PRINCIPLES OF COMPUTER PROGRAMMING

หลักการเขียนโปรแกรมคอมพิวเตอร์

ระบบคอมพิวเตอร์

- โดยทั่วไปคอมพิวเตอร์หมายถึง เครื่องมือหรือเครื่องจักรที่ประกอบขึ้นจากอุปกรณ์อิเล็กทรอนิกส์ที่มีคุณสมบัติในการรับข้อมูลเข้า จดจำหรือจัดเก็บข้อมูลได้ และสามารถนำข้อมูลมาคำนวณหรือประมวลผลเพื่อให้ได้ผลลัพธ์ที่ต้องการออกมาได้

องค์ประกอบของระบบคอมพิวเตอร์

- ฮาร์ดแวร์ (Hardware)
- ซอฟต์แวร์ (Software)
- พีเพิลแวร์ (People ware)

ฮาร์ดแวร์ (Hardware)

หน่วยรับข้อมูล
Input Unit

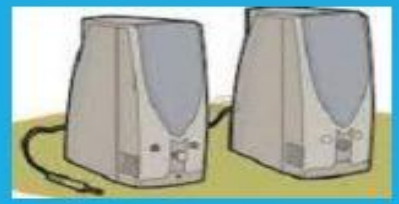
หน่วยประมวลผลกลาง
Central Processing Unit

Control Unit

Arithmetic & Logic Unit

หน่วยความจำ
Memory Unit

หน่วยแสดงผล
Output Unit



ซอฟต์แวร์ (Software)

- **ซอฟต์แวร์ระบบ (System Software)**
  - โปรแกรมที่ทำหน้าที่บริการงานขั้นพื้นฐาน งานที่ต้องติดต่อควบคุมฮาร์ดแวร์ และการจัดระบบระเบียบในการทำงานโดยรวมของคอมพิวเตอร์
- **ซอฟต์แวร์ประยุกต์ (Application Software)**
  - โปรแกรมสั่งงานต่าง ๆ ที่มีผู้เขียนขึ้นมาเพื่อใช้ในการทำงานบางอย่าง เราสามารถหาโปรแกรมประยุกต์มาใช้งานได้ใน 3 ลักษณะ คือ
    - 1 การเขียนโปรแกรมประยุกต์ใช้งานขึ้นเอง โดยใช้ภาษาโปรแกรมต่าง ๆ
    - 2 การจ้างให้ผู้อื่นเขียนโปรแกรมตามที่เรากำหนดให้ตรงกับความต้องการ
    - 3 การซื้อโปรแกรมสำเร็จรูปที่มีผู้ทำมาวางขายทั่วไป

พีเพิลแวร์ (People ware)

- หมายถึง ผู้คนที่มีส่วนเกี่ยวข้องหรือทำงานกับคอมพิวเตอร์ บุคคลเหล่านี้อาจมีความรู้ความสามารถแต่ต่างกัน ทำงานที่แตกต่างกันแต่ก็มีส่วนเกี่ยวข้องกับคอมพิวเตอร์ เราจะถือว่าเป็นพีเพิลแวร์ทั้งสิ้น เช่น นักเขียนโปรแกรม ผู้ดูแลระบบ พนักงานพิมพ์งาน เป็นต้น

ภาษาโปรแกรม

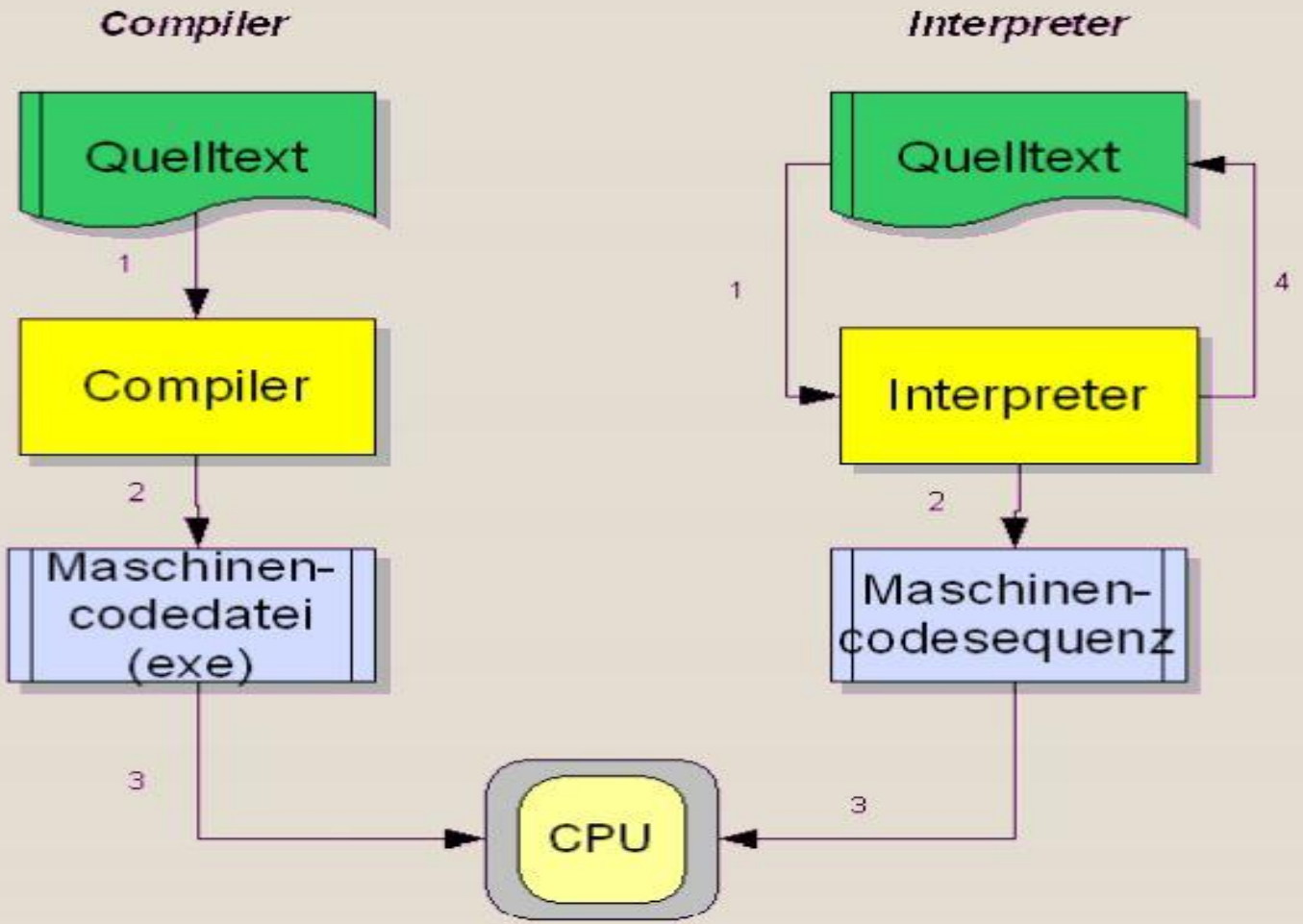
- **ภาษาคอมพิวเตอร์**
  - หมายถึง รูปแบบไวยากรณ์ของคำ หรือรหัสที่ใช้เพื่อการติดต่อสื่อสารสั่งงานกับคอมพิวเตอร์
- **โปรแกรม**
  - หมายถึง คำ กลุ่มคำสั่ง สัญลักษณ์ รหัส หรือตัวเลขที่เรียงร้อยกันตามหลักไวยากรณ์ที่ได้มีการกำหนดไว้ หรืออาจหมายถึง ลำดับของคำสั่งเพื่อควบคุมทำงานของคอมพิวเตอร์

ชนิดของภาษาคอมพิวเตอร์

- ภาษาระดับต่ำ (Low Level Language)
  - ภาษาเครื่อง (Machine Language)
  - ภาษาสัญลักษณ์ (Symbolic Language)
    - ภาษาแอสเซมบลี (Assembly Language)
- ภาษาระดับสูง (High Level Language)

ตัวแปลภาษา

- อินเตอร์พรีเตอร์ (Interpreter)
- คอมไพเลอร์ (Compiler)



ขั้นตอนการพัฒนาโปรแกรม

- กำหนดและวิเคราะห์ปัญหา (Problem Definition and Problem Analysis)
- เขียนผังงานและซูโดโค้ด (Flowchart and Pseudo coding)
- เขียนโปรแกรม (Programming)
- ทดสอบและแก้ไขโปรแกรม (Program Testing and Debugging)
- ทำเอกสารและบำรุงรักษาโปรแกรม (Program Documentation and Maintenance)

การกำหนดและวิเคราะห์ปัญหา
Problem Definition and Problem Analysis

1) กำหนดขอบเขตของปัญหา

โดยกำหนดรายละเอียดให้ชัดเจนว่าจะให้คอมพิวเตอร์ทำอะไร ตัวแปร ค่าคงที่ต้องเป็นไปในลักษณะใด

2) กำหนดลักษณะของข้อมูลเข้าและออกจากระบบ (Input/Output specification)

โดยต้องรู้ว่าข้อมูลที่ส่งไปเป็นอย่างไร มีอะไรบ้าง เพื่อทำให้โปรแกรมประมวลผลและแสดงผลลัพธ์ เช่น การรับค่าจากคีย์บอร์ด การใช้เมาส์ การกำหนดปุ่มต่าง ๆ ลักษณะการแสดงผลหน้าจอว่าจะให้มีรูปร่างอย่างไร โดยขึ้นอยู่กับผู้ใช้งานโปรแกรมเป็นหลัก เช่น ข้อมูลประเภท ตัวเลข ตัวอักษร ข้อความ ใช้ทศนิยมกี่ตำแหน่ง

3) กำหนดวิธีการประมวลผล (process specification)

ต้องรู้ว่าจะให้คอมพิวเตอร์ประมวลผลอย่างไร จึงจะได้ผลลัพธ์ตามที่ต้องการ

เขียนผังงานและซูโดโค้ด
Flowchart and Pseudo coding

หลังจากที่ได้วิเคราะห์ปัญหาแล้ว ขั้นตอนต่อไปเป็นการนำเครื่องมือมาช่วยในการออกแบบโปรแกรม โดยเขียนเป็นลำดับขั้นตอนการทำงานของโปรแกรมที่เรียกว่า อัลกอริทึม (Algorithm) ซึ่งจะแสดงขั้นตอนการแก้ปัญหา

- อาจเขียนในรูปของรหัสจำลอง หรือซูโดโค้ด (Pseudo code)

  เขียนคำอธิบายการทำงานของโปรแกรมเป็นแบบย่อ ไม่มีรูปแบบเฉพาะตัว แต่ละส่วนจะเป็นแนวทางในการเขียนโปรแกรม ซึ่งนำไปใช้กับทุก ๆ โปรแกรมได้ง่ายขึ้น

- หรือเขียนในรูปของผังงาน (Flowchart)

  ผังงานเป็นสัญลักษณ์แทนการทำงานและทิศทางของโปรแกรม

ผังงาน (Flowchart)

❖ แผนผังสัญลักษณ์เพื่อใช้อธิบายลำดับงานที่ต้องกระทำ

Terminator จุดเริ่มต้นหรือจุดสิ้นสุดของการทำงาน

Process การกำหนดค่าหรือการประมวลผล

Data การรับข้อมูลเข้าโดยไม่กำหนดชนิดของสื่อ

Decision การตัดสินใจตามเงื่อนไข

Connector แสดงการเชื่อมโยงลำดับงาน

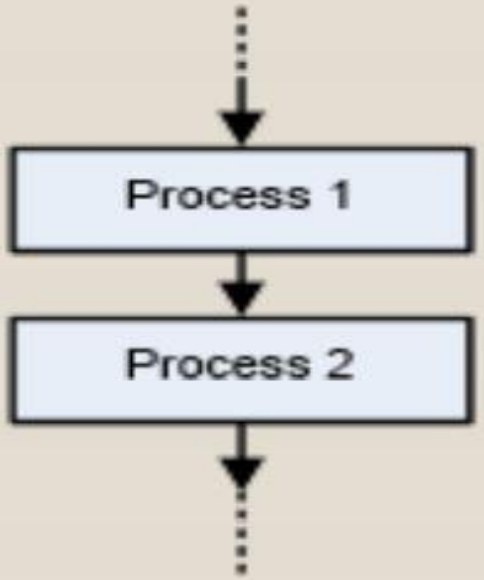
On-page reference การเชื่อมโยงไปโฟลวชาร์ตอื่นภายในหน้ากระดาษเดียวกัน

Off-page reference การเชื่อมโยงไปโฟลวชาร์ตอื่นในหน้ากระดาษอื่น ๆ

Manual input รับข้อมูลเข้าด้วยแป้นพิมพ์

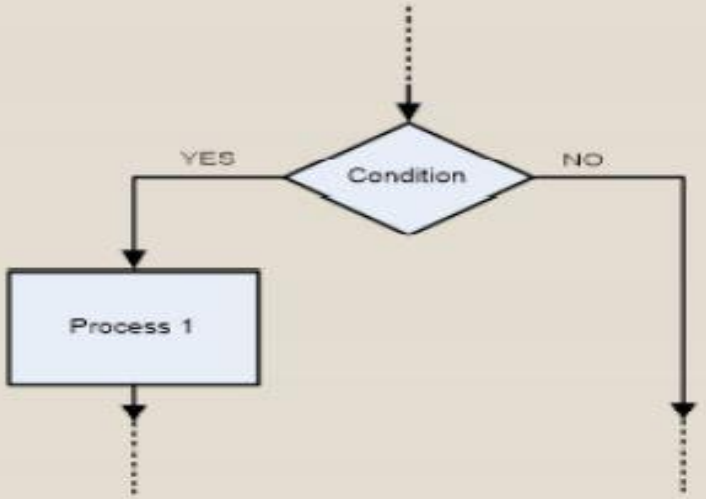
Display การแสดงผลข้อมูลด้วยจอภาพ

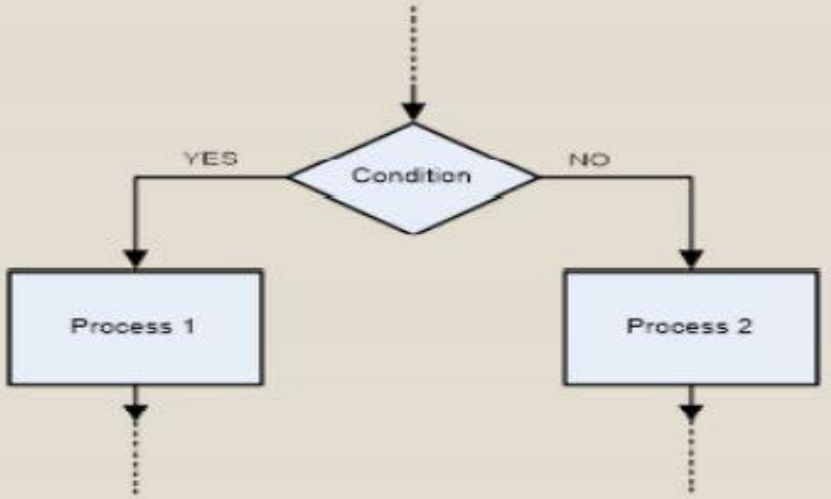
การเขียนผังงานกับโครงสร้างควบคุมแบบลำดับ (Sequence Structure)



เป็นแบบพื้นฐานที่โพรเซสแต่ละอันจะเรียงลำดับกันไป(จากบนลงล่าง) การทำงานต้องทำให้เสร็จทีละโพรเซสไปตามลำดับจะข้ามลำดับหรือย้อนกลับไม่ได้

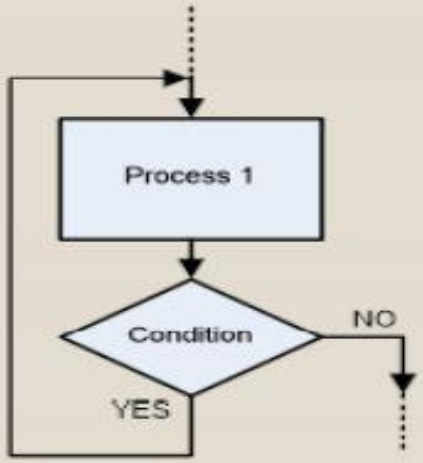
แบบเลือกกระทำตามเงื่อนไข (Condition Branch)

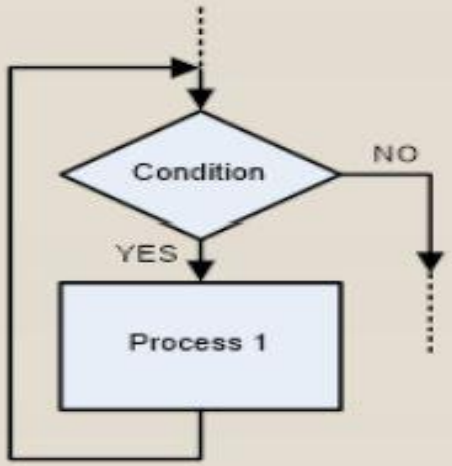




การเขียนโปรแกรมจำเป็นต้องเลือกทำงานอย่างใดอย่างหนึ่งจากเงื่อนไข ดังนั้น จะมีเพียงโพรเซสเดียวเท่านั้นที่ถูกเลือกให้ทำงานจากทางเลือกที่เป็นจริง

แบบกระทำซ้ำ (Looping)





งานบางอย่างต้องการกระทำซ้ำ ๆ หลาย ๆ หน จำนวนรอบของการกระทำซ้ำจะถูกควบคุมด้วยเงื่อนไข มีทั้งแบบตรวจสอบเงื่อนไขก่อนจึงสั่งให้ทำโพรเซส และแบบทำโพรเซสก่อนแล้วจึงตรวจสอบเงื่อนไข

ตัวอย่างการเขียนผังงาน

- ต้องการหาผลรวมของข้อมูลที่อยู่ในตัวแปร A และ B เก็บในตัวแปร C แล้วแสดงผล

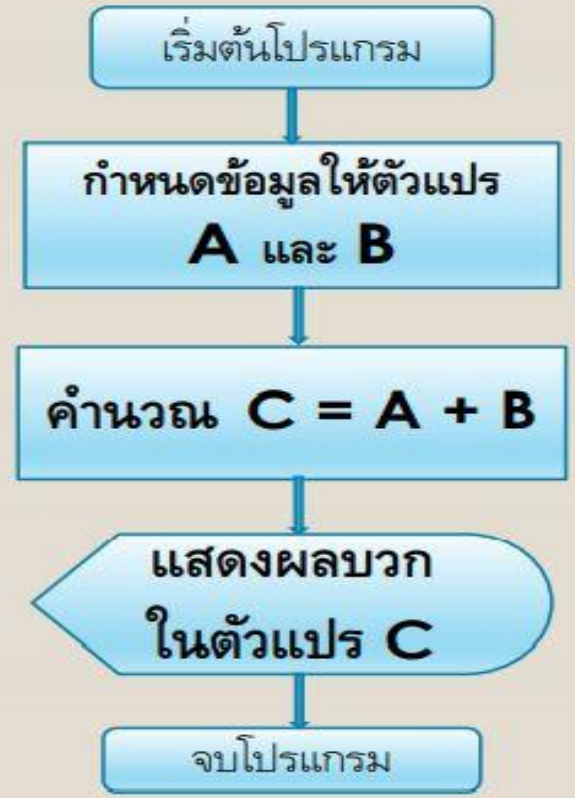
วิเคราะห์ปัญหา (Problem Analysis): 3 ขั้นตอน

1. กำหนดขอบเขตของปัญหา
2. กำหนดลักษณะของข้อมูลเข้าและออกจากระบบ
3. กำหนดวิธีการประมวลผล

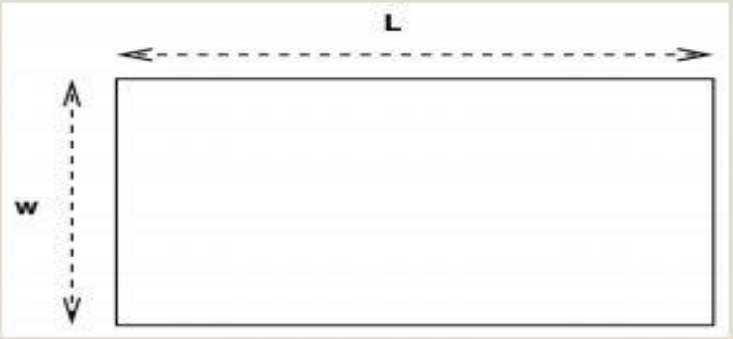
1. Input: กำหนดข้อมูลให้ A, B

2. Process: คำนวณ C = A + B

3. Output: พิมพ์ค่า C



ตัวอย่างการเขียนผังงาน



- ต้องการหาพื้นที่สี่เหลี่ยม จากสูตร A = W * L แล้วแสดงผล

วิเคราะห์ปัญหา (Problem Analysis): 3 ขั้นตอน

1. กำหนดขอบเขตของปัญหา
2. กำหนดลักษณะของข้อมูลเข้าและออกจากระบบ
3. กำหนดวิธีการประมวลผล

1. Input: รับข้อมูลจากแป้นพิมพ์ให้ W, L

2. Process: คำนวณ A = W * L

3. Output: พิมพ์ค่า A

ตัวอย่างการเขียนผังงาน

- ป้อนคะแนนแล้วแสดงผลการตัดเกรดผ่านหรือไม่ผ่าน

วิเคราะห์ปัญหา (Problem Analysis): 3 ขั้นตอน

1. กำหนดขอบเขตของปัญหา
2. กำหนดลักษณะของข้อมูลเข้าและออกจากระบบ
3. กำหนดวิธีการประมวลผล

1. Input: รับข้อมูลจากแป้นพิมพ์ให้ score

2. Process: เปรียบเทียบ score >= 50 หรือไม่

3. Output: แสดงข้อความผ่านหรือไม่ผ่าน