| | | | |
|---|---|---|---|
| **Course Name:** | **Object Oriented Programming Methodology** | **Semester:** | **III** |
| **Date of Performance:** | **08/08/2023** | **Batch No:** | **B2** |
| **Faculty Name:** | **Prof: KIRAN THALE** | **Roll No:** | 16010122151 |
| **Faculty Sign & Date:** | | **Grade/Marks:** | ___/15 |

# Experiment No: 2 Title:
# Control Statements

## Aim and Objective of the Experiment:

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.

- $e^x = 1 + (x/1!) + (x2/2!) + (x3/3!) + (x4/4!) + \ldots$ 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).

**Note: Do not make use of inbuilt functions. Use the functions of user defined class MyMath.**

## COs to be achieved:

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

## Tools used:

JDK, VScode / Eclipse

## Theory:

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 |
|---|---|

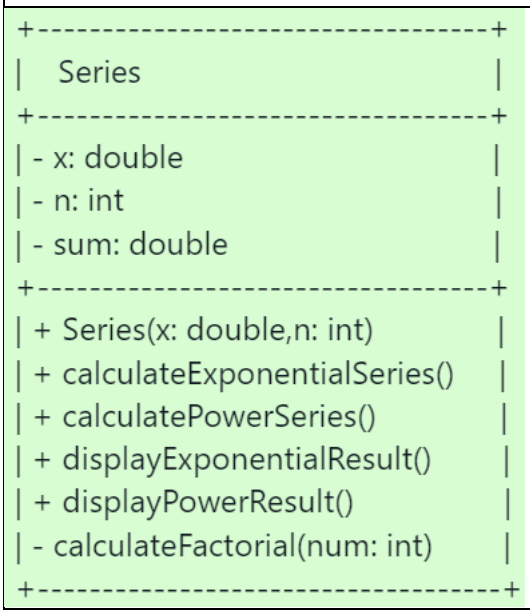
| | |
|---|---|
| 1 | **while loop**<br>Repeats a statement or group of statements while a given condition is true. It tests the condition before executing the loop body. |
| 2 | **for loop**<br>Execute a sequence of statements multiple times and abbreviates the code that manages the loop variable. |
| 3 | **do...while loop**<br>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**<br>Terminates the loop or switch statement and transfers execution to the statement immediately following the loop or switch. |
| 2 | **continue statement**<br>Causes the loop to skip the remainder of its body and immediately retest its condition prior to reiterating. |

**Class Diagram:**



## Algorithm:

1 Import Necessary Classes: The code begins by importing the java.util.* package, which includes classes for handling user input, and the Scanner class, which is used to read input from the console.

2 Define the expo Class: The main class of the program is defined as expo.

3 Declare Variables: Two float variables, x and n, are declared to store input values. Additionally, a Scanner object named s is created to read user input from the console.

4 Read Input: The program prompts the user to enter two floating-point numbers (x and n) and uses the Scanner object to read and store these values.

5 Initialize Variables: Two float variables, result1 and result2, are initialized to 1. These variables will be used to store the results of exponential calculations.

6. Calculate `e^x` (result1) using a loop:
   - Start a loop with a variable `a` from 1 to `n - 1`.
   - Inside the loop, calculate `myMath.pow(x, a)` (x raised to the power of `a`) using the custom
   - `pow` function from the `myMath` class.
   - Calculate `myMath.fact(a)` (factorial of `a`) using the custom `fact` function from the
   - `myMath` class.
     Increment `result1` by `(myMath.pow(x, a)) / (myMath.fact(a))`.

7. Calculate `(1+x)^n` (result2) using another loop:

- Declare a float variable `k` and initialize it with the value of `n`.
- Start a loop with a variable `a` from 1 to `n`.
- Inside the loop, calculate `myMath.pow(x, a)` (x raised to the power of `a`) using the custom pow` function.
- Calculate `myMath.fact(a)` (factorial of `a`) using the custom `fact` function.
- Increment `result2` by `(k) * (myMath.pow(x, a)) / (myMath.fact(a))`.
- Update the value of `k` by multiplying it with `(n - a)`.

8. Print the results: After the loops, the program prints the calculated values of `e^x` and `(1+x)^n`.
9. Define the `myMath` class: A separate class `myMath` is defined to encapsulate custom mathematical functions used in the main `expo` class.

10. Implement custom `pow` function: The `myMath` class defines a static method `pow` that calculates the power of a given number (`x`) raised to another number (`y`) using a loop.
11. Implement custom `fact` function: The `myMath` class defines another static method `fact`
12. that calculates the factorial of a given number (`x`) using a loop.

**Code:**

```
import java.util.Scanner;

public class ExpoCalculator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter the values of x and n:");
        float x = scanner.nextFloat();
        float n = scanner.nextFloat();

        float result1 = calculateExponential(x, n);
        float result2 = calculatePowerSeries(x, n);

        System.out.println("e^x is " + result1);
        System.out.println("(1+x)^n is " + result2);
    }

    static float calculateExponential(float x, float n) {
        float result = 1;
        for (float a = 1; a <= n; a++) {
            result += myMath.pow(x, a) / myMath.fact(a);
        }
        return result;
    }

    static float calculatePowerSeries(float x, float n) {
        float result = 1;
        float k = n;
        for (float a = 1; a <= n; a++) {
            result += k * myMath.pow(x, a) / myMath.fact(a);
            k *= (n - a);
        }
        return result;
    }
}

class myMath {
    static float pow(float x, float y) {
        float p = 1;
        for (float i = 1; i <= y; i++) {
            p *= x;
        }
        return p;
```

```
    }

    static float fact(float x) {
        float f = 1;
        for (float i = 1; i <= x; i++) {
            f *=
;
        }
        return f;
    }
}
```

**Output:**

```
Enter the values of x and n:
2 3
e^x is 6.3333335
(1+x)^n is 27.0
```

**Post Lab Subjective/Objective type Questions:-**

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

```java
import java.util.Scanner;

public class LargestNumberFinder {

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

        System.out.println("Enter three numbers: ");
        float a, b, c;

        a = scanner.nextFloat();
        b = scanner.nextFloat();
        c = scanner.nextFloat();

        if (a >= b && a >= c) {
            System.out.println(a + " is the largest number.");
        } else if (b >= a && b >= c) {
            System.out.println(b + " is the largest number.");
        } else {
            System.out.println(c + " is the largest number.");
        }
    }
}
}
```

**2.** Output:

```
Enter three numbers:
4.99 5.13 6.99
6.99 is the largest number.
```

**2.** Write a program to determine the sum of the following series for a given value of n:
1+½+⅓+ ...+1/n

```java
import java.util.Scanner;

public class SeriesSumCalculator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter a number: ");
        int n = scanner.nextInt();

        double sum = 0;

        for (int i = 1; i <= n; i++) {
            sum += 1.0 / i;
        }

        System.out.println("The sum of the series 1 + 1/2 + 1/3 + 1/4 + 1/" + n + " is: " + sum);
    }
}

```

Output:

```
Enter a number:
8
The sum of the series 1 + 1/2 + 1/3 + 1/4 + 1/8 is: 2.7178571428571425
```

**Conclusion:**

Thus we learned how to write a program using control loop statements.

**Signature of faculty in-charge with Date:**