



ภาษาการเขียนโปรแกรม

WEEK3

02-212-213

อ.ธิดาวรรณ คล้ายศรี



- 1.1 An Introduction to Programming (บทนำการเขียนโปรแกรม)
- 1.2 Traditional Programming Languages (ภาษาการเขียนโปรแกรมแบบดั้งเดิม)
- 1.3 Syntax of the Programming Languages (ไวยากรณ์ของภาษาเขียนโปรแกรม)
 - 1.3.1 Control Block (การควบคุมบล็อก)
 - 1.3.2 Block Structure (โครงสร้างบล็อก)
 - 1.3.3 Recursive (การทำงานแบบเรียกตัวเอง)
- 1.4 Review Data Structure (ทบทวนโครงสร้างข้อมูล)
- 1.5 Sorting and Searching Algorithms (อัลกอริทึมจัดลำดับและสืบค้นข้อมูล)
- 1.6 Testing Programs (การทดสอบโค้ด)
 - 1.6.1 Compiler
 - 1.6.2 Running Time and Debugging
 - 1.6.3 Creating Test Cases



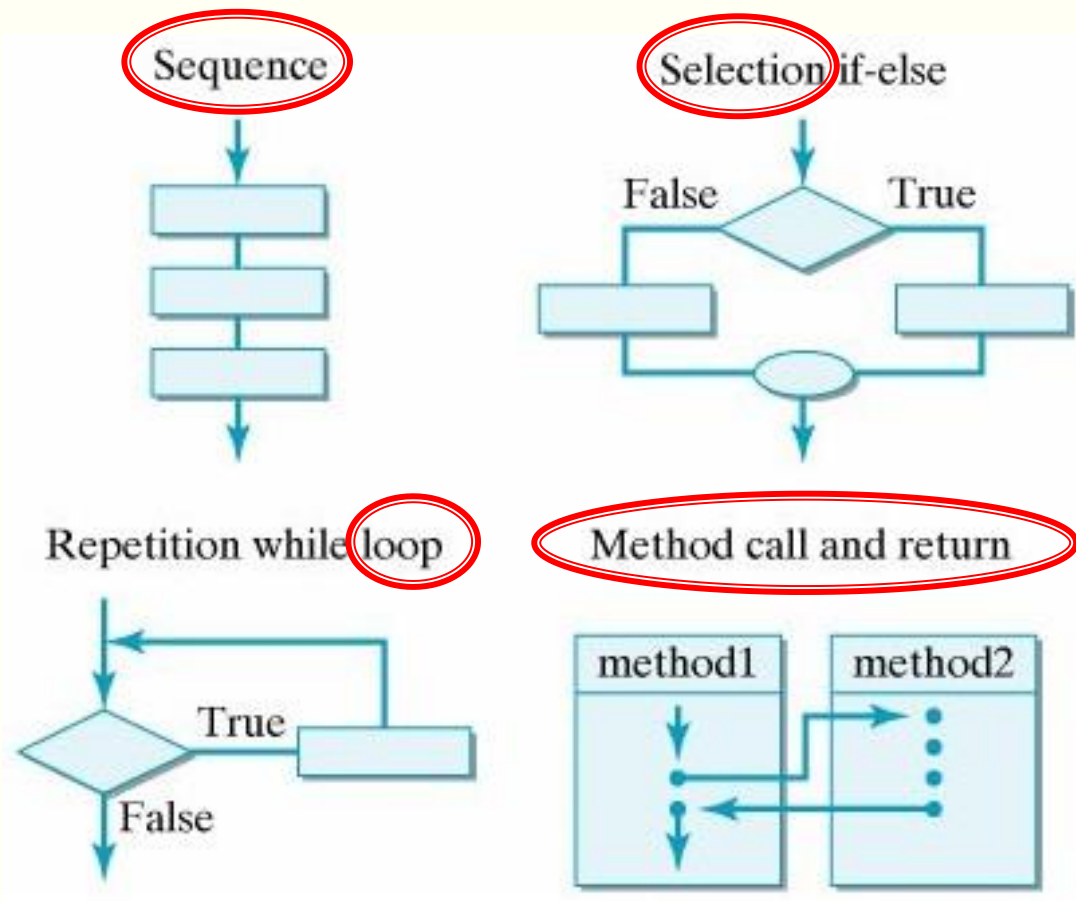
1.3 Syntax of Programming Languages

ไวยากรณ์ของภาษาการเขียนโปรแกรม

ตัวอย่างโจทย์ปัญหา 5 กลุ่ม

1. โปรแกรมแจ้งเตือนน้ำหนักร่างกาย (ว่าอยู่ในเกณฑ์ปกติ/พอม/อ้วน)
2. โปรแกรมคำนวณรายได้สุทธิของพนักงานในแผนกหนึ่งๆ หลังจากหักภาษีรายได้บุคคลธรรมดา
3. โปรแกรมแจ้งโปรโมชั่นพิเศษจากระดับคะแนนสะสมในบัตรสมาชิกซื้อสินค้า เช่น The One Card, M-card, Boots
4. โปรแกรมแจ้งสิทธิพิเศษจากการใช้บริการต่างๆ เช่น Grab, Loyalty card ที่มีมูลค่าเป็นเงิน/ใช้บริการฟรี
5. โปรแกรมคิดยอดเงินชำระในการสั่งซื้อสินค้าออนไลน์ (อาจมีค่าจัดส่ง/ช่องทางของการชำระเงิน) เช่น Check out ตะกร้าสินค้าในลาซาด้า

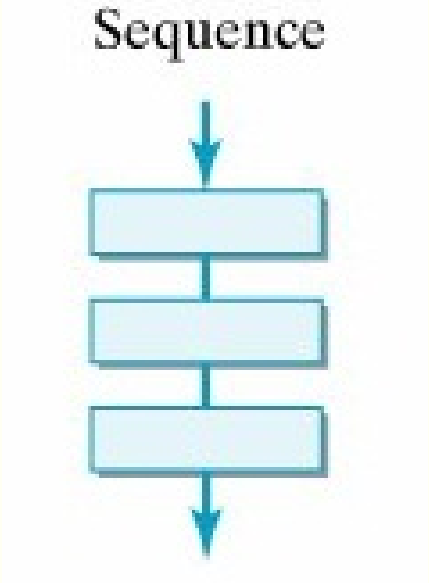
1.3.1-1.3.2 การควบคุม Block and โครงสร้าง Block



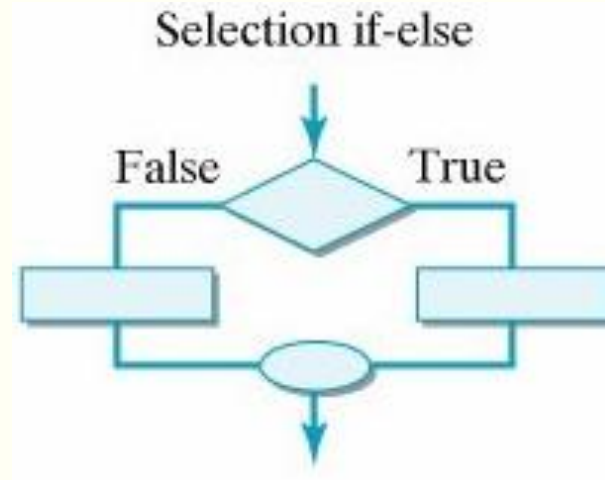
- พื้นฐานของการเขียนโปรแกรม=?
- โปรแกรมจะประกอบไปด้วย 3 Structures นี้
- Structures: Sequence, Selection & Repetition/Loop
- การเรียกใช้ method ในอีก method หนึ่งเพื่อลดความซับซ้อน ไม่เป็นระเบียบ/โครงสร้าง

- กำหนดการทำงานแบบเป็นลำดับ
- การรับข้อมูลเข้า field-รหัสพนักงาน
- การรับข้อมูลเข้า field-เงินเดือน
-

```
instruction 1;  
instruction 2;  
instruction 3;  
...
```

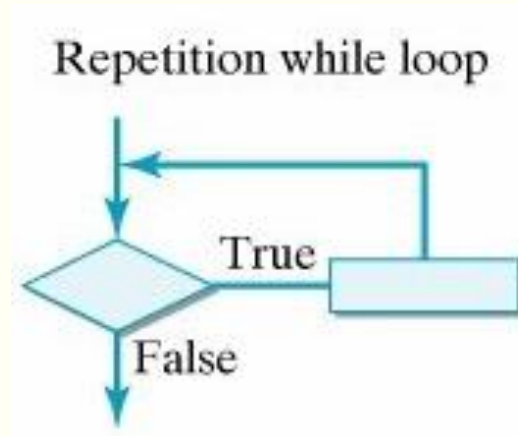


- ประมวลผลด้วยการกำหนดทางเลือกเพื่อให้เกิดเงื่อนไขเลือกทำงาน
- ระดับเงินเดือน-เกณฑ์คิดภาษี
- ระดับสะสมคะแนน-กำหนดโปรโมชั่น
-



```
IF condition is true THEN
    do this;
ELSE
    do that;
ENDIF
```

- ปริมาณของงานมากแต่กระบวนการเหมือนเดิม
- คำนวณรายได้สุทธิพนักงานจำนวนมาก
- ...



WHILE (more items to process)

process the next item;

ENDWHILE

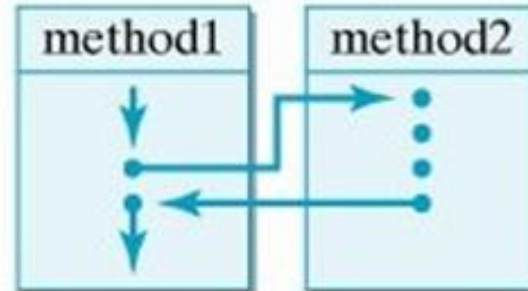
FOR month = 1 to 12

do monthly processing

ENDFOR

- การเรียกคำสั่งระหว่าง Methods

Method call and return





ฝึกปฏิบัติ

โจทย์ตัวอย่าง-จงหา factorial ของจำนวนเต็มบวกใดๆ

นิยามของ $N!$ คือ $N * (N-1) * (N-2) * \dots * 3 * 2 * 1$ โดยที่ $n \geq 0$

- เมื่อให้ $N = 3$; $3! = 3*2*1 = 6$
- เมื่อให้ $N = 4$; $4! = 4*3*2*1 = 24$
- เมื่อให้ $N = 5$; $5! = 5*4*3*2*1 = 120$

The image shows an IDE window with the following components:

- Project Explorer (Left):**
 - Project: JavaWeek3_SyntaxProgramming
 - Source Packages
 - FactorialCalculation
 - Factorial_Version1.java
 - Factorial_Version2.java
 - Test Packages
 - <default package>
 - Test_Factorial_Version1.java
 - Test_Factorial_Version2.java
 - Libraries
 - Test Libraries
- Code Editor (Right):**
 - File: Factorial_Version1.java
 - Code:

```
10  /**
11  *
12  * @author TEE
13  */
14  public class Factorial_Version1 {
15
16      int N;    //input number
17      int fact; //output factorial number
18
19      /**1) Input-a number N
20      *
21      */
22      public void inputNumber() {
23          System.out.println("Enter a number N: ");
24          Scanner in = new Scanner(System.in);
25          N = in.nextInt();
26      }
27  }
```

Lab-ปฏิบัติ Sequence Statements

```
/**1) Input-a number N
 *
 */
public void inputNumber() {
    System.out.println("Enter a number N: ");
    Scanner in = new Scanner(System.in);
    N = in.nextInt();
}
```

Lab-ปฏิบัติ If Statements

```
/**2) Calculating factorial of the N
 *
 */
public void factorial() {
    if (N < 0) {
        System.out.println("Number should be non-negative.");
    }

    else
    {
        for (int i = 1; i <= N; i++) {
            fact = fact * i;
        }
    }
}
```

Lab-ปฏิบัติ Loop

```
/**2) Calculating factorial of the N
 *
 */
public void factorial() {
    if (N < 0) {
        System.out.println("Number should be non-negative.");
    }

    else
    {
        for (int i = 1; i <= N; i++) {
            fact = fact * i;
        }
    }
}
```

Lab-ปฏิบัติ Method Call and Return

```
/**1) Input-a number N
```

```
*  
*/
```

```
public void inputNumber() {  
    System.out.println("Enter a number N: ");  
    Scanner in = new Scanner(System.in);  
    N = in.nextInt();  
}
```

```
/**2) Calculating factorial of the N
```

```
*  
*/
```

```
public void calculatingFactorial() {
```

```
    inputNumber();
```

```
    if (N < 0) {
```

```
        System.out.println("Number should be non-negative.");
```

```
    }
```

```
    else
```

```
    {
```

```
        for (int i = 1; i <= N; i++) {
```

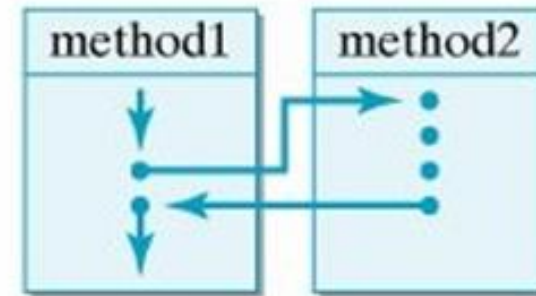
```
            fact = fact * i;
```

```
        }
```

```
    }
```

```
}
```

Method call and return



1.3.3 การทำงานแบบเรียกตัวเอง (Recursive)

คือ โปรแกรมย่อย/function ที่เรียกใช้งานตัวเอง

หลักการรีเคอร์ซีฟฟังก์ชัน → เขียนโปรแกรมเรียกตัวเอง เพื่อลดปัญหาของโปรแกรมที่ซับซ้อน

จะมีการ return ค่ากลับในฟังก์ชันเพื่อทำงานเรียกตัวเองเมื่อ base case ที่ทำงานซ้ำๆ กัน

- เมื่อให้ $N = 3$; $3! = 3*2*1 = 6$
- เมื่อให้ $N = 4$; $4! = 4*3*2*1 = 24$
- เมื่อให้ $N = 5$; $5! = 5*4*3*2*1 = 120$

โจทย์ตัวอย่าง-หา factorial ของจำนวนเต็มบวกใดๆ

นิยามของ $N!$ คือ $N * (N-1) * (N-2) * \dots * 3 * 2 *$

โดยที่ $n \geq 0$

$$n! = n * (n-1)!$$

- เมื่อให้ $N = 3$; $3! = 3*2*1 = 6$
- เมื่อให้ $N = 4$; $4! = 4*3*2*1 = 24$
- เมื่อให้ $N = 5$; $5! = 5*4*3*2*1 = 120$

```
/**2) (Recursive) Calculating factorial of the N
 *
 * @param N input number
 * @return factorial of N
 */
public int factorial(int N) {
    inputNumber();
    if (N < 0) {
        System.out.println("Number should be non-negative.");
    }
    if (N==1) {
        fact = 1;
    }
    else {
        fact = N*factorial(N-1);
    }
    return fact;
}
```

สรุป-บททวน

```
22  [-] public void inputNumber() {
23      System.out.println("Enter a number N: ");
24      Scanner in = new Scanner(System.in);
25      N = in.nextInt();
26  }
27
28  [-] /**2) Calculating factorial of the N
29      *
30      */
31  [-] public void factorial() {
32
33      inputNumber();
34      if (N < 0) {
35          System.out.println("Number should be non-negative.");
36      }
37
38      else
39      {
40          for (int i = 1; i <= N; i++) {
41              fact = fact * i;
42          }
43      }
44  }
```



1.4 A Brief Review of Data Structure

ทบทวนโครงสร้างข้อมูล

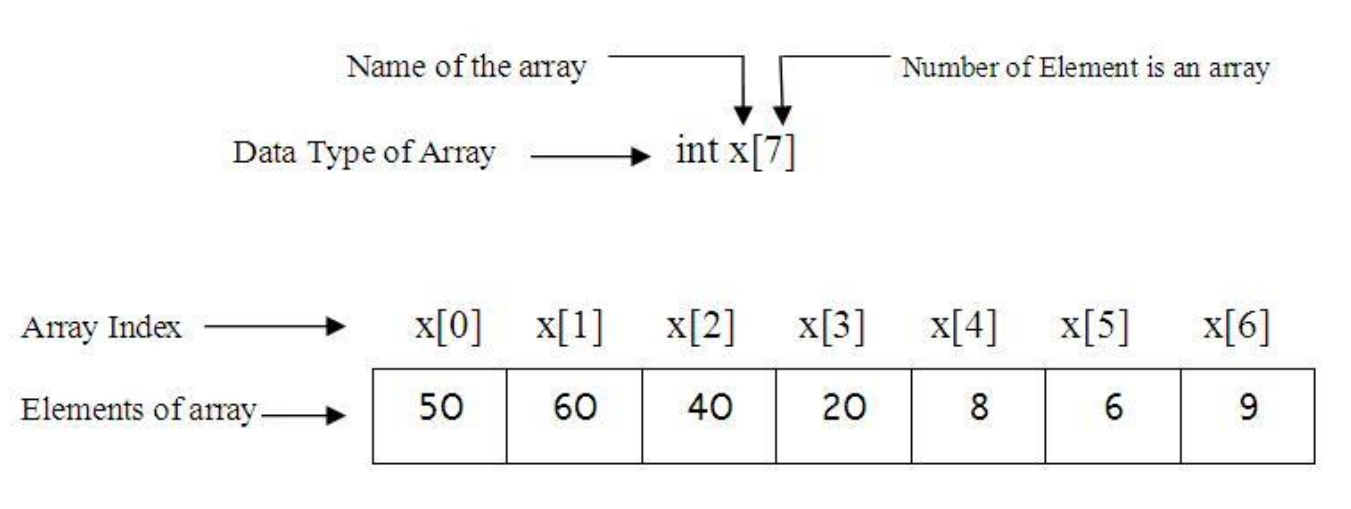
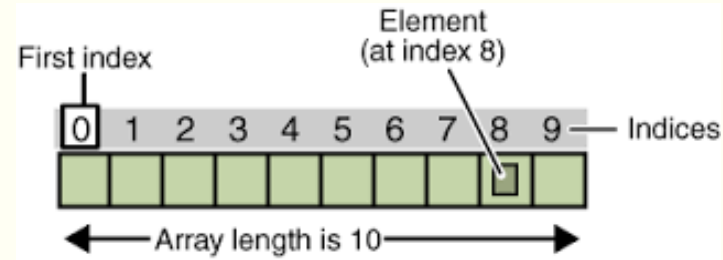
Primitive Data Types (ประเภทข้อมูลแบบพื้นฐาน)

- Numbers
 - เลขจำนวนเต็ม int/Integer/Long Integer
 - เลขทศนิยม Floating/ Doubles
- String
 - ตัวอักษร Character, 'a'
 - ข้อความ Strings, "Java"
- Boolean
 - True/False
 - 1/0

Other Data Types (ประเภทข้อมูลแบบอื่นๆ)

- ถ้าต้องการเก็บรหัสพนักงานหลายๆ คน ทำ?
- ถ้าต้องการเก็บคะแนนคน. หลายๆ คน ทำ?
- ถ้าต้องการเก็บคะแนน. หลายๆ คน และหลายๆ วิชา ทำ?
- What is the common thing here?
- ใช้ข้อมูลแบบไหน?

Collection / Arrays

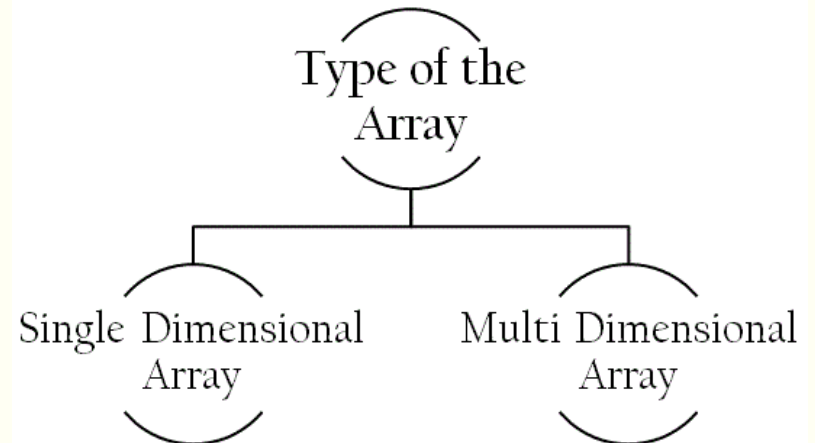


Other Data Types (ประเภทข้อมูลแบบอื่นๆ)

- ถ้าต้องการเก็บคะแนนคนหลายๆ คน ทำ?
- What is the common thing here? ใช้ข้อมูลแบบไหน?

Collection / Arrays

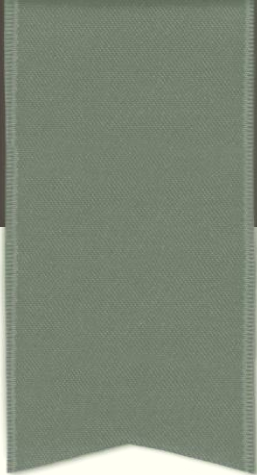
	Column 0	Column 1	Column 2	Column 3
Row 0	a[0][0]	a[0][1]	a[0][2]	a[0][3]
Row 1	a[1][0]	a[1][1]	a[1][2]	a[1][3]
Row 2	a[2][0]	a[2][1]	a[2][2]	a[2][3]



Arrays (แถวลำดับ)

- enable us to store data of **similar type together**. เก็บข้อมูลประเภทเดียวกันไว้ด้วยกัน
- enables us to **handle varying size data**. ขนาดของข้อมูลอาจต่างกันได้ในแถวลำดับ
- Lines of code do not increase with more data! ข้อมูลเยอะแต่ก็เก็บไว้ในตัวแปรเดียว

```
FOR each item in array  
    add item to total  
ENDFOR
```

ฝึกปฏิบัติ

แบบฝึกหัดกลุ่ม

- แบ่งกลุ่ม 5 กลุ่ม
- ทำการวิเคราะห์งานที่ได้รับมอบหมาย โดยลือไปตาม
 - The Input-Process-Output (IPO) model ที่ใช้ในการวิเคราะห์ระบบ (Systems Analysis) และในวิศวกรรมซอฟต์แวร์ (Software Engineering)
 - SDLC วงจรการพัฒนาซอฟต์แวร์
- ทำสไลด์เพื่อจัดนำเสนอการวิเคราะห์ปัญหางาน บรรยายสั้นๆ หน้าชั้นเรียน ประมาณ 5-7 นาที

โจทย์ปัญหา 5 กลุ่ม

1. โปรแกรมแจ้จมวลน้ำหน้าร่างกาย (ว่าอยู่ในเกณฑ์ปกติ/พอม/อ้วน)
2. โปรแกรมคำนวณรายได้สุทธิของพนักงานในแผนกหนึ่งๆ หลังจากหักภาษีรายได้บุคคลธรรมดา
3. โปรแกรมแจ้จโปรโมชั่นพิเศษจากระดับคะแนนสะสมในบัตรสมาชิกซื้อสินค้า เช่น The One Card, M-card, Boots
4. โปรแกรมแจ้จสิทธิพิเศษจากการใช้บริการต่างๆ เช่น Grab, Loyalty card ที่มีมูลค่าเป็นเงิน/ใช้บริการฟรี
5. โปรแกรมคิดยอดเงินชำระในการสั่งซื้อสินค้าออนไลน์ (อาจมีค่าจัดส่ง/ช่องทางของการชำระเงิน) เช่น Check out ตะกร้าสินค้าในลาซาด้า