

05/11/2015

The Logic Behind Drawing Triangles:

n=1

*

n=3

* *
* * *

n=5

* * *
* * * *
* * * * *

n=7

* * * * *
* * * * *
* * * * *
* * * * *
* * * * *

␣: Blank Space

(S)

number of
levels:
(integer division)

$$\frac{3}{2} + 1 = 2$$

$$\frac{5}{2} + 1 = 3$$

$$\frac{7}{2} + 1 = 4$$

n=1

*

n=3

* * *
* * *

n=5

* * * * *
* * * * *
* * * * *

n=7

* * * * *
* * * * *
* * * * *
* * * * *
* * * * *

Loop Writing Methods:

$$\text{Sum} = 1 + \frac{1}{2!} + \frac{1}{3!} + \dots + \frac{1}{n!}$$

Stop when we reach the limiting
value for n

element ≥ 0.001

— Java —

1) double sum, factor, element;

int i;

sum = 0.0;

i = 1;

element = 1;

do
{

sum = sum + element;

i++;

factor = 1.0;

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

factor = factor * j;

(28)

↳ element = 1.0 / factor;

05/11/2015

{ while (i <= n && element >= 0.001);
threshold

2)

$$\text{sum} = 1 - \frac{1}{2!} + \frac{1}{3!} - \frac{1}{4!} \dots$$

do
do

double mult = -1.0;

do
{

mult = -1.0 * mult;

sum = sum + mult * element;

i++;

factor = 1.0;

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

factor = factor * j;

}

element = 1.0 / factor;

} while (i <= n && element >= 0.001);

05/11/2015

Writing Method (like we did with max2):

```
main
{
    double val1, val2;
    val1 = SerialSum(5, 0.001);
    val2 = SerialSum(10);
}
```

actual parameters

you can't write like that!

```
public static double SerialSum(int n, double threshold)
{
    do
    {
        sum = sum + element;
        n++;
        factor = 1.0;
        for(int j = 1; j <= n; j++)
        {
            factor = factor * j;
        }
        element = 1.0 / factor;
    } while (i <= n && element >= threshold)

    return(sum);
}
```

formal parameters

method invocation
↳ method call

n	threshold
5	0.001

Function for factorial:

```
main
{
    double val1;
    value1 = Factorial(5);
}

public static double Factorial(5);
{
    double factor;
    factor = 1.0;
```

```
for(int j = 1; j <= n; j++) // if we define j as integer in here, it will be just for that loop
{
    factor = factor * j;
}
return(factor);
```

30