



# Assignment 1 Recursion

✚ **Topics** เรียนรู้การทำงานของฟังก์ชันที่มีการเรียกซ้ำ(Recursion)

✚ **Learning Outcomes**

- การใช้ IDE (Eclipse) สำหรับเขียนภาษาจาวา
- มีแนวคิดในการใช้การเรียกซ้ำ แทนการใช้คำสั่งวนรอบ

✚ **โจทย์** สร้างฟังก์ชัน factorial แบบ recursion ด้วยภาษาจาวา แล้วแทรกคำสั่งให้แสดงผลลัพธ์ ตามที่เงื่อนไขกำหนด (ไม่ให้มีการใช้คำสั่งวนรอบ)

- ค่าตัวแปรในฟังก์ชันก่อนที่มีการเรียกซ้ำ
- ค่าตัวแปรระหว่างการเรียกซ้ำ
  - รอบที่ของการเรียกซ้ำ
  - ค่าตัวแปรกรณี Base case
  - ค่าตัวแปรกรณี Recursive
- ค่าตัวแปรหลังจาก recursion ก่อนออก(return) จากฟังก์ชัน
- เคสตัวอย่างที่ใช้ในการทดสอบ(Test case) : 4, 2.0, 2. , x, 2x, -1, 20 , 0, 1

✚ **งานที่ต้องส่ง**

- source code (ต้องมีการแสดงชื่อผู้เขียนโปรแกรมทุกครั้งก่อนหรือหลังจบโปรแกรม)
- ไฟล์รายงานเป็น pdf ซึ่งประกอบด้วย source code คำอธิบายและผลการรันโปรแกรม
  - อธิบายหน้าจอลงผลการรันที่เหมือนกับ Test Case
  - อธิบายการทำงานและผลลัพธ์ ในแต่ละรอบของการ recursion

# Assignment 1 (Test Case)

## My Recursion Programs.

Program calculate n! by recursion (n<=15)

Enter n = 4

4! is recursive case. Answer = 4 \* recursion of 3!

Recursion to calculate 3!

3! is recursive case. Answer = 3 \* recursion of 2!

Recursion to calculate 2!

2! is recursive case. Answer = 2 \* recursion of 1!

Recursion to calculate 1!

1! is recursive case. Answer = 1 \* recursion of 0!

Recursion to calculate 0!

0! is base case return answer of 0! = 1

Calculate 0! complete.

Return answer from 0! = 1 to calculate 1! = [1 \* 0!] = 1 \* 1 = 1

Calculate 1! complete.

Return answer from 1! = 1 to calculate 2! = [2 \* 1!] = 2 \* 1 = 2

Calculate 2! complete.

Return answer from 2! = 2 to calculate 3! = [3 \* 2!] = 3 \* 2 = 6

Calculate 3! complete.

Return answer from 3! = 6 to calculate 4! = [4 \* 3!] = 4 \* 6 = 24

Complete calculation of 4! , answer = 24

press [y] to continue, others to exit. **y**

เข้าสู่ recursion

recursive case

base case 0!

ก่อนออกจาก recursion

ต้องจำคำตอบของ recursion  
ไว้เพื่อนำมาแสดงผลก่อน  
แสดงผลเสร็จค่อย return

สรุปคำตอบ



# *Assignment 1 (Test Case)*

```
Enter n = 2.0
Input error, please enter number between 0-15
Enter n = 2.
Input error, please enter number between 0-15
Enter n = x
Input error, please enter number between 0-15
Enter n = 2x
Input error, please enter number between 0-15
Enter n = -1
Input error, please enter number between 0-15
Enter n = 20
Input error, please enter number between 0-15
Enter n = 0
0! is base case return answer of 0! = 1
Calculate 0! complete.
Complete calculation of 0! , answer = 1
press [y] to continue, others to exit. y
Enter n = 1
1! is recursive case. Answer = 1 * recursion of 0!
    Recursion to calculate 0!
0! is base case return answer of 0! = 1
Calculate 0! complete.
    Return answer from 0! = 1 to calculate 1! = [1 * 0!] = 1 * 1 = 1
Complete calculation of 1! , answer = 1
press [y] to continue, others to exit. n
End Program.
Program writen by 620705010xx XXXXXXXXXXXX XXXXXXXXXXXX
```

# ตัวอย่างการใช้คำสั่งจาวา

✚ เรียกใช้ IDE ของภาษาจาวา เช่น Code Block

- สร้างคลาสหลักของโปรแกรม และตั้งชื่อไฟล์เป็นชื่อเดียวกับคลาสหลัก(\*.java)
- **import library (Class)** ที่จำเป็นต้องใช้
- ในคลาสจะต้องมี **method** ชื่อ **public static void main (String [] args)** เพื่อใช้สำหรับรันโปรแกรม (คล้ายการใช้ **int main()** ของภาษาซี)
  - กำหนดตัวแปร **object** เป็นชนิดของคลาส และสร้าง **object** เพื่อเรียกใช้คำสั่งของคลาสนั้น

ตัวอย่างคำสั่ง input/output ภาษาซี	ตัวอย่างคำสั่งภาษาจาวา
<code>#include &lt;stdio.h&gt; // ใช้ include stdio.h</code>	<code>import java.util.Scanner; // ใช้ import คลาส Scanner เพื่ออ่านคีย์บอร์ด</code>
<code>printf("strconst1 %d strconst2 %f\n",i,d); printf("strconst1 %d strconst2 %f\n",i,d);</code>	<code>system.out.printf("strconst1 %d strconst2 %f\n",i,d); //print format system.out.println("strconst1"+i+"strconst2"+d); //print concat</code>
<code>char str[40]; // ใช้อาร์เรย์ของ char แทนสตริง long long n; // %llu จำนวนเต็ม 64 bits int check = 0; // ใช้ int = 0 แทน false</code>	<code>String str; // มีตัวแปรประเภทสตริง long n = 0; // ควรกำหนดค่าเริ่มต้นให้ตัวแปรทุกครั้ง boolean check = false; // มีตัวแปรชนิด boolean ซึ่งมีค่าเป็น true , false</code>
<code>.... scanf("%d", &amp;i); scanf("%llu", &amp;n); scanf("%lf",&amp;d); scanf("%s",&amp;str); scanf("%ch", &amp;ch); gets(str); rewind(stdin); // ลบ buffer if (scanf("%d", &amp;n) == 0 ) // อ่าน int เก็บใน n printf("error\n"); // ถ้าอ่านไม่ได้แสดง error</code>	<code>Scanner in = new Scanner(System.in); // กำหนดให้ชื่อ in เป็น input stream i = in.nextInt(); // อ่านจำนวนเต็ม 32 bits n = in.nextLong(); // อ่านจำนวนเต็ม 64 bits d = in.nextDouble(); // อ่านจำนวนจริง 64 bits str = in.next(); // อ่าน string 1 ตัว ch = in.next().charAt(0); // อ่านสตริงและเช็คตัวอักษรที่ตำแหน่ง .. str = in.nextLine(); // อ่าน string 1 บรรทัด in.nextLine(); // อ่านให้หมดบรรทัด(ไม่ได้เก็บในตัวแปร) if (in.hasNextInt() ) n = in.nextInt(); //ถ้าตัวถัดไปอ่านเป็น int ให้เก็บใน n else System.out.println("error"); // ถ้าไม่ใช่ int แสดง error</code>
<code>long long factorial(int n) int main()</code>	<code>public static long factorial(int n) // public static เป็นการกำหนดการมองเห็น public static void main(String[] args) // ฟังก์ชันที่ใช้ในการรันโปรแกรม</code>

# เปรียบเทียบคำสั่ง C กับ JAVA

```
long long factorial(int n){  
    long long x;  
    if (n==0){ // Base Case  
        return 1;  
    } else { // Recursive Case  
        x=n*factorial(n-1);  
        return x;  
    }  
}
```

```
// ใช้ static เพื่อให้ method factorial ถูกสร้างไว้ล่วงหน้า จะได้เรียกใช้ได้ทันที ไม่ต้องเรียกผ่าน instance (ไม่ต้องสั่ง new)  
static long factorial(int n) {  
    long x;  
    if (n == 0) { // Base Case  
        return 1;  
    } else { // Recursive Case  
        x = n * factorial(n - 1);  
        return x;  
    }  
}
```

# เปรียบเทียบคำสั่ง C กับ JAVA

```
int main() {
    int n;
    char ch;
    long long ans;
    do { printf("My Recursion Programs.\n");
        printf("Program calculate n! by recursion.\n");
        n=getint("Enter n : ",0,15);
        ans=factorial(n);
        printf("Complete calculation of %d! , answer = %llu\n",n,ans);
        printf("press[y] to continue, others to exit. \n");
        scanf("%c",&ch);
    } while(ch=='y' || ch=='Y');
    return 0;
}
```

```
#include <stdio.h>
public class Recursion {
    .....
    int main() {
        .....
    }
}
```

```
public static void main(String[] args) {
    // TODO Auto-generated method stub
    int n;
    char ch=0;
    long ans;
    Scanner in = new Scanner(System.in);
    do {
        System.out.printf("My Recursion Programs.\n");
        System.out.printf("Program calculate n! by recursion.\n");
        n = getint("Enter n = ", 0, 15);
        ans = factorial(n);
        System.out.printf("Complete calculation of %d! , answer = %d\n", n, ans);
        System.out.printf("press[y] to continue, others to exit. \n");
        if (in.hasNext())
            ch = in.next().charAt(0);
    } while (ch == 'y' || ch == 'Y');
    System.out.printf("End Program.\n");
}
```

```
import java.util.Scanner;
import java.io.IOException;
import java.lang.System;
public class Recursion {
    .....
    public static void main(String[] args) {
        .....
    }
}
```

# เปรียบเทียบคำสั่ง C กับ JAVA

```
int getint(char *msg, int min, int max)
{ int a; char ch=0;
  do { printf("%s",msg);
        scanf("%d%c",&a,&ch);
        if (ch !='\n' || a<min || a>max)
            printf("Input Error,Please Enter number between %d - %d\n", min,max);
        rewind(stdin);
    } while (ch!='\n' || a<min || a>max);
  return a;
}
```

```
static int getint(String msg, int min, int max) { // java มีชนิดข้อมูลเป็น String ไม่กำหนดความยาว
  int a = 0; //เมื่อจองตัวแปรควรกำหนดค่าเริ่มต้นทุกครั้ง
  boolean success = false; //java มีชนิดข้อมูล boolean มีค่าเป็น true หรือ false
  Scanner stdin = new Scanner(System.in); // สร้าง object ของ Scanner ชื่อ stdin เพื่อใช้อ่านข้อมูล
  while (!success) { //วนรอบขณะที่ยังอ่านไม่สำเร็จ
    System.out.print(msg); // คำสั่งแสดงผลแบบไม่ระบุ format (.print, .println)
    try { //ใช้ try {...} catch() {...} ถ้าเกิดความผิดพลาดใน try {...} จะกระโดดไปทำส่วน catch() {...}
      a = stdin.nextInt(); // อ่านตัวเลขจำนวนเต็ม เก็บใน a อาจเกิด error ได้ในบรรทัดนี้
      if (a >= min && a <= max)
        success = true;
    } else
      System.out.printf("Input error, please enter between %d - %d\n", min, max);
  } catch (Exception err) { // ถ้าอ่าน a ไม่สำเร็จจะกระโดดมาทำส่วน catch ,err คือประเภทความผิดพลาดที่เกิด
    System.out.printf("Input error, please enter between %d - %d\n", min, max);
    stdin.nextLine(); // อ่านทิ้งจนหมดบรรทัด
  }
}
return a;
}
```