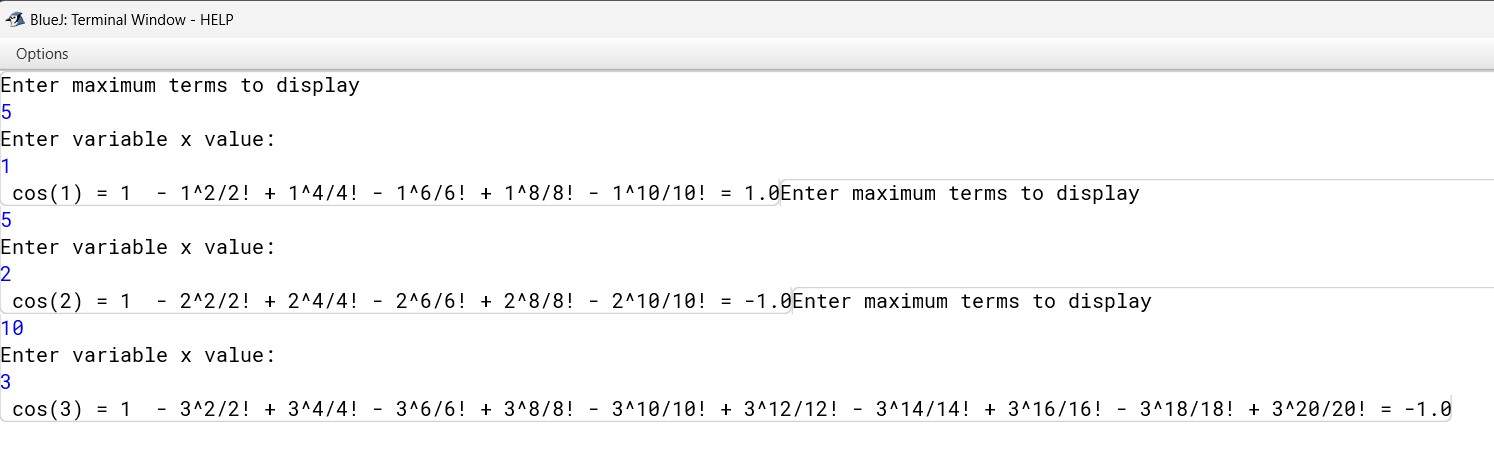
}

System.out.print(" = "+ans);

}

}

**Output:**



**Conclusion:**

**We successfully implemented packages in java.**

**Date: \_\_\_\_\_\_\_\_\_\_ Signature of faculty in-charge**

**Post Lab Descriptive Questions**

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

**Answer:**

Packages are a fundamental concept in Java programming that serve as organizational units for related classes and interfaces. They provide a hierarchical structure similar to folders on a computer, preventing naming conflicts and promoting code reusability.

Packages are defined using the package keyword at the beginning of a Java source file. They can be nested within other packages, creating a hierarchical structure that reflects the organization of your project. The java.lang package is automatically imported into every Java program, containing essential classes like String, Math, and Object.

They are crucial for organizing and managing code effectively. They prevent naming conflicts between classes, promote code reusability, and provide a hierarchical structure for better organization. Packages also control the visibility of classes and members, ensuring that code is accessible only to relevant parts of the application. This enhances code modularity, maintainability, and overall project structure.

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

**Answer:**

In Java, importing a package does not automatically import its sub-packages. To access classes from a sub-package, we need to explicitly import it using the full package path. This allows for better control and organization of your code by making clear which classes and packages are being used.

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.)

Answer:

**Code:**

**Package:** package MyPack;

public class Trignometry

{ public static void sine(int angle)

{ if(angle==0)

{

System.out.println("0");

}

else if(angle==30)

{

System.out.println("0.5");

} else if(angle==60)

{

System.out.println("0.866");

} else if(angle==90)

{

System.out.println("1"); }

else

{

System.out.println("Invalid Angle ");

} }

public static void cosine(int angle)

{ if(angle==0)

{

System.out.println("1"); } else if(angle==30)

{

System.out.println("0.866");

} else if(angle==60)

{

System.out.println("0.5");

} else if(angle==90)

{

System.out.println("0"); }

else

{

System.out.println("Invalid Angle ");

} }

public static void tan(int angle)

{ if(angle==0)

{

System.out.println("0"); } else if(angle==30)

{

System.out.println("0.577");

} else if(angle==60)

{

System.out.println("1.732");

} else if(angle==90)

{

System.out.println("Undefined ");

}

else

{

System.out.println("Invalid Angle ");

} }

public static void cosec(int angle)

{ if(angle==0)

{

System.out.println("Undefined ");

} else if(angle==30)

{

System.out.println("2");

} else if(angle==60)

{

System.out.println("1.155");

} else if(angle==90)

{

System.out.println("1"); }

else

{

System.out.println("Invalid Angle ");

} }

public static void sec(int angle)

{ if(angle==0)

{

System.out.println("1"); } else if(angle==30)

{

System.out.println("1.155");

} else if(angle==60)

{

System.out.println("2"); } else if(angle==90)

{

System.out.println("Undefined ");

}

else

{

System.out.println("Invalid Angle ");

}

}

public static void cot(int angle)

{ if(angle==0)

{

System.out.println("Undefined ");

} else if(angle==30)

{

System.out.println("1.733");

} else if(angle==60)

{

System.out.println("0.577");

} else if(angle==90)

{

System.out.println("0"); }

else

{

System.out.println("Invalid Angle ");

}

}

}

**main:**

import MyPack.\*; import java.util.Scanner;

public class myclass

{ public static void main(String[]args)

{

Scanner sc=new Scanner(System.in);

Trignometry t=new Trignometry();

System.out.println("Enter Angle (0,30,60,90)"); int an=sc.nextInt(); System.out.print("sin("+an+") = "); t.sine(an);

System.out.print("cos("+an+") = "); t.cosine(an);

System.out.print("tan("+an+") = "); t.tan(an);

System.out.print("cosec("+an+") = "); t.cosec(an);

System.out.print("sec("+an+") = "); t.sec(an);

System.out.print("cot("+an+") = "); t.cot(an);

}

}

**Output:**

