

Basic program design

ฉัตรชัย เกษมทวีโชค

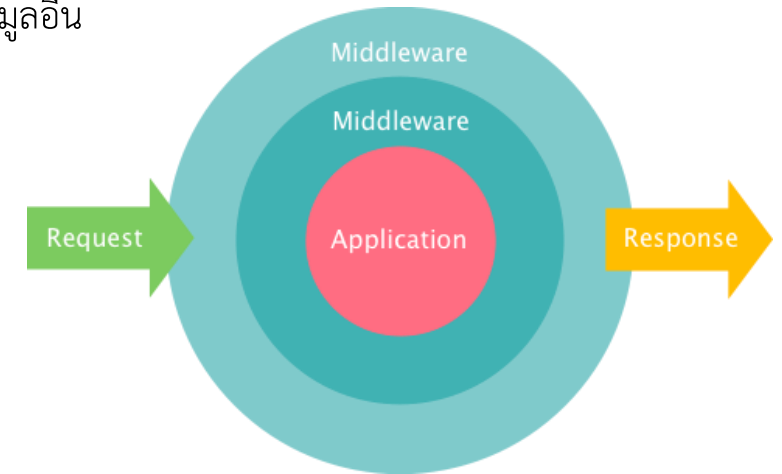
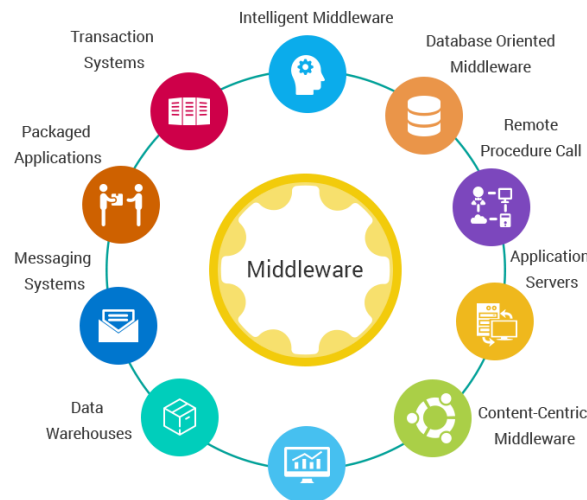
Chatchai.kase@ku.th

การสั่งงานคอมพิวเตอร์

- คอมพิวเตอร์เป็นอุปกรณ์อิเล็กทรอนิกส์ ในการสั่งให้คอมพิวเตอร์ทำงานจะต้องป้อนคำสั่งที่เครื่องคอมพิวเตอร์เข้าใจให้กับมัน
- **โปรแกรมคอมพิวเตอร์ (Software)** เป็นเหตุผลหลักที่ทำให้มนุษย์ซื้อคอมพิวเตอร์ คำสั่งของโปรแกรมสั่งการทำงานของอุปกรณ์ (Hardware) ในเครื่องคอมพิวเตอร์ โดยเฉพาะหน่วยประมวลผลกลาง (CPU)
 - ตัวอย่างเช่น คำสั่งในโปรแกรมสามารถสั่งให้ CPU ดึงข้อมูลจากหน่วยความจำหลัก (RAM) หรือ
 - ถ้าโปรแกรมต้องการสั่งให้ CPU เล่นไฟล์เพลง โปรแกรมจะทำการสร้าง interrupt
- **Interrupt** เป็นสัญญาณ (signal) ที่ไปยัง CPU เพื่อสั่งให้ประมวลผล (Execute) ชุดคำสั่งที่มีการเตรียมไว้แล้วเพื่อตอบสนองสัญญาณ interrupt
 - ในตัวอย่างนี้ หมายถึงอุปกรณ์จะส่งไฟล์เพลงไปยังอุปกรณ์กระจายเสียง อาทิเช่น ลำโพง หรือหูฟัง เป็นต้น

โปรแกรมคอมพิวเตอร์

- โปรแกรมคอมพิวเตอร์ (computer Program) เป็นชุดคำสั่งหรือรูปแบบบรรทัดคำสั่ง (codes) เพื่อใช้งานให้เครื่องคอมพิวเตอร์ประมวลผลตามที่ต้องการ โดยแบ่งออกเป็น 3 ประเภท
 - ระบบปฏิบัติการ (Operating System หรือ System software) เป็นโปรแกรมที่ออกแบบมาเพื่อใช้งานฮาร์ดแวร์ของคอมพิวเตอร์ และจัดเตรียมและสร้างระบบพื้นฐานสำหรับใช้งานโปรแกรมประยุกต์อาทิเช่น Windows, MacOS, Linux, Unix เป็นต้น
 - โปรแกรมตัวกลาง (Middleware software) คือโปรแกรมคอยช่วยเหลือดูแลโปรแกรมประยุกต์ที่รันอยู่บน OS หรือจะเรียกว่าตัวเชื่อมระหว่าง application และ operating system ก็ได้ ซึ่งมันช่วยให้ผู้พัฒนาโปรแกรมสามารถเชื่อมต่อสื่อสารกับภายนอกได้ง่ายขึ้น ทำให้ลดภาระในการดูแลรายละเอียดรอบข้างและเน้นแต่งงานหลักที่ต้องการ
 - ประกอบชุดของบริการที่ช่วยให้สามารถทำงานได้หลายกระบวนการ บนเครื่องหนึ่งเครื่องหรือมากกว่าเพื่อประสานงาน สนับสนุนและทำให้การกระจายงานง่ายขึ้น นอกจากนี้ยังสามารถให้ข้อมูลมีอยู่ในฐานข้อมูลเดียวที่สามารถเข้าถึงได้ผ่านทางฐานข้อมูลอื่น



Source: <http://www.girnarsez.com/middleware>

<https://dracony.org/wp-content/uploads/2015/03/middleware.png>

โปรแกรมคอมพิวเตอร์

- โปรแกรมประยุกต์ (application software) ถูกออกแบบมาเพื่อช่วยให้ผู้ใช้งานหนึ่ง ๆ หรือหลายอย่างที่เกี่ยวข้องกัน แอปพลิเคชันสามารถจัดการกับข้อความ ตัวเลข กราฟฟิคหรือการรวมกันขององค์ประกอบเหล่านี้ขึ้นอยู่กับงานที่ออกแบบไว้ อาทิเช่น office suites, web browsers, video games, media players



<https://www.cygnismedia.com/web-application/web-app-development.html>



<https://www.agriya.com/services/mobile-application-development>



<https://fossbytes.com/best-web-browsers-for-pc/>

Programmer

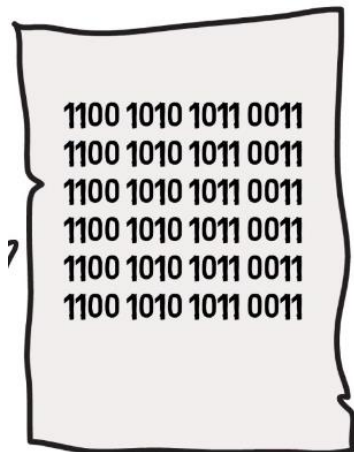
- ผู้คนที่ผ่านการฝึกฝนขั้นสูงที่เรียกว่า คอมพิวเตอร์โปรแกรมเมอร์ (Computer Programmer) จะเป็นผู้ที่สร้างรายการคำสั่ง (instruction) ในโปรแกรม
- รายการคำสั่งดังกล่าวเราเรียกว่า “code”
- และกระบวนการในการเขียนคำสั่ง เราเรียกว่า “coding”
- <https://www.youtube.com/watch?v=YF1ljAeTbq0>

คำสั่งคอมพิวเตอร์ (Code)

- **Code** : เป็นคำศัพท์ที่แทนความหมายของชุดบรรทัดคำสั่งที่ถูกเขียนด้วยภาษาคอมพิวเตอร์ใดก็ได้ เช่น ภาษาเครื่อง ภาษาระดับสูง
- เครื่องคอมพิวเตอร์คิดและสื่อสารในระบบเลขฐานสอง แต่ระบบเลขฐานสองที่ซับซ้อนและยุ่งยากเกินไปสำหรับมนุษย์ที่สามารถเข้าใจหรือทำงานด้วยเป็นเวลานาน
- เป็นเหตุผลที่ทำให้เกิดภาษาคอมพิวเตอร์ขึ้นเพื่อใช้ประโยชน์ในการเขียนโปรแกรมหรือชุดคำสั่งเพื่อใช้สั่งการคอมพิวเตอร์ทำงาน

ภาษาคอมพิวเตอร์ (Programming Language)

- การนำคำสั่งมาเรียงต่อกันให้ทำงานอย่างใดอย่างหนึ่งเรียกว่า โปรแกรมภาษาที่คอมพิวเตอร์เข้าใจเรียกว่า ภาษาเครื่อง (Machine Language) ซึ่งเป็นเลขฐานสอง
- ต่อมาได้มีการออกแบบตัวอักษรภาษาอังกฤษให้แทนคำสั่งเลขฐานสอง เรียกว่า ภาษาโปรแกรม (Programming Language)
 - ภาษาระดับต่ำ (Low-level Language) คือภาษา Assembly ต้องแปลภาษาเป็นภาษาเครื่องด้วย Assembler
 - ภาษาระดับสูง (High-level Language) เช่น ภาษา C, Pascal, Basic, php, Python, Objective C, R, เป็นต้น



Source: <http://www.androidauthority.com> (Youtube)

```
// i = 15;
mov r3, #15
str r3, [r11, #-8]

// j = 25;
mov r3, #25
str r3, [r11, #-12]

// i = i + j;
ldr r2, [r11, #-8]
ldr r3, [r11, #-12]
add r3, r2, r3
str r3, [r11, #-8]
```

```
1 <?php
2 function answerDistribution($dbconn, $date) {
3     $result = mysqli_query($dbconn, "SELECT * FROM TechTerms WHERE Date='$date'");
4     if($result != false) {
5         $distArray = array();
6         $row = mysqli_fetch_assoc($result);
7         $correctAnswer = $row['Correct'];
8         $distArray['A'] = $row['Anum'];
9         $distArray['B'] = $row['Bnum'];
10        $distArray['C'] = $row['Cnum'];
11        $distArray['D'] = $row['Dnum'];
12        $distArray['Correct'] = $correctAnswer;
13        $distArray['Answer'] = rtrim($row['CorrectAnswer'], ".");
14        $distArray['Query'] = "SELECT * FROM TechTerms WHERE Date='$date'";
15        return $distArray;
16    } else {
17        $distArray['Error'] = 'Quiz load query failed';
18        return $distArray;
19    }
20 }
21 ?>
```

Source: <https://techterms.com/>

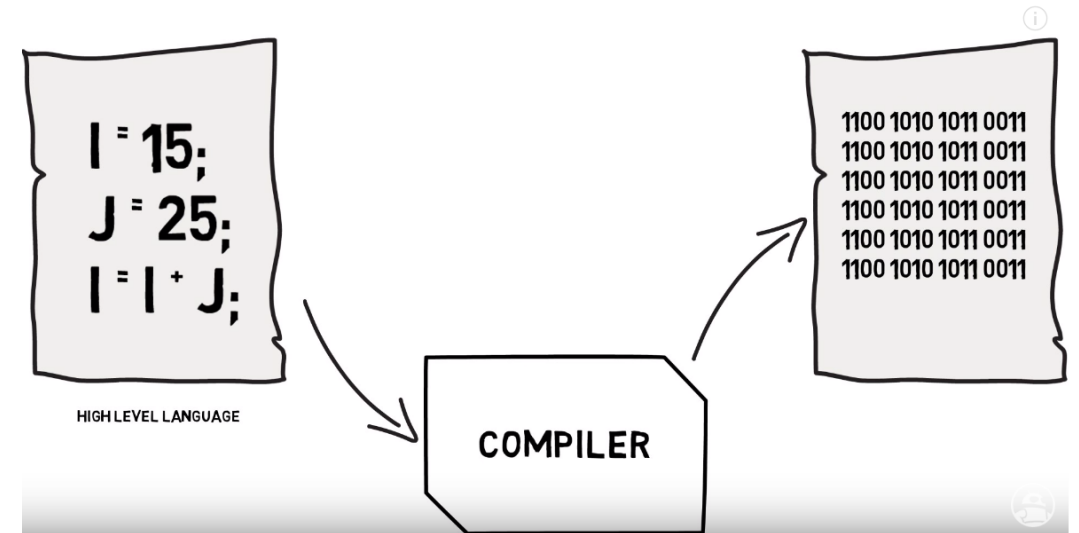
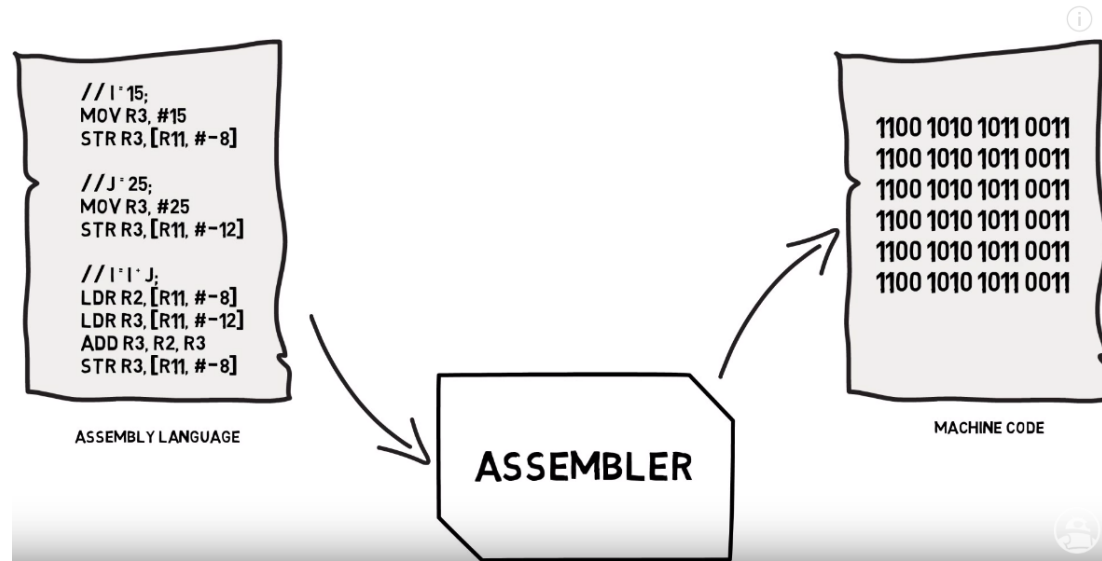
ตัวแปลภาษาระดับสูง

- **ตัวแปลภาษาระดับสูง** : เป็นตัวแปลภาษาระดับสูงเป็นภาษาเครื่อง เราสามารถแบ่งตัวแปลภาษาประเภทนี้ออกเป็น 2 ประเภทคือ
 1. **Compilers** เป็นการแปลคำสั่งโปรแกรม source code ทั้งหมดไปเป็นภาษาเครื่อง เพื่อเป็นไฟล์ที่ประมวลผลได้ (Executable file)
 - ในบางภาษาจะต้องมีการแปลเป็นไฟล์ Object Code ก่อนแล้วลิงก์กับไฟล์ภายนอกที่เกี่ยวข้องเพื่อสร้างเป็นไฟล์ .exe (Executable file)
 - ในบางภาษาจะใช้ไฟล์ Object Code ในการประมวลผลได้โดยตรง
 2. **Interpreters** เป็นการแปลคำสั่งภาษาคอมพิวเตอร์เป็นภาษาเครื่องแบบที่ละบรรทัดคำสั่งแล้วนำไปประมวลผลทันที (on fly) โดยไม่ต้องสร้างไฟล์ .exe (Executable file)
 - ข้อดี ยืดหยุ่นในการทำงานและการแก้ไขโปรแกรมที่สูงกว่าแบบ Compilers
 - ข้อเสีย ทำงานน้อยกว่าแบบ Compilers เพราะจะแปลก่อนแล้วค่อยทำ และต้องมีตัวแปลภาษา (Interpreters) ไปด้วยเสมอ

Assembler & Compiler

<https://www.youtube.com/watch?v=wA2oMRmbrfo>

<https://www.youtube.com/watch?v=1OukpDfsuXE>



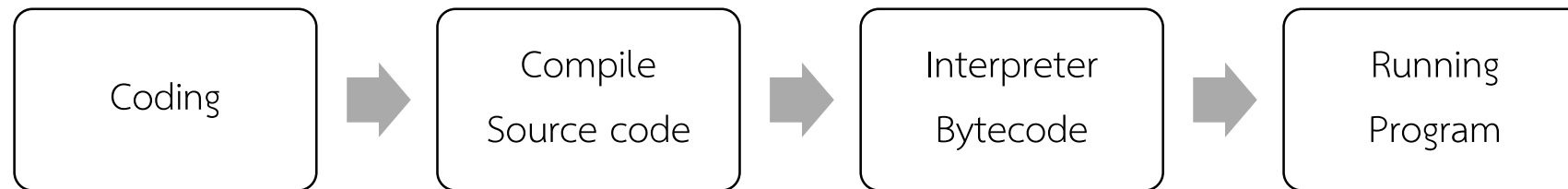
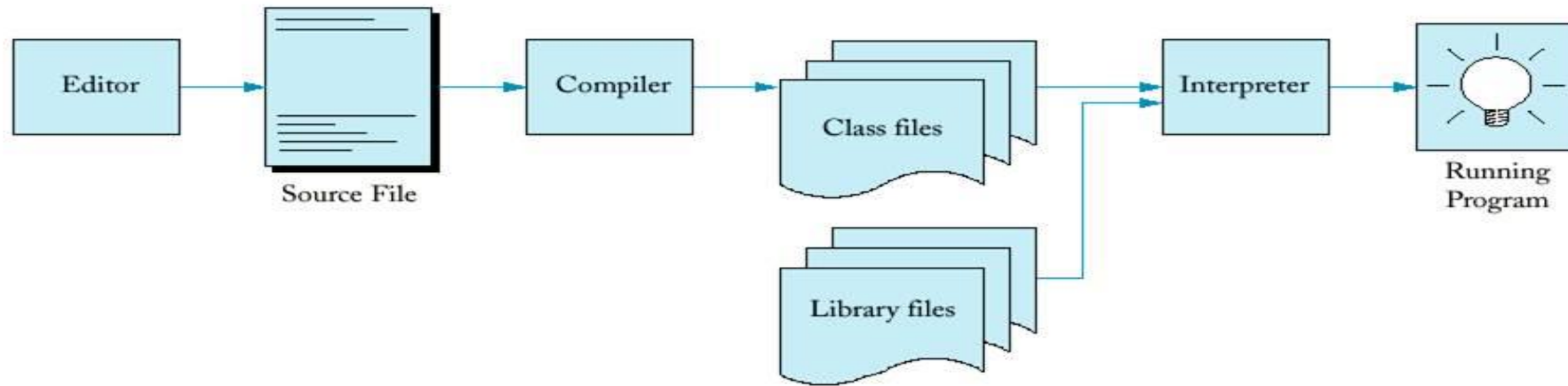
Assembler เป็นตัวแปลภาษา Assembly ไปเป็นไฟล์ภาษาเครื่องที่ใช้ในการรันการทำงาน

Compiler เป็นตัวแปลภาษาชั้นสูงไปเป็นไฟล์ภาษาเครื่องที่สามารถรันการทำงานได้ก็ครั้งก็ได้

Interpreter เป็นตัวแปลภาษาชั้นสูงที่ละบรรทัดคำสั่งไปสั่งงานคอมพิวเตอร์ทันที

Source: <http://www.androidauthority.com> (Youtube)

Hybrid method (Java program)



Program Design

- ก่อนที่จะลงมือเขียนโปรแกรมจริงๆนั้น การออกแบบเพื่อแสดงขั้นตอนการทำงานเอาไว้ก่อนเป็นสิ่งสำคัญอย่างมาก ทำให้การลงมือเขียนโปรแกรมทำได้ง่ายขึ้น และยังทำให้การกลับมาศึกษาโปรแกรมภายหลังทำได้ง่ายขึ้นด้วย
- เราเรียกขั้นตอนการทำงานว่า “Algorithm (อัลกอริทึม)” โดยอาจเขียนอัลกอริทึมในลักษณะของผังงาน (Flowchart) หรือรหัสจำลอง (Psuedocode) ก็ได้ ขึ้นอยู่กับความเหมาะสมและความถนัด

WHAT IS A Algorithm


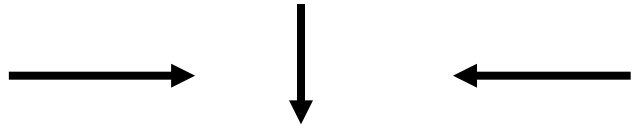

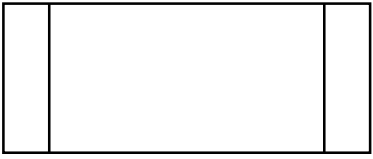
1. ลำดับการทำงานที่ตรงไปตรงมาไม่ซับซ้อน
2. มีควบคุมการไหลก่อนหลังของคำสั่งที่ต้องถูกประมวลผล
3. เมื่อการระบุจะหยุดการทำงานเมื่อไร

$1+2+3 = a \text{ algorithm}$

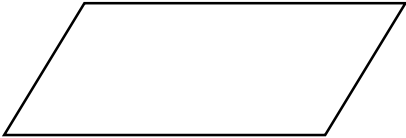
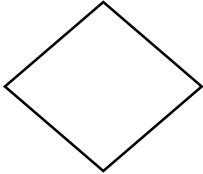

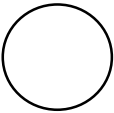
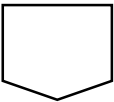
การเขียนผังงาน (Flowchart)

- เป็นแผนภาพที่ใช้อธิบายการทำงานของโปรแกรมโดยอาศัยรูปทรงต่างๆ ควบคู่ไปกับลูกศรแสดงทิศทางการทำงานจากขั้นตอนหนึ่งสู่อีกขั้นตอนหนึ่ง
- ประโยชน์ของผังงาน
 1. ช่วยอธิบายลำดับขั้นตอนการทำงานของโปรแกรม
 2. ทำให้ตรวจสอบข้อผิดพลาดของโปรแกรมได้ง่าย
 3. ทำให้ผู้อื่นสามารถศึกษาการทำงานของโปรแกรมและแก้ไขโปรแกรมได้ง่าย

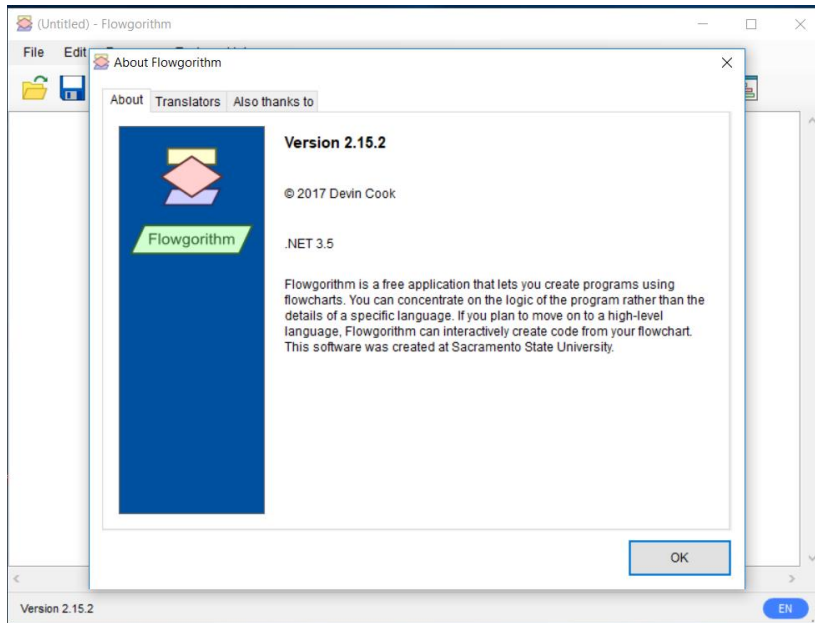
สัญลักษณ์ที่ใช้ในการเขียนผังงาน

สัญลักษณ์	ความหมาย
	การเริ่มต้นและการสิ้นสุดการทำงานของโปรแกรม
	ลูกศรแสดงทิศทางการทำงานของโปรแกรม
	การประมวลผล การคำนวณ (Process)
	การทำงานย่อย (Subprocess)

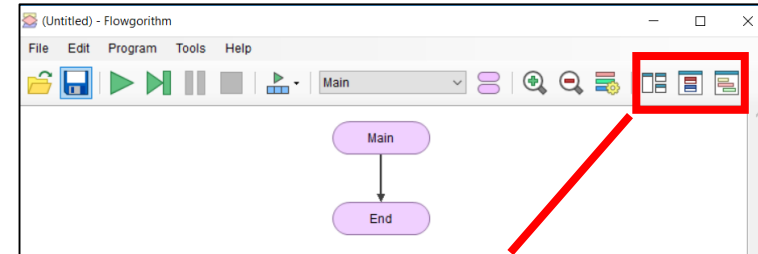
สัญลักษณ์ที่ใช้ในการเขียนผังงาน

สัญลักษณ์	ความหมาย
	การรับหรือแสดงผลข้อมูล (Input/Output) โดยไม่จำกัดอุปกรณ์
	การตรวจสอบเงื่อนไขเพื่อเลือกทำอย่างใดอย่างหนึ่ง มีลูกศรพุ่งออกมากกว่า 1 ลูกศรเสมอ
	แสดง Output เป็นรูปแบบเอกสาร
	จุดเชื่อมต่อผังงาน (Connector) หรือใช้เป็นจุดเชื่อมต่อเริ่มต้นในหน้าใหม่ (On-page Connector)
	จุดเชื่อมต่อเพื่อขึ้นหน้าถัดไป (Off-page Connector)

เครื่องมือที่ใช้เขียน Flowchart



- Flowgorithm เป็นโปรแกรมเครื่องมือที่ใช้เขียน Flowchart แล้วจะสร้างชุดคำสั่งภาษา C ให้อัตโนมัติ
- <http://www.flowgorithm.org/download/>



เปิดและจัดหน้าต่าง



Windows

(Untitled) * - Flowgorithm

File Edit Program Tools Help

Main

```
graph TD
    Main([Main]) --> Declare[Integer sum, n, i]
    Declare --> Read[Read average, score]
    Read --> Sum0[sum = 0]
    Sum0 --> InputN[Input n]
    InputN --> CondN{ n > 0 }
    CondN -- False --> End([End])
    CondN -- True --> LoopStart{i = 1 to n}
    LoopStart --> InputScore[Input Score]
    InputScore --> CondScore{ score > 0 }
    CondScore -- True --> SumAdd[sum = sum + score]
    SumAdd --> LoopStart
    CondScore -- False --> LoopStart
    LoopStart -- Done --> AvgCalc[average = sum / n]
    AvgCalc --> OutputAvg[Output average]
    OutputAvg --> End
```

Flow window : หน้าต่าง
ที่ให้ผู้ใช้งานสร้าง
Flowchart

Source Code Viewer

Auto Pseudocode

```
0 Function Main
1   Declare Integer sum, n, i
2   Declare Real average, score
3
4   Assign sum = 0
5   Input n
6   If n > 0
7     For i = 1 to n
8       Input Score
9       If score > 0
10        Assign sum = sum + score
11      End
12 End
```

Source code viewer :
หน้าต่างแสดงรายการ
Pseudocode หรือคำสั่งภาษา
ต่าง ๆ ที่สร้างขึ้น

Console

172

242

80.66666666666667

92

70

Console window : หน้าต่าง
แสดงผลลัพธ์ของการรันคำสั่ง

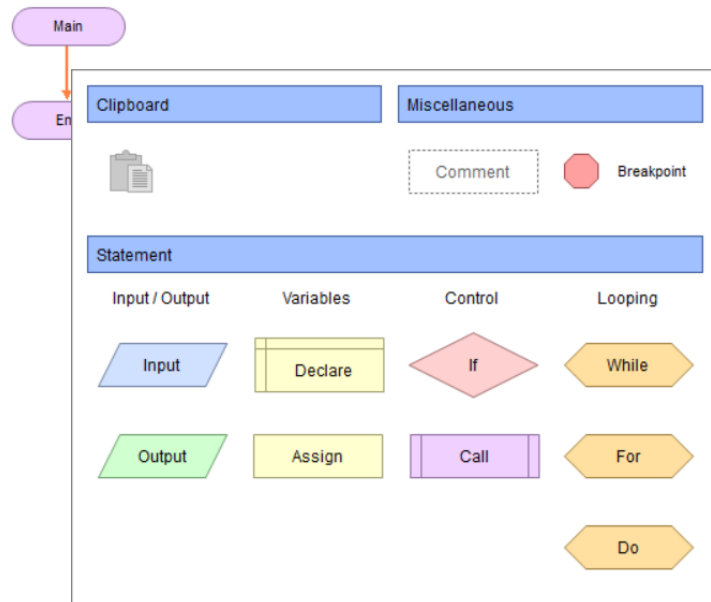
Variable ...

Main

sum	155
n	3
i	3
average	Uninitialized
score	75

Variable watch window : หน้าต่างแสดงรายการตัว
แปรในขณะรันโปรแกรม

Add Statement



- Right Click หรือ Double Click ที่เส้นเชื่อมระหว่าง Statement ประเภทของ Statement
1. Declare เป็นการประกาศตัวแปร
 2. Assign เป็นการกำหนดค่าตัวแปร
 3. Input เป็นการรับค่าตัวแปร
 4. Output เป็นการแสดงผลออกทางหน้าต่าง Console
 5. If เป็นการสร้างเงื่อนไข
 6. Call เป็นการเรียกฟังก์ชัน
 7. For เป็นการวนลูปตามจำนวนที่กำหนดไว้
 8. While เป็นการวนซ้ำตามเงื่อนไขตามที่กำหนดไว้ (ตรวจสอบเงื่อนไขก่อนวนซ้ำ)
 9. Do เป็นการวนซ้ำตามเงื่อนไขตามที่กำหนดไว้ (วนซ้ำคำสั่งก่อนตรวจสอบเงื่อนไข)

INPUT / OUTPUT

- OUTPUT

- แสดงผลออกหน้าจอ

- PROCESS

- การกำหนดค่า การคำนวณ เงื่อนไข และการวนลูป

- INPUT

- รับค่าใส่ตัวแปร

VARIABLES

● DECLARE

- ประกาศ “ตัวแปร”
- เก็บข้อมูลตอนทำงาน
- สามารถประกาศ หลายตัวแปรได้โดยใช้ comma
- กำหนดขอบเขตString โดยใช้ Double quote " "
- ใช้เครื่องหมาย & ในการเชื่อมต่อ
- case insensitive อักษรเล็กใหญ่คือตัวเดียวกัน

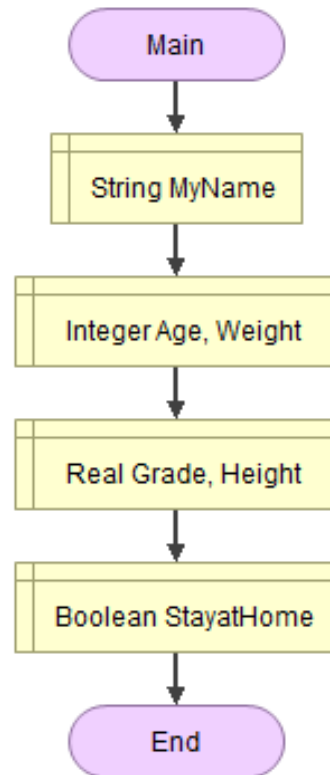
● ASSIGN

- กำหนดค่าให้แก่ “ตัวแปร”
- เก็บผลลัพธ์จากการคำนวณ

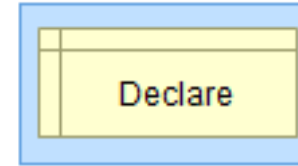
ตัวแปร

- รับค่าเพื่อไปทำงาน เปลี่ยนแปลงค่าไปตามการกำหนด
- ต้องกำหนด “ชนิดข้อมูล” ของตัวแปร (Data Type)
 - Integer เลขจำนวนเต็ม
 - เก็บ “ตัวเลข” ได้ทั้งค่าบวก,ลบ ศูนย์ ไม่สามารถเลขทศนิยม 0 , -10 , 1234
 - Real เลขจำนวนจริง
 - เก็บ “ตัวเลข” อะไรก็ได้ 0, 3.14, -10, -0.1132
 - Boolean ค่าความจริง
 - เก็บค่าความจริงค่าใดค่าหนึ่งใน 2 ค่า คือ True, False
 - String ตัวอักษร
 - เก็บข้อมูล ประโยค คำ ตัวอักษร ช่องว่าง ระบุขอบเขตด้วย double quote “A”, “สวัสดี ชาวโลก”, “Happy”

DECLARE VARIABLES



Variables



Declare Properties

Declare A Declare Statement is used to create variables and arrays. These are used to store data while the program runs.

Variable Names:
MyName

Type:
String ☐ Array?

OK Cancel

String

Declare Properties

Declare A Declare Statement is used to create variables and arrays. These are used to store data while the program runs.

Variable Names:
Age, Weight

Type:
Integer ☐ Array?

OK Cancel

Integer

Declare Properties

Declare A Declare Statement is used to create variables and arrays. These are used to store data while the program runs.

Variable Names:
StayatHome

Type:
Boolean ☐ Array?

OK Cancel

Boolean

Declare Properties

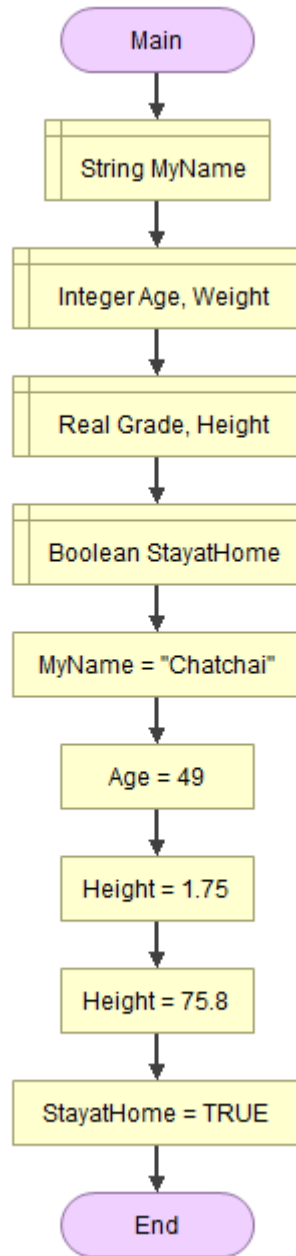
Declare A Declare Statement is used to create variables and arrays. These are used to store data while the program runs.

Variable Names:
Grade, Height

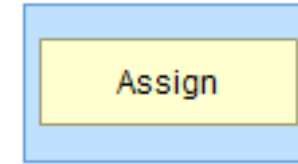
Type:
Real ☐ Array?

OK Cancel

Real



ASSIGN VARIABLES



Assignment Properties

Assign An Assignment Statement calculates an expression and then stores the result in a variable.

Variable: MyName = Expression: "Chatchai"

ชื่อตัวแปร ค่าตัวแปรอยู่ในเครื่องหมาย ""

OK Cancel

String

Assignment Properties

Assign An Assignment Statement calculates an expression and then stores the result in a variable.

Variable: Age = Expression: 49

ชื่อตัวแปร ค่าตัวแปรตัวเลขจำนวนเต็ม

OK Cancel

Integer

Assignment Properties

Assign An Assignment Statement calculates an expression and then stores the result in a variable.

Variable: StayatHome = Expression: TRUE

OK Cancel

Boolean

Assignment Properties

Assign An Assignment Statement calculates an expression and then stores the result in a variable.

Variable: Height = Expression: 1.75

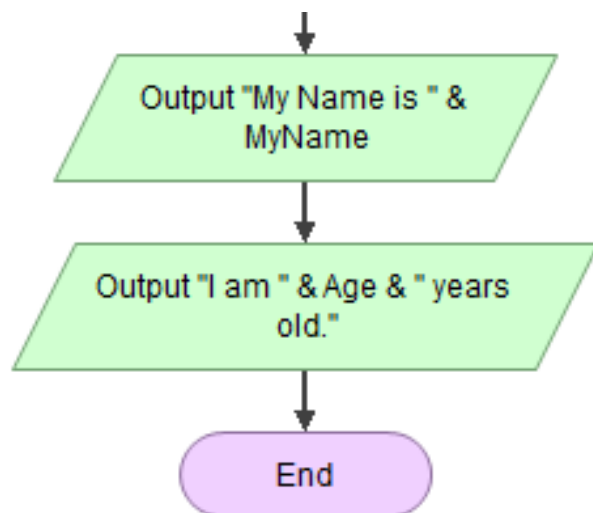
OK Cancel

Real

DISPLAY VARIABLES

- การแสดงผลข้อความจะใช้เครื่องหมาย & (Ampersand) เป็นตัวเชื่อมระหว่างข้อความกับตัวแปร หรือตัวแปรกับตัวแปร

- "My Name " & MyName
- "I am " & Age & " years old."

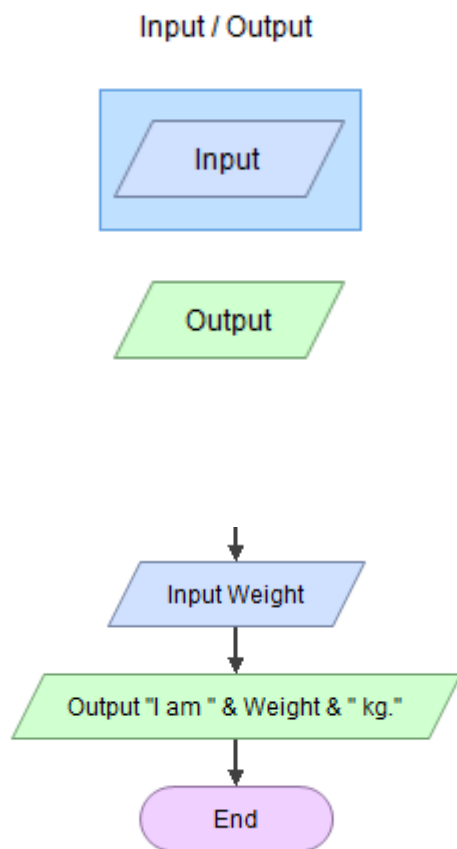


The screenshot shows the "Output Properties" dialog box. It has a title bar with a close button. Below the title bar is a blue header with a green "Output" tab and the text "An Output Statement evaluates an expression and then displays the result to the screen." Below this is a text area with the prompt "Enter an expression below:" and the text "My Name is " & MyName". There is a "New Line" checkbox which is checked. At the bottom are "OK" and "Cancel" buttons.

The screenshot shows the "Output Properties" dialog box. It has a title bar with a close button. Below the title bar is a blue header with a green "Output" tab and the text "An Output Statement evaluates an expression and then displays the result to the screen." Below this is a text area with the prompt "Enter an expression below:" and the text "I am " & Age & " years old."". There is a "New Line" checkbox which is checked. At the bottom are "OK" and "Cancel" buttons.

INPUT VARIABLE

- การรับค่าใส่ในตัวแปรจากคีย์บอร์ด โดยระบุชื่อตัวแปรที่ต้องการกำหนดค่า



Input Properties

Input An Input Statement reads a value from the keyboard and stores the result in a variable.

Enter a variable name below:

Weight

OK Cancel

Output Properties

Output An Output Statement evaluates an expression and then displays the result to the screen.

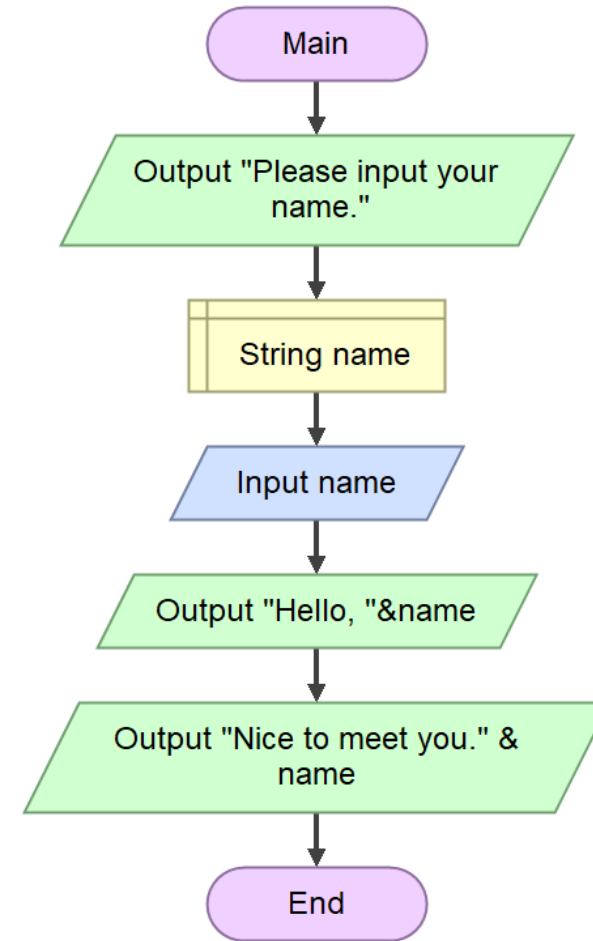
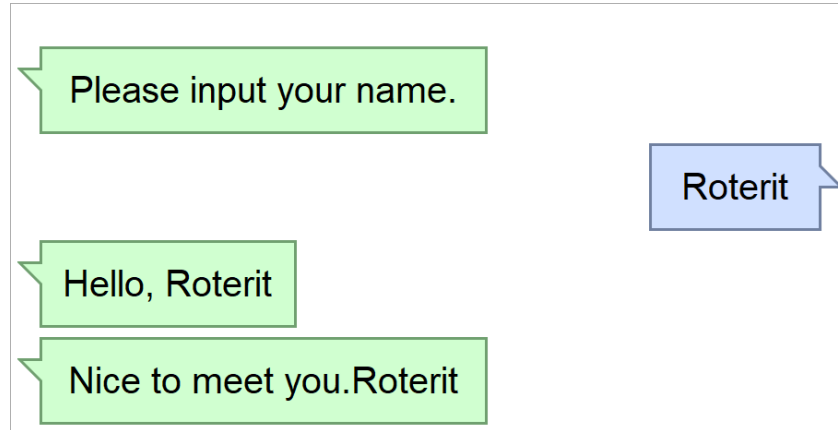
Enter an expression below:

"I am " & Weight & " kg."

☒ New Line

OK Cancel

OUTPUT -> INPUT->OUTPUT



นิพจน์ (EXPRESSIONS)

- นิพจน์เป็นการรวมระหว่างตัวแปร (Variable) 2 ตัว หรือที่เรียกว่า (Operands) และตัวดำเนินการ (Operators)
- นิพจน์จะให้ผลลัพธ์เป็นค่าที่มีประเภทข้อมูล (Data Type) ประเภทใดประเภทหนึ่งได้แก่ Integer, Real, Boolean
- การเขียนนิพจน์ จะมี Operator คั่นกลางระหว่างตัวแปร 2 ตัว
 <Variable 1> <operator> <Variable 2>
- ตัวอย่างเช่น

Expression	Variable 1	Operator	Variable 2	=
3 * 5	3	*	5	
9 / 2	9	/	2	
"Hello" + " World"	"Hello"	+	" World"	
SEX == "MALE"	SEX	==	"MALE"	
AGE > 25	AGE	>	25	

ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic operators)

กำหนดให้ $x = 15$ และ $y = 2$

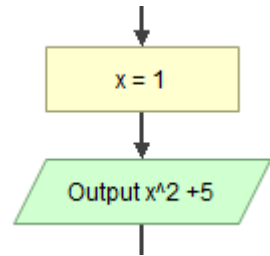
Operator	Name	Example	Answer
+	Addition บวก	$5 + 4$	9
-	Subtraction ลบ	$5 - 4$	1
*	Multiplication คูณ	$5 * 4$	20
/	Division หาร	$5 / 4$	1.25
%	Modulus หารเอาเศษ	$5 \% 4$	1
^	Exponentiation ยกกำลัง	$5 \wedge 4$	625

Operator	Answer
$x + y$	
$x - y$	
$x * y$	
x / y	
$x \% y$	
$x \wedge y$	

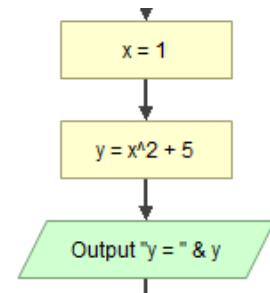
การกำหนดค่าตัวแปรจากการคำนวณ

- เมื่อมีการสร้างนิพจน์คำนวณทางคณิตศาสตร์ จะมีการนำผลลัพธ์มาใช้งานใน 2 รูปแบบคือ

1. นำผลลัพธ์มาแสดงในทันที (ใช้ค่าผลลัพธ์ครั้งเดียวทิ้งแล้วไม่สามารถใช้ค่าได้อีก)



2. นำผลลัพธ์ไปเก็บในตัวแปร (นำค่าผลลัพธ์กลับมาใช้ได้หลายครั้ง)



ลำดับการคำนวณ

- ทำงานในวงเล็บก่อน (parentheses)
- หลังจากการทำงานในวงเล็บ จะทำงานตามลำดับต่อไปนี้

1. \wedge

2. $*$, $/$, $\%$,

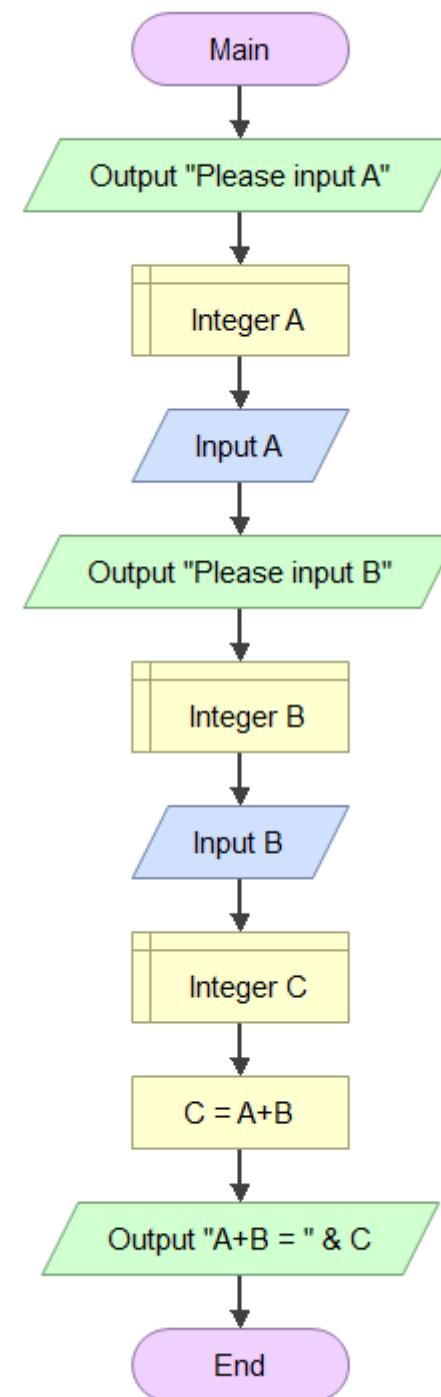
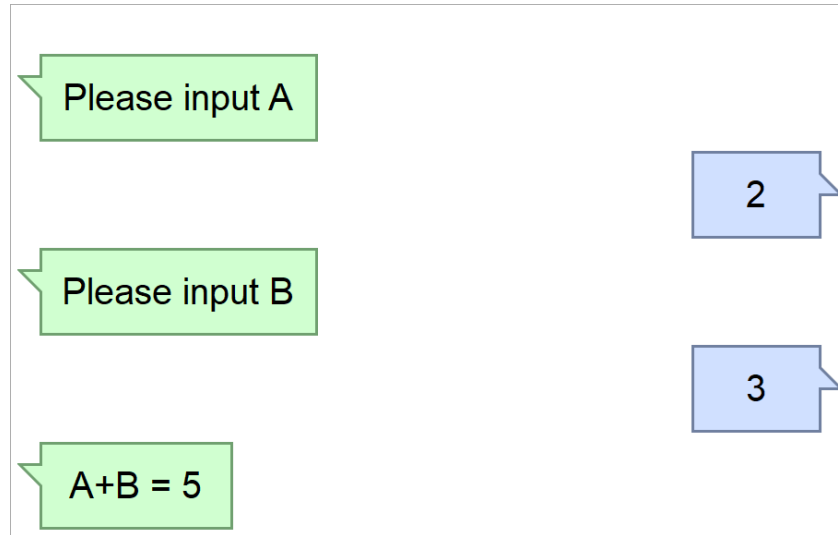
3. $+$ และ $-$ จากคำสั่งซ้ายไปขวา

- ตัวอย่างเช่น

$$(8+2)**2-4*2/2 = 96$$

$$8+2**2-4*2/2 = 8 + 4 - 4 = 8$$

Programming.



LAB Exercise

แบบฝึกหัด

- สร้างตัวแปร width และ height ชนิดเลขจำนวนเต็ม
- นำเข้าค่าตัวแปร width และ height
- คำนวณค่าตัวแปร area

$$area = width \times height$$

- แสดงค่าตัวแปร area

- สร้างตัวแปร X และ Y ชนิดเลขจำนวนเต็ม
- นำเข้าค่าตัวแปร X และ Y
- คำนวณค่าตัวแปร Z

$$Z = \sqrt{x^2 + y^2}$$

- แสดงค่าตัวแปร Z

แบบฝึกหัด

- สร้างตัวแปรรัศมีวงกลม r ชนิดเลขจำนวนจริง
- นำเข้าค่าตัวแปร r
- คำนวณพื้นที่ $area$ และความยาวเส้นรอบวง $circumference$

$$area = \left(\frac{22}{7}\right) * r^2$$

$$circumference = 2 * \left(\frac{22}{7}\right) * r$$

- แสดงค่าตัวแปร $area$ และ $circumference$

- สร้างตัวแปร $weight$ เป็นเลขจำนวนเต็ม kg (กิโลกรัม)
- สร้างตัวแปร $height$ เป็นเลขจำนวนทศนิยม $meter$ (เมตร)
- คำนวณค่า BMI

$$BMI = weight / (height^2)$$

- แสดงค่าตัวแปร BMI

แบบฝึกหัด

- สร้างตัวแปรจำนวนเงิน money และราคาน้ำมัน บาทต่อลิตร price ชนิดเลขจำนวนเต็ม
- นำเข้าค่าตัวแปร money และ price
- คำนวณจำนวนลิตรของน้ำมันที่ซื้อได้ liter และเงินทอนที่เหลือ change

$$liter = Int \left(\frac{money}{price} \right)$$

$$change = money \% price$$

- แสดงจำนวนลิตรที่ซื้อ และ เงินทอน

- สร้างตัวแปร Animal ชนิดข้อความ
- นำเข้าค่าตัวแปร Animal เป็นคำศัพท์ที่เกี่ยวข้อง สัตว์ เช่น Dog, Cat, Elephant เป็นต้น
- คำนวณค่าตัวแปร length เท่ากับจำนวนตัวอักษรของชื่อสัตว์ในตัวแปร Animal เช่น

Elephant => length = 8

Cat => length = 3

- แสดงค่าตัวแปร length
- แสดงตัวอักษรตัวที่สองของชื่อสัตว์ในตัวแปร Animal

$length = Len(Animal)$
 $second = Char(Animal, 1)$