**Batch: B-1 Roll No.: 16010122104**

**Experiment / assignment / tutorial No. 02**

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

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

|  |
| --- |
| **TITLE : Control Statement** |

**AIM:** Create 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.

* ex = 1 +(x/1!)+ (x2/2!) + (x3/3!) + (x4/4!) + … upto n terms (n given by user).
* (1+x)n = 1 +(nx/1!)+ ((n(n-1)x2)/2!) ........ upto n terms (n given by user).

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

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

**Expected OUTCOME of Experiment:**

**CO2:** Explore arrays, vectors, classes and objects in C++ and Java.

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

**Books/ Journals/ Websites referred:**

1. E. Balagurusamy , “Programming with Java” McGraw-Hill.
2. Sachin Malhotra, Saurabh Choudhary, “Programming in Java”, Oxford Publications.

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

**Pre Lab/ Prior Concepts**

Java basic constructs (like if else statement, control structures, and data types

Programming languages provide various control structures that allow for more complicated execution paths.

A loop statement allows us to execute a statement or group of statements multiple times and following is the general form of a loop statement in most of the programming languages −

|  |  |
| --- | --- |
| **Sr.No.** | **Loop & Description** |
| 1 | [**while loop**](https://www.tutorialspoint.com/java/java_while_loop.htm)  Repeats a statement or group of statements while a given condition is true. It tests the condition before executing the loop body. |
| 2 | [**for loop**](https://www.tutorialspoint.com/java/java_for_loop.htm)  Execute a sequence of statements multiple times and abbreviates the code that manages the loop variable. |
| 3 | [**do...while loop**](https://www.tutorialspoint.com/java/java_do_while_loop.htm)  Like a while statement, except that it tests the condition at the end of the loop body. |

**Loop Control Statements**

Loop control statements change execution from its normal sequence. When execution leaves a scope, all automatic objects that were created in that scope are destroyed.

Java supports the following control statements. Click the following links to check their details.

|  |  |
| --- | --- |
| **Sr.No.** | **Control Statement & Description** |
| 1 | [**break statement**](https://www.tutorialspoint.com/java/java_break_statement.htm)  Terminates the loop or switch statement and transfers execution to the statement immediately following the loop or switch. |
| 2 | [**continue statement**](https://www.tutorialspoint.com/java/java_continue_statement.htm)  Causes the loop to skip the remainder of its body and immediately retest its condition prior to reiterating. |

**Class Diagram:**

**Algorithm:**

1. Start

2. Define a static function power(x, y)

a. If y is 0, return 1

b. If y is even, recursively calculate power(x\*x, y/2)

c. If y is odd, recursively calculate x \* power(x\*x, (y-1)/2)

3. End of power function

4. Define a static function fact(x)

a. If x is 0 or 1, return 1

b. Otherwise, recursively calculate x \* fact(x-1)

5. End of fact function

6. Define the main function

a. Create a Scanner object named scanner to read input

b. Print "Enter the value of x: "

c. Read x from the user using scanner

d. Print "Enter the value of n: "

e. Read n from the user using scanner

f. Initialize resultExpX to 1

g. For i from 1 to n, do steps h and i

h. Calculate myMath.power(x, i) / myMath.fact(i) and add it to resultExpX

i. End loop

j. Initialize result1PlusXPowerN to 1

k. Initialize term to 1

l. For i from 1 to n, do steps m and n

m. Calculate term as (x \* (n - i + 1)) / i

n. Add term to result1PlusXPowerN

o. End loop

p. Print "e^x = resultExpX"

q. Print "(1 + x)^n = result1PlusXPowerN"

7. End of main function

8. End

**Implementation details:**

import java.util.Scanner;

class myMath

{

    // Method to compute x^y

    static double power(double x, int y)

    {

        if (y == 0)

            return 1;

        else if (y % 2 == 0)

            return power(x \* x, y / 2);

        else

            return x \* power(x \* x, (y - 1) / 2);

    }

    // Method to compute x!

    static int fact(int x)

    {

        if (x == 0 || x == 1)

            return 1;

        else

            return x \* fact(x - 1);

    }

    public static void main(String[] args)

    {

        Scanner scanner = new Scanner(System.in);

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

        double x = scanner.nextDouble();

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

        int n = scanner.nextInt();

        // Evaluate e^x using series expansion: e^x = 1 + x + (x^2/2!) + (x^3/3!) + ... + (x^n/n!)

        double resultExpX = 1;

        for (int i = 1; i <= n; i++)

        {

            resultExpX += myMath.power(x, i) / myMath.fact(i);

        }

        // Evaluate (1+x)^n using series expansion: (1+x)^n = 1 + n\*x + (n\*(n-1)\*x^2/2!) + ... + (x^n)

        double result1PlusXPowerN = 1;

        double term = 1;

        for (int i = 1; i <= n; i++)

        {

            term \*= (x \* (n - i + 1)) / i;

            result1PlusXPowerN += term;

        }

        System.out.println("e^" + x + " = " + resultExpX);

        System.out.println("(1 + " + x + ")^" + n + " = " + result1PlusXPowerN);

    }

}

**Output:**

A screenshot of a computer

Description automatically generated

**Conclusion:**

Hence, we learn the use of control loop statements, like for, while and do…..while.

**Date:07/08/2023 Signature of faculty in-charge**

**Post Lab Descriptive Questions**

Q.1 Write a program to find the largest of three numbers using the if-else construct.

**Ans:**

import java.util.Scanner;

public class LargestOfThree {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the first number: ");

double num1 = scanner.nextDouble();

System.out.print("Enter the second number: ");

double num2 = scanner.nextDouble();

System.out.print("Enter the third number: ");

double num3 = scanner.nextDouble();

double largest;

if (num1 >= num2 && num1 >= num3) {

largest = num1;

} else if (num2 >= num1 && num2 >= num3) {

largest = num2;

} else {

largest = num3;

}

System.out.println("The largest number is: " + largest);

}

}

A screenshot of a computer

Description automatically generated

Q.2 Write a program to determine the sum of the following series for a given value of n:

1+½+⅓+....+1/n

**Ans:**

import java.util.Scanner;

public class SeriesSum {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int n = scanner.nextInt();

double sum = 0.0;

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

sum += 1.0 / i;

}

System.out.println("Sum of the series: " + sum);

}

}

A screenshot of a computer

Description automatically generated