บทที่ 3
ขั้นตอนการเขียนโปรแกรม

จุดประสงค์การเรียนรู้

- ผู้เรียนสามารถอธิบายถึงขั้นตอนการพัฒนาโปรแกรมแต่ละขั้นตอนได้
- ผู้เรียนสามารถอธิบายโปรแกรมแบบโครงสร้างได้

เนื้อหา

- การวิเคราะห์ปัญหา (Analysis the Problem)
- การออกแบบโปรแกรม (Design a Program)
- การเขียนโปรแกรมโดยใช้ภาษาใดภาษาหนึ่ง (Coding)
- การตรวจสอบข้อผิดพลาดของโปรแกรม (Testing and Debugging)
- การทดสอบความถูกต้องของโปรแกรม (Testing and validating)
- การทำเอกสารประกอบโปรแกรม (Documentation)
- การบำรุงรักษาโปรแกรม (Program Maintenance)
- โครงสร้างโปรแกรม

ขั้นตอนการเขียนโปรแกรมมี 7 ขั้นตอนดังนี้

- การวิเคราะห์ปัญหา (Analysis the Problem)
- การออกแบบโปรแกรม (Design a Program)
- การเขียนโปรแกรมโดยใช้ภาษาใดภาษาหนึ่ง (Coding)
- การตรวจสอบข้อผิดพลาดของโปรแกรม (Testing and Debugging)
- การทดสอบความถูกต้องของโปรแกรม (Testing and validating)
- การทำเอกสารประกอบโปรแกรม (Documentation)
- การบำรุงรักษาโปรแกรม (Program Maintenance)

การวิเคราะห์ปัญหา (Analysis the Problem)

- การระบุข้อมูลออก (Output Specification)
  - พิจารณาว่างานที่ทำมีเป้าหมายหรือวัตถุประสงค์อะไร
  - ต้องการผลลัพธ์ที่มีรูปร่าง หน้าตาเป็นอย่างไร
- การระบุข้อมูลเข้า(Input Specification)
  - ต้องรู้ว่าข้อมูลเข้าที่จะส่งเข้ามาในโปรแกรมนั้น มีอะไรบ้าง เป็นข้อมูลที่จะป้อนเข้าสู่คอมพิวเตอร์
- กำหนดวิธีการประมวลผล(Process Specification)
  - ต้องรู้สูตรหรือวิธีการประมวลผลเพื่อให้ผลลัพธ์ตามต้องการ

การวิเคราะห์ปัญหา (Analysis the Problem)

- ตัวอย่าง 1 ต้องการหาค่าคะแนนเฉลี่ยของนักเรียนในชั้น

- ระบุผลลัพธ์ : รายชื่อ คะแนน

  ....

  ....

  ....

  คะแนนเฉลี่ย = ...

การวิเคราะห์ปัญหา (Analysis the Problem)

- ตัวอย่าง 1 ต้องการหาค่าคะแนนเฉลี่ยของนักเรียนในชั้น

- ข้อมูลเข้า : 1. จำนวนนักเรียนทั้งหมด
  2. คะแนนของนักเรียนแต่ละคน

- วิธีการประมวลผล : ใช้สูตรในการคำนวณ

$$\text{คะแนนเฉลี่ย} = \frac{\text{ผลรวมของคะแนนทั้งหมด}}{\text{จำนวนนักเรียนทั้งหมด}}$$

การวิเคราะห์ปัญหา (Analysis the Problem)

- ตัวอย่าง 2 จงเขียนโปรแกรมเพื่อคำนวณหาพื้นที่สามเหลี่ยม

"

- ข้อมูลออก : พื้นที่สามเหลี่ยม
- ข้อมูลเข้า : 1. ฐาน
  2. สูง
- วิธีการประมวลผล : สูตรพื้นที่สามเหลี่ยม = ½ * ฐาน * สูง

การวิเคราะห์ปัญหา (Analysis the Problem)

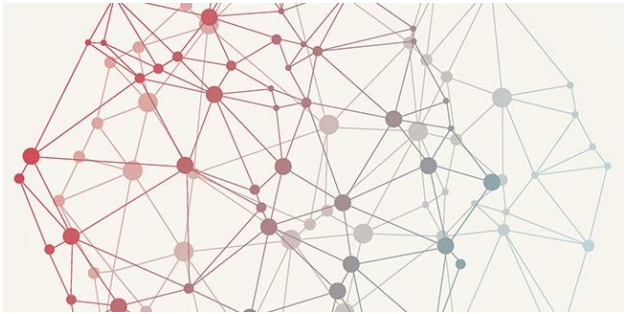
- ตัวอย่าง 3 โปรแกรมทำการอ่านวามยาวและความกว้างของที่ดิน พร้อมความยาวและความกว้างของอาคาร จงคำนวณหาพื้นที่สนามหญ้า พร้อมแสดงผลทางหน้าจอ
- ข้อมูลออก : พื้นที่สนามหญ้า
- ข้อมูลเข้า : 1. ความยาวและความกว้างของที่ดิน
  2. ความยาวและความกว้างของอาคาร
- วิธีการประมวลผล : ใช้สูตรในการคำนวณ

พื้นที่สนามหญ้า = (ความกว้าง*ความยาวของที่ดิน)-(ความกว้าง*ความยาวของอาคาร)

การออกแบบโปรแกรม (Design a Program)

- **อัลกอริทึม (Algorithm)** หมายถึง แนวคิดอย่างมีเหตุมีผลที่ผู้พัฒนาโปรแกรม โปรแกรมเมอร์ หรือนักวิเคราะห์ระบบ ใช้ในการอธิบายวิธีการทำงานอย่างเป็นขั้นตามลำดับในการที่จะพัฒนาโปรแกรมนั้นๆ ให้กับผู้ที่สนใจหรือผู้ที่เป็นเจ้าของงาน หรือผู้ที่รับผิดชอบได้ทราบถึงขั้นตอนต่างๆ ในการเขียนหรือพัฒนาโปรแกรม

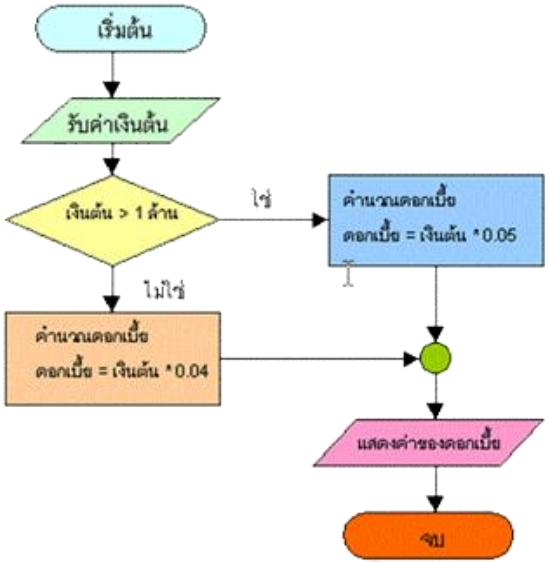
อัลกอริทึม



ลำดับขั้นตอนในการสั่งงาน

การออกแบบโปรแกรม (Design a Program)

- เครื่องมือช่วยอธิบายการทำงาน
  - ผังงาน ซึ่งจะแสดงถึงขั้นตอนการแก้ปัญหาที่ละขั้นตอนในลักษณะ ของรูปภาพทำให้สามารถอ่าน และทำความเข้าใจได้ง่าย



การออกแบบโปรแกรม (Design a Program)

เครื่องมือช่วยอธิบายการทำงาน รหัสจำลอง (Pseudo-code) จะมีรูปแบบเป็นภาษาพูดง่าย ๆ จะเป็นภาษาอังกฤษหรือภาษาไทยก็ได้ โดยจะแสดงขั้นตอนการแก้ปัญหาเป็นขั้นตอนหลัก ๆ แต่ไม่ต้องเจาะเข้าไปในรายละเอียดของการทำงานในแต่ละส่วน

ตัวอย่างรหัสจำลอง

Algorithm mean

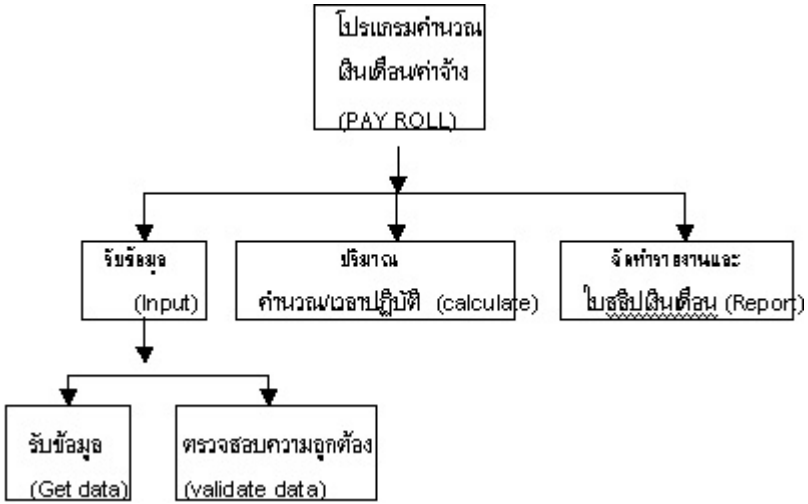
1. total ← 0 , count ← 0
2. Loop (not end of file)
   - 2.1 read number
   - 2.2 total ← total + number
   - 2.3 count ← count + 1
3. average ← total / count
4. Print average
5. end

Algorithm max, min

1. read number
2. max ← number , min ← number
3. Loop (not end of file)
   - 2.1 read number
   - 2.2 if number > max
     - 2.2.1 max ← number
   - 2.3 if number < min
     - 2.3.1 min ← number
4. Print max, min
5. end

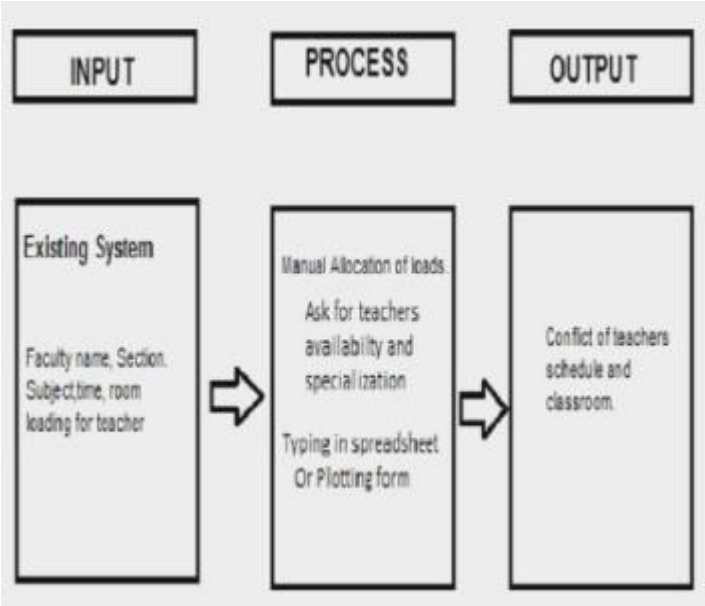
การออกแบบโปรแกรม (Design a Program)

- เครื่องมือช่วยอธิบายการทำงาน
  - <u>แผนภูมิโครงสร้าง (Structure Charts)</u>
  - จะมีลักษณะการแบ่งงานใหญ่ออกเป็นโมดูลย่อย ๆ ซึ่งเรียกว่า การออกแบบจากบนลงล่าง (Top-Down Design) และแต่ละโมดูลย่อยก็ยังสามารถแตกออกได้อีกจนถึงระดับที่ล่างสุด ที่สามารถเขียนโปรแกรมได้อย่างง่าย



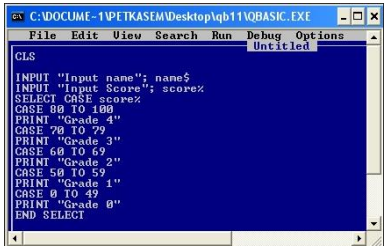
การออกแบบโปรแกรม (Design a Program)

- เครื่องมือช่วยอธิบายการทำงาน
  - ฮิโปชารต์ (HIPO Chart : Hierarchy Input/Output Chart)
  - จะมีการบอกว่าข้อมูลเข้าคืออะไร มีโปรเซสทำอะไรบ้าง และมีผลลัพธ์อะไรบ้าง แต่จะเห็นภาพได้ไม่ชัดเจนเท่ากับแผนภูมิโครงสร้าง



การเขียนโปรแกรมโดยใช้ภาษาใดภาษาหนึ่ง (Coding)

- ในการเขียนโปรแกรมคอมพิวเตอร์นั้น สามารถเลือกใช้ได้หลายภาษา ตัวอย่างของภาษาคอมพิวเตอร์ได้แก่
  - ภาษาเบสิก(BASIC)
  - ภาษาโคบอล(COBOL)
  - ภาษาปาสคาล(PASCAL)
  - ภาษาซี(C)
  - ฯลฯ




```
METHOD-ID. GET PROPERTY Result AS "Result" IS PUBLIC.
DATA DIVISION.
LINKAGE SECTION.
01 PROP-VAL PIC S9(9) COMP-5.
PROCEDURE DIVISION RETURNING PROP-VAL.
    COMPUTE PROP-VAL = WS-RESULT.
END METHOD.

METHOD-ID. SET PROPERTY Result AS "Result" IS PUBLIC.
DATA DIVISION.
LINKAGE SECTION.
01 PROP-VAL PIC S9(9) COMP-5.
PROCEDURE DIVISION USING BY VALUE PROP-VAL.
    COMPUTE WS-RESULT = PROP-VAL.
END METHOD.
```

การตรวจสอบข้อผิดพลาดของโปรแกรม (Testing and Debugging the Program)

- สามารถแบ่งรูปแบบของข้อผิดพลาด(Error) นี้ออกได้เป็น 3 แบบ
  - Syntax Error
    - เป็นข้อผิดพลาดจากการใช้ไวยกรณ์ของภาษาที่ผิด
  - Run-time Error
    - เป็นข้อผิดพลาดที่ทำให้เกิดความผิดปกติทางด้านการทำงานของโปรแกรมในระหว่างการปฏิบัติงาน(Execution)
  - Logical Error
    - เกิดจากการตีความหมายของปัญหาผิดไป

การทดสอบความถูกต้องของโปรแกรม (Testing and Validating)

- การทดสอบความถูกต้องของข้อมูลจะมีอยู่หลายวิธีดังนี้
  - กรณีที่ข้อมูลถูกต้อง
    - จะทดสอบโดยใส่ข้อมูลที่ถูกต้องลงในโปรแกรม เพื่อทดสอบผลลัพธ์ที่ได้ว่าตรงกับที่ต้องการหรือไม่
  - การใช้ขอบเขตและความถูกต้องของข้อมูล
    - เช่น วันที่ที่เป็นไปได้จะต้องไม่เกินวันที่ 31 ถ้าเกินจะต้องไม่ผ่าน
  - การใช้ความสมเหตุสมผล
    - เช่น ถ้ามีฟิลด์(Field) ข้อมูลที่เป็นเพศ(หญิง,ชาย)
  - ข้อมูลที่เป็นตัวเลขและตัวอักษร
  - ข้อมูลเป็นไปตามข้อกำหนด
    - ข้อมูลที่ป้อนต้องเป็นไปตามที่กำหนดไว้แน่นอนแล้วเท่านั้น

การทดสอบความถูกต้องของโปรแกรม (Testing and Validating)

- ในการทดสอบโปรแกรมนี้ยังสามารถแบ่งได้อีก 2 แบบ
  - Program Testing เป็นการทดสอบโปรแกรมแต่ละโปรแกรมต่างหาก แล้วแก้ไขข้อผิดพลาดที่เกิดขึ้น
  - System Testing กรณีที่มีการเขียนโปรแกรมที่เป็นระบบ

การทำเอกสารประกอบโปรแกรม (Documentation)

- เอกสารประกอบโปรแกรมจะมีอยู่ 2 แบบ
  - เอกสารประกอบโปรแกรมสำหรับผู้ใช้ (User Documentation)
    - โปรแกรมนี้ทำอะไร ใช้งานในด้านไหน
    - ข้อมูลเข้ามีลักษณะอย่างไร
    - ข้อมูลออกหรือผลลัพธ์มีลักษณะอย่างไร
    - การเรียกใช้โปรแกรมทำอย่างไร
    - คำสั่งหรือข้อมูล ที่จำเป็นให้โปรแกรมเริ่มทำงาน มีอะไรบ้าง
    - อธิบายเกี่ยวกับประสิทธิภาพ และความสามารถของโปรแกรม
  - เอกสารประกอบโปรแกรมสำหรับผู้เขียนโปรแกรม (Technical Documentation)
    - ส่วนที่เป็นคำอธิบายหรือหมายเหตุในโปรแกรม
    - ส่วนอธิบายด้าน Technical

การบำรุงรักษาโปรแกรม (Program Maintenance)

- เมื่อโปรแกรมผ่านการตรวจสอบตามขั้นตอนเรียบร้อยแล้ว และถูกทำมาให้ผู้ใช้(user) ได้ใช้งาน ในช่วงแรกผู้ใช้อาจจะยังไม่คุ้นเคย ก็อาจทำให้เกิดปัญหาขึ้นมาบ้าง
- ดังนั้นจึงต้องมีผู้คอยควบคุมดูแลและคอยตรวจสอบการทำงาน
- ซึ่งเมื่อมีการใช้งานไปนาน ๆ ก็อาจจะต้องมีการปรับปรุงแก้ไขโปรแกรมให้เหมาะกับเหตุการณ์ และความต้องการของผู้ใช้ที่เปลี่ยนแปลงไป

โครงสร้างโปรแกรม

- การออกแบบจากบนลงล่าง (Top-Down Design)
  - โดยการแบ่งปัญหาออกเป็นส่วน ๆ ในแต่ละส่วนมีรายละเอียดของ การทำงานอย่างใดอย่างหนึ่งเสร็จสมบูรณ์ในตัว ภายหลังแบ่งหน้าที่การทำงานแล้วจึงทำการเขียนเป็น ภาษาคอมพิวเตอร์
- การออกแบบส่วนจำเพาะ หรือ โมดูล (Modular design)
  - โดยการจับกลุ่มการทำงานที่มีลักษณะการทำงานอย่างเดียวกันเข้าด้วยกัน
- ทฤษฎีโครงสร้าง (Structure Theorem)
  - โปรแกรมคอมพิวเตอร์ทุกโปรแกรมสามารถเขียนโดยใช้โครงสร้างควบคุมพื้นฐาน 3 แบบ อันได้แก่ แบบเรียงลำดับ (Sequence) แบบทางเลือก (Selection) และแบบวนซ้ำ (Repetition)

บทสรุป

- ขั้นตอนการพัฒนาโปรแกรมสามารถแบ่งได้ 7 ขั้นตอนคือ
  - (1) ขั้นตอนการวิเคราะห์ปัญหา (Analysis the Problem)
  - (2) ขั้นตอนการออกแบบโปรแกรม (Design a Program)
  - (3) ขั้นตอนการเขียนโปรแกรมโดยใช้ภาษาใดภาษาหนึ่ง (Coding)
  - (4) ขั้นตอนการตรวจสอบข้อผิดพลาดของโปรแกรม (Testing and Debugging)
  - (5) ขั้นตอนการทดสอบความถูกต้องของโปรแกรม (Testing and validating)
  - (6) ขั้นตอนการทำเอกสารประกอบโปรแกรม (Documentation)
  - (7)ขั้นตอนการบำรุงรักษาโปรแกรม (Program Maintenance)

แบบฝึกหัดท้ายบท

- จงเขียนขั้นตอนวิธีแสดงการรับตัวอักษร 3 ตัวจากหน้าจอ และแสดงข้อความตอบรับจากหน้าจอ เช่น “Welcome”
- โปรแกรมทำการรับตัวเลข 2 จำนวน จากหน้าจอ ให้แสดงผลรวมและผลต่างของเลข 2 จำนวนทางหน้าจอเช่นกัน
- โปรแกรมทำการอ่านอัตราภาษีเป็นเปอร์เซ็นต์ พร้อมราคาสินค้าที่ขาย 5 ชนิด โปรแกรมทำการหาผลรวมราคาขายก่อนคิดภาษี การคิดภาษีให้นำอัตราภาษีคูณด้วยยอดขายรวม ให้พิมพ์ยอดขายรวม ภาษี และยอดขายรวมภาษี

แบบฝึกหัดท้ายบท

- โปรแกรมทำการอ่านยอดคงเหลือในบัญชีต้นเดือน ยอดเงินฝากตลอดเดือนและยอดถอนตลอดเดือน ธนาคารคิดค่าบริการ 1% จากยอดเงินฝาก และถอน จงพิมพ์ยอดคงเหลือสิ้นเดือน โดยบวกยอดเงินฝากหัดยอดเงินถอน และเงินค่าบริการ จากยอดบัญชีต้นเดือน
- จงหาค่าจ้างทั้งสิ้นต่อสัปดาห์ โดยที่โปรแกรมทำการอ่านจำนวนชั่วโมงทำงานตามปกติ จำนวนชั่วโมงทำงานล่วงเวลาและอัตราค่าจ้าต่อชั่วโมง ค่าจ้างปกติคิดจากชั่วโมงทำงานปกติ กับอัตราค่าจ้างต่อชั่วโมง ส่วนค่าจ้างล่วงเวลาคิดเป็นเท่าครึ่งของอัตราปกติ ค่าจ้างต่อสัปดาห์คิดจากค่าจ้างปกติรวมกับค่าจ้างล่วงเวลา