

Basic program design

ฉัตรชัย เกษมทวีโชค

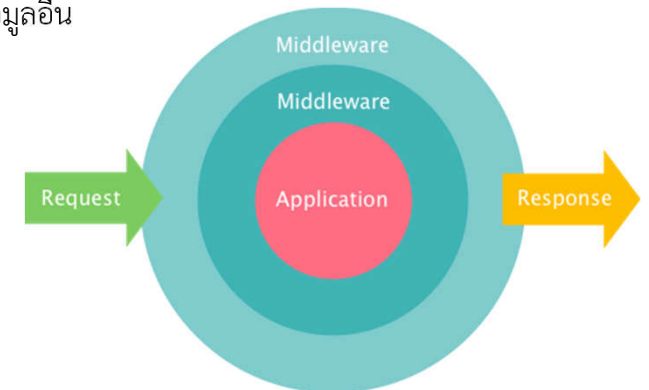
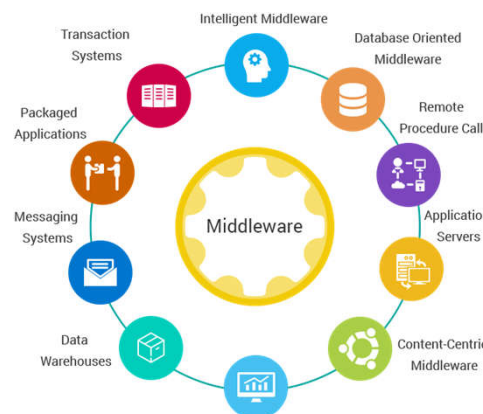
Chatchai.kase@ku.th

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) เป็นโปรแกรมที่ออกแบบมาเพื่อใช้งานฮาร์ดแวร์ของคอมพิวเตอร์และจัดเตรียมและสร้างระบบพื้นฐานสำหรับใช้งานโปรแกรมประยุกต์อาทิเช่น Windows, MacOS, Linux, Unix เป็นต้น
 - โปรแกรมตัวกลาง (Middleware software) คือโปรแกรมคอยช่วยเหลือดูแลโปรแกรมประยุกต์ที่รันอยู่บน OS หรือจะเรียกว่าตัวเชื่อมระหว่าง application และ operating system ก็ได้ ซึ่งมันช่วยให้ผู้พัฒนาโปรแกรมสามารถเชื่อมต่อสื่อสารกับภายนอกได้ง่ายขึ้น ทำให้ลดภาระในการดูแลรายละเอียดรอบข้างและเน้นแต่งานหลักที่ต้องการ
 - ประกอบชุดของบริการที่ช่วยให้สามารถทำงานได้หลายกระบวนการ บนเครื่องหนึ่งเครื่องหรือมากกว่าเพื่อประสานงาน สนับสนุนและทำให้การกระจายงานง่ายขึ้น นอกจากนี้ยังสามารถให้ข้อมูลมีอยู่ในฐานข้อมูลเดียวที่สามารถเข้าถึงได้ผ่านทางฐานข้อมูลอื่น

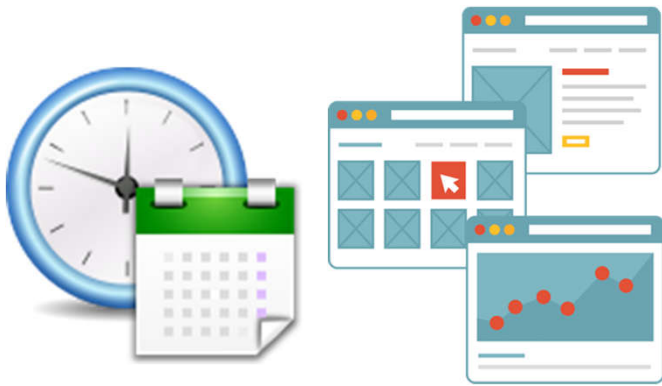


Source: <http://www.girnarsez.com/middleware>

<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

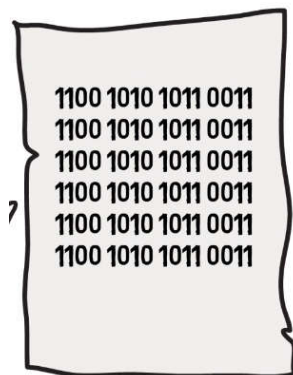
- ผู้คนที่ผ่านการฝึกฝนขั้นสูงที่เรียกว่า คอมพิวเตอร์โปรแกรมเมอร์ (Computer Programmer) จะเป็นผู้ที่สร้างรายการคำสั่ง (instruction) ในโปรแกรม
- รายการคำสั่งดังกล่าวเราเรียกว่า “code”
- และกระบวนการในการเขียนคำสั่ง เราเรียกว่า “coding”
- <https://www.youtube.com/watch?v=YF1IjAeTbq0>

คำสั่งคอมพิวเตอร์ (Code)

- **Code** : เป็นคำศัพท์ที่แทนความหมายของชุดบรรทัดคำสั่งที่ถูกเขียนด้วยภาษาคอมพิวเตอร์ใดก็ได้ เช่น ภาษาเครื่อง ภาษาระดับสูง
- เครื่องคอมพิวเตอร์คิดและพูดในระบบเลขฐานสอง 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 >
```

© TechTerms.com

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

ตัวแปลภาษาระดับสูง

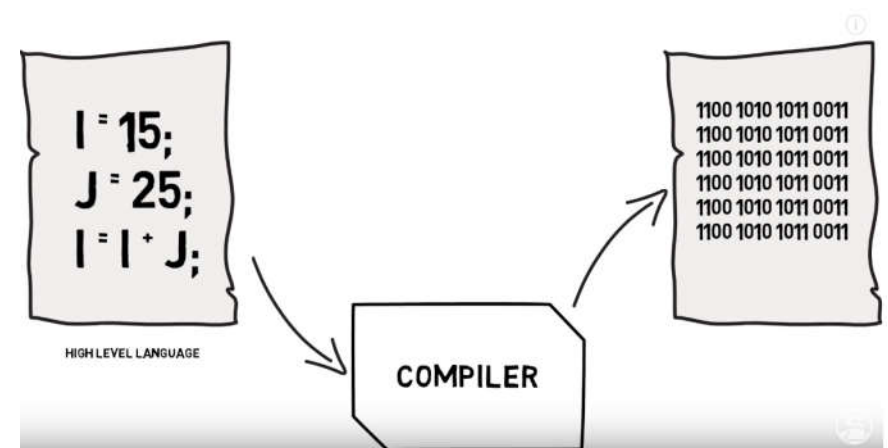
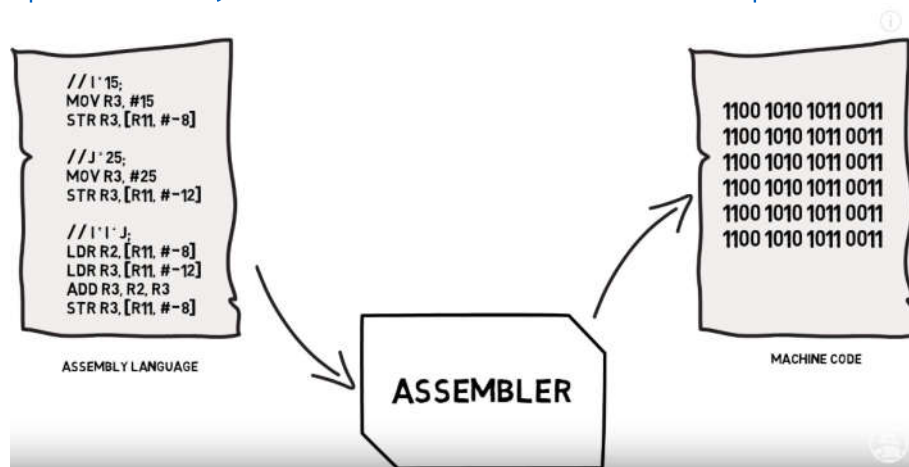
- ตัวแปลภาษาระดับสูง : เป็นตัวแปลภาษาระดับสูงเป็นภาษาเครื่อง เราสามารถแบ่งตัวแปลภาษาประเภทนี้ออกเป็น 2 ประเภทคือ
 1. **Compilers** เป็นการแปลคำสั่งโปรแกรม **source code** ทั้งหมดไปเป็นภาษาเครื่อง เพื่อเป็นไฟล์ที่ประมวลผลได้ (Executable file)
 - ในบางภาษาจะต้องมีการแปลเป็นไฟล์ **Object Code** ก่อนแล้วจึงลิงก์กับไฟล์ภายนอกที่เกี่ยวข้องเพื่อสร้างเป็นไฟล์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

Interpre = ล่าม

Assembler & Compiler

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

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



Assembler เป็นตัวแปลภาษา Assembly ไปเป็นไฟล์ภาษาเครื่องที่ใช้ในการรันการทำงาน

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

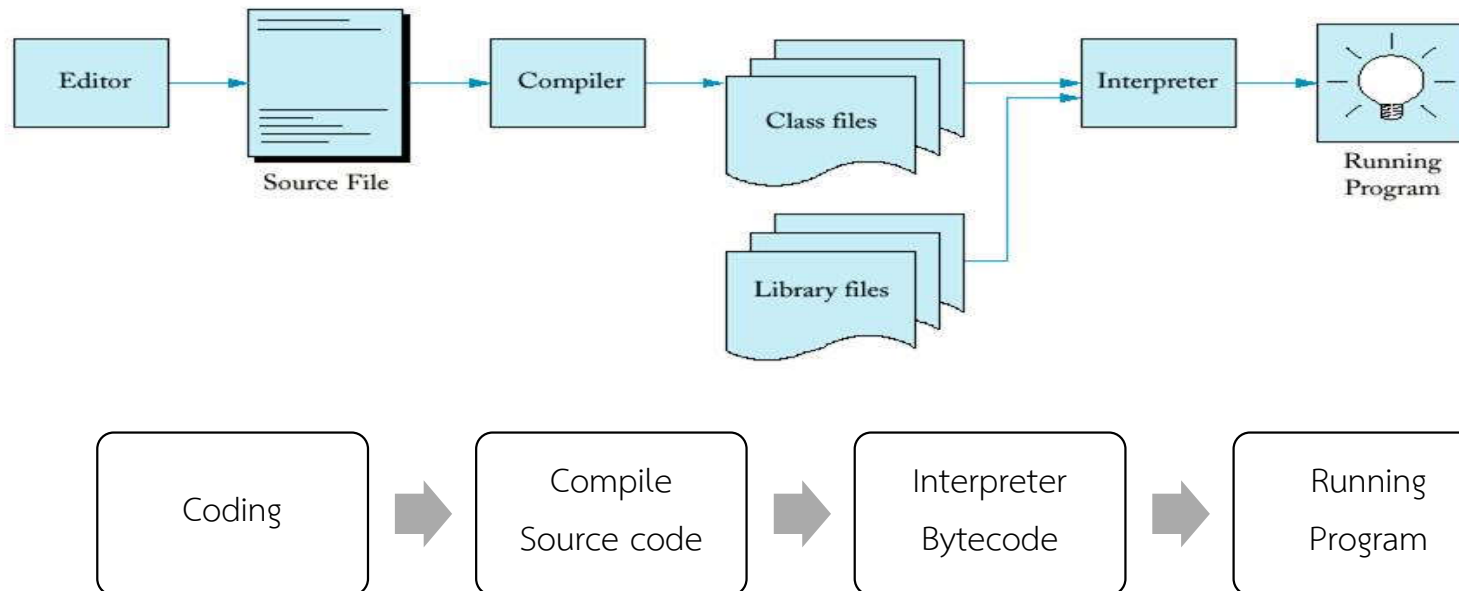
Compiler เป็นตัวแปลภาษาชั้นสูงไปเป็นไฟล์ภาษาเครื่องที่สามารถรันการทำงานได้ก็ครั้งก็ได้

Interpreter เป็นตัวแปลภาษาชั้นสูงทีละบรรทัดคำสั่งไปสั่งงานคอมพิวเตอร์ทันที

Bytecode รันได้ทุกแพลตฟอร์ม
Java JVM = Java Virtual Machine
ตัวแปลคำสั่ง Bytecode ให้ทำงานในเครื่องคอมพิวเตอร์

Hybrid method (Java program)

bytecode -> JVM (Windows)
bytecode -> JVM (MacOS)
bytecode -> JVM (Unix)



Source: <http://www.cs.ust.hk/emb/itepc2006/itepc01a/t01/>

Program Design

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

Pseudocode (รหัสเทียม)

- ซูโดโค้ดเป็นคำอธิบายขั้นตอนการทำงานของโปรแกรม
- ใช้ถ้อยคำผสมระหว่างภาษาอังกฤษ (หรือภาษาอื่นๆ แต่ควรใช้ภาษาอังกฤษ) และภาษาการเขียนโปรแกรมแบบโครงสร้าง
- รูปแบบของซูโดโค้ด ปกติจะเป็นไปตามรูปด้านล่าง

<ol style="list-style-type: none">1. IF lamp works, go to step 7.2. Check if lamp is plugged in.3. IF not plugged in, plug in lamp.4. Check if bulb is burnt out.5. IF blub is burnt, replace bulb.6. IF lamp doesn't work, buy new lamp.7. Quit ... problem is solved.

รูปแบบการเขียน Pseudocode (รหัสเทียม)

- การรับข้อมูล ใช้ READ หรือ INPUT
- การแสดงผล ใช้ PRINT หรือ SHOW
- การคำนวณ ใช้ Compute
- เงื่อนไข ใช้ IF-THEN-ELSE และใช้ ENDIF ปิดท้ายเงื่อนไข
 - หากมีตัวเลือกมากกว่าสองทางใช้ CASE และ ENDCASE
- การทำแบบวนซ้ำ ใช้ FOR – ENDFOR หรือ REPEAT-UNTIL หรือ WHILE-DO
- การกระโดดข้าม ใช้ LABEL และ GOTO

ตัวอย่าง

Pseudocode ของการหาพื้นที่สามเหลี่ยม

Algorithm: Calculate triangle area

BEGIN

READ *base* -> เขียนเป็น READ base,height ก็ได้

READ *height*

Compute $area = (base \times height) / 2$

PRINT *area*

END

- Pseudocode ของการบวกเลข $1+2+3+\dots+100$ และแสดงผลลัพธ์ออกมา

Algorithm: Calculate summation from 1 to 100

BEGIN

1. $i = 1$

2. $sum = 0$

3. **WHILE** ($i \leq 100$) **DO**

4. **Compute** $sum = sum + i$

5. **Compute** $i = i + 1$

6. **ENDWHILE**


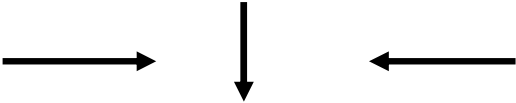


7. **PRINT** *sum*

END

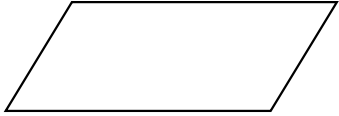
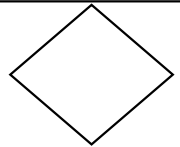

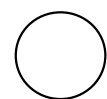
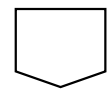
การเขียนผังงาน (Flowchart)

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

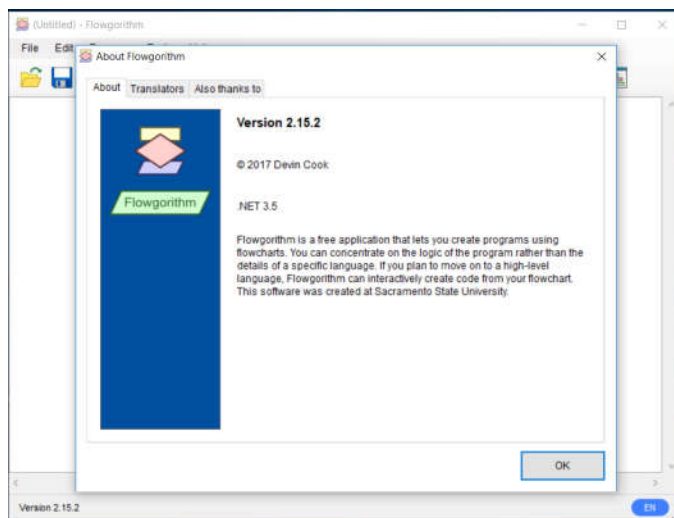
สัญลักษณ์ที่ใช้ในการเขียนผังงาน

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

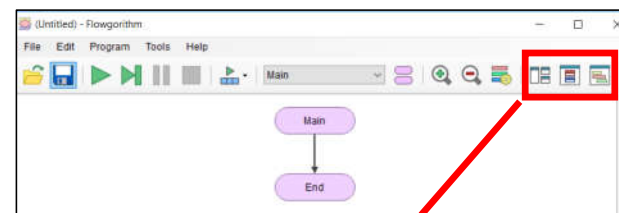
สัญลักษณ์ที่ใช้ในการเขียนผังงาน

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

The image displays a programming IDE with four distinct windows, each highlighted with a red border and a descriptive text box:

- Flow window:** Displays a flowchart for calculating the average of 'n' scores. The process starts with 'Main', followed by 'Integer sum, n, i', 'Real average, score', 'sum = 0', and 'Read n'. A loop 'For i = 1 to n' contains 'Input Score' and 'If score > 0' leading to 'sum = sum + score'. After the loop, 'average = sum / n' is calculated, followed by 'Output average' and 'End'.
- Source Code Viewer:** Shows the corresponding pseudocode for the flowchart. It includes declarations for 'sum, n, i' (Integer) and 'average, score' (Real), followed by the same logic as the flowchart.
- Console window:** Displays the output of the program. It shows three input values (172, 242, 80) and two output values (92, 70).
- Variable watch window:** Lists the variables 'sum', 'n', 'i', 'average', and 'score' with their current values: 155, 3, 3, Uninitialized, and 75 respectively.

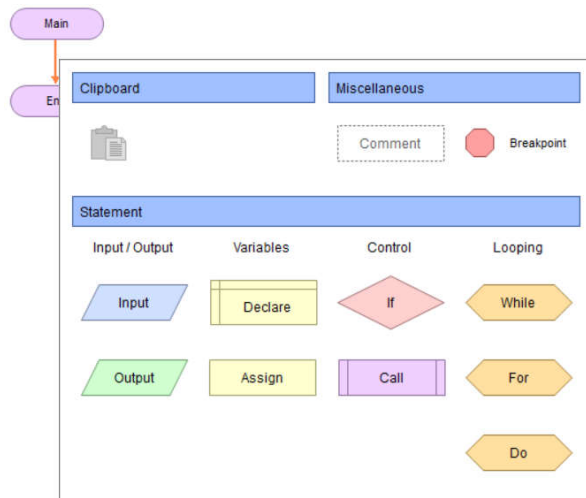
Flow window : หน้าต่าง
ที่ให้ผู้ใช้งานสร้าง
Flowchart

Source code viewer :
หน้าต่างแสดงรายการ
Pseudocode หรือคำสั่งภาษา
ต่าง ๆ ที่สร้างขึ้น

Console window : หน้าต่าง
แสดงผลลัพธ์ของการรันคำสั่ง

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 (วนซ้ำ

WHAT IS A Algorithm

1. ลำดับการทำงานที่ตรงไปตรงมาไม่ซับซ้อน
2. มีควบคุมการไหลก่อนหลังของคำสั่งที่ต้องถูกประมวลผล
3. เมื่อการระบุจะหยุดการทำงานเมื่อไร

$1+2+3 = a$ algorithm

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

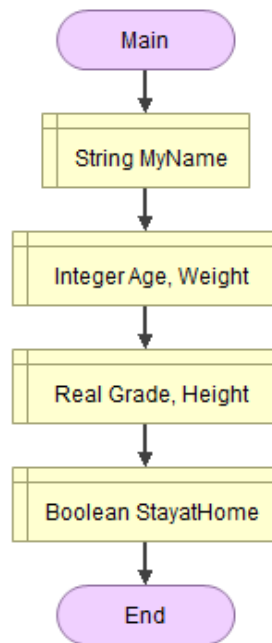
● ASSIGN

- กำหนดค่าให้แก่ “ตัวแปร”
- เก็บผลลัพธ์จากการคำนวณ

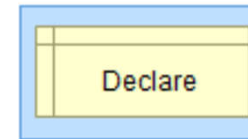
ตัวแปร

- 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

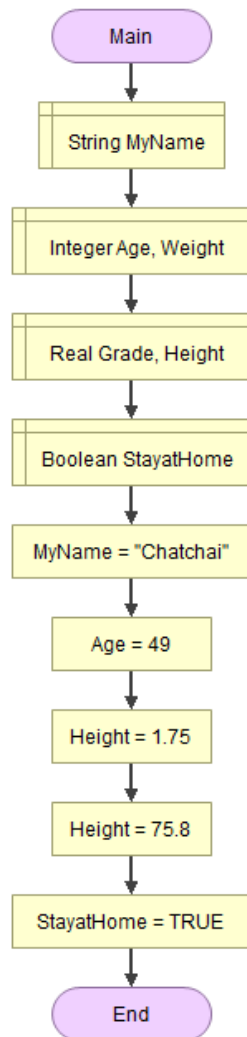
Declare A Declare Statement is used to create variables and arrays. These are used to store data while the program runs.

Variable Names: Grade, Height

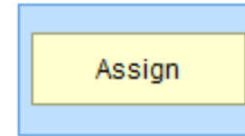
Type: Real ☐ Array?

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 ค่าตัวแปรตัวเลข จำนวนเต็ม

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

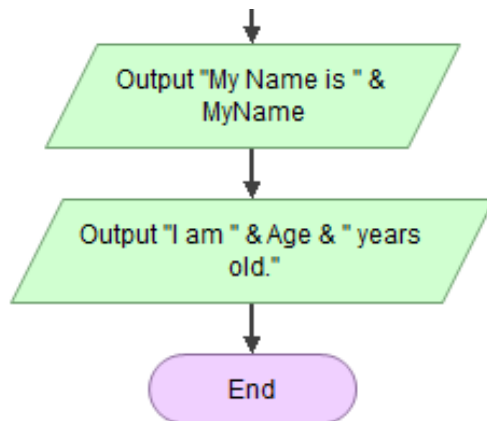
OK Cancel

Real

DISPLAY VARIABLES

- การแสดงผลข้อความจะใช้เครื่องหมาย & (Ampersand) เป็นตัวเชื่อมระหว่างข้อความกับตัวแปร หรือตัวแปรกับตัวแปร

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

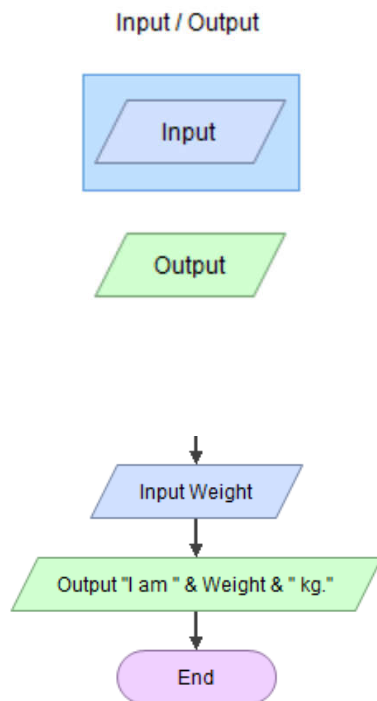


The screenshot shows the "Output Properties" dialog box. It has a title bar with a close button. Below the title bar is a blue header with the word "Output" and a description: "An Output Statement evaluates an expression and then displays the result to the screen." Below this is a text area labeled "Enter an expression below:" containing 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 the word "Output" and a description: "An Output Statement evaluates an expression and then displays the result to the screen." Below this is a text area labeled "Enter an expression below:" containing 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.

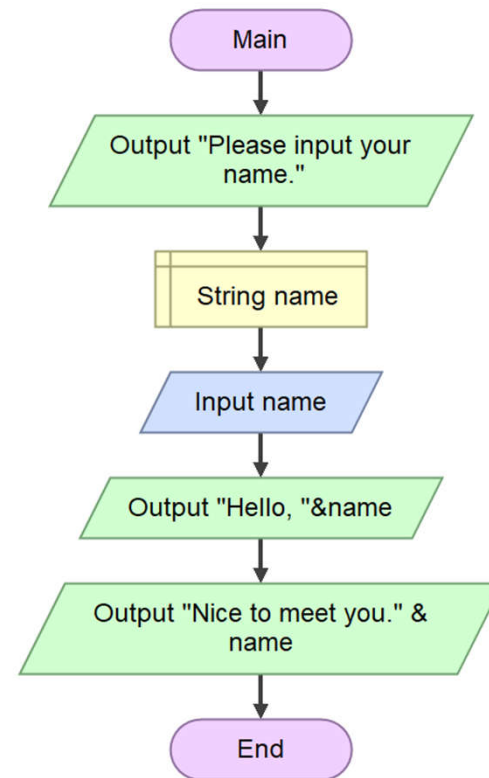
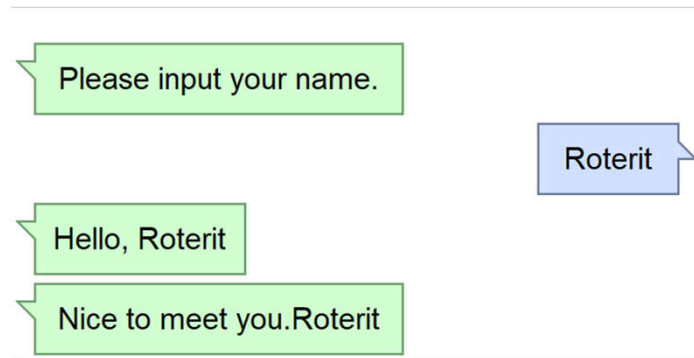
Enter an expression below:

"I am " & Weight & " kg."

☒ New Line

OK Cancel

OUTPUT -> INPUT->OUTPUT



Programming.

