บทที่ 7
การเขียนโปรแกรมเบื้องต้น : แนวโครงสร้าง

วัตถุประสงค์ของการเรียน

- ทำความเข้าใจว่า โปรแกรมเมอร์ ทำอะไรและไม่ต้องทำอะไร
- เรียนรู้ว่า โปรแกรมเมอร์ นิยามปัญหา วางแผนแก้ปัญหา ลงรหัส ทดสอบ และทำเอกสารโปรแกรมอย่างไร

เขียนโปรแกรมทำไม? (Why Programming?)

- โปรแกรม หมายถึง เซตของ คำสั่งทำทีละขั้นตอนสั่งคอมพิวเตอร์ให้ทำงาน ซึ่งเราต้องการให้มันทำ และให้ผลลัพธ์ออกมา

  (A program is a set of step-by-step instructions that directs the computer to do the tasks you want it to do and produce the results you want.)

- เซตของกฎต่างๆ ซึ่งจัดหาวิธีบอกคอมพิวเตอร์ว่า ปฏิบัติการอะไรที่จะกระทำ เรียกว่า ภาษาเขียนโปรแกรม

  (A set of rules that provides a way of telling a computer what operations to perform is called a programming language.)

โปรแกรมเมอร์ทำอะไร (What Programmers Do)

- โดยทั่วไป งานของโปรแกรมเมอร์ คือ แปลงผันการแก้ปัญหาของผู้ใช้ ให้เป็นคำสั่งสำหรับคอมพิวเตอร์

  (In general, the programmer's job is to convert user problem solutions into instructions for the computer.)

- โปรแกรมเมอร์ เตรียมคำสั่งต่างๆ ของ โปรแกรมคอมพิวเตอร์ และ วิ่ง ทดสอบ และแก้ไขโปรแกรมให้ถูกต้อง นอกจากนี้แล้ว โปรแกรมเมอร์ ต้องเขียน รายงานโปรแกรม ด้วยเช่นกัน

  (The programmer prepares the instructions of a computer program and runs, tests, and corrects the program. The programmer also writes a report on the program.)

- โปรแกรมเมอร์ ช่วยเหลือผู้ใช้ โดยพัฒนาโปรแกรมใหม่ เพื่อแก้ไขปัญหา แยกข้อผิดพลาดออกจากโปรแกรม หรือทำการเปลี่ยนแปลงในโปรแกรม ให้เป็นผลลัพธ์ของความต้องการใหม่

  (Programmers help the user develop new programs to solve problems, weed out errors in existing programs, or perform changes on programs as a result of new requirements.)

- โปรแกรมเมอร์ โดยปกติโต้ตอบกับ บุคคลต่างๆ หลากหลาย ได้แก่ โปรแกรมเมอร์อื่นๆ และมืออาชีพคอมพิวเตอร์ ผู้ใช้ และผู้จัดการ

  (A programmer typically interacts with a variety of people : other programmers and computer professionals, users, and managers.)

กระบวนการเขียนโปรแกรม (The Programming Process)

- การพัฒนาโปรแกรม มี ห้าขั้นตอน ดังนี้

1. นิยามปัญหา (define the problem)
2. วางแผนแก้ปัญหา (plan the solution)
3. ลงรหัสโปรแกรม (code the program)
4. ทดสอบโปรแกรม (test the program)
5. ทำเอกสารโปรแกรม (document the program)

I, **การนิยามปัญหา** (Defining the Problem)

- โปรแกรมเมอร์ (บ่อยครั้ง รวมทั้ง นักวิเคราะห์ระบบ) พบกับ ผู้ใช้ จาก องค์กรลูกค้า เพื่อวิเคราะห์ปัญหา

(The programmer (often through the systems analyst), meets with users from the client organization to analyze the problem.)

- ในที่สุด เขาเหล่านี้ มีความเห็นตรงกัน เกี่ยวกับสิ่งอื่นๆ กำหนดชนิดของอินพุต การประมวลผล และเอาต์พุตที่ต้องการ

(Eventually they come to an agreement which, among other things, specifies the kind of input, processing, and output required.)

2. **การวางแผนแก้ปัญหา** (Planning the Solution)

- มีสองวิธี ซึ่งใช้วางแผนแก้ปัญหา คือ วาดผังงาน และ/หรือ เขียนรหัสเทียม

(Two common ways of planning to a problem are to draw a flowchart and/or write pseudocode.)

- ผังงาน หมายถึง การแทนที่เชิงภาพของลำดับการแก้ไขปัญหาทำทีละขั้นตอน

(A **flowchart** is a pictorial representation of an ordered, step-by-step solution to a problem.)

- **รหัสเทียม** หมายถึง ภาษาคล้ายกับภาษาอังกฤษ ซึ่ง เราสามารถใช้กล่าวถึง การแก้ปัญหาด้วยความถูกต้องมากกว่าการใช้ภาษาอังกฤษธรรมดา แต่ มีความถูกต้องน้อยกว่า เมื่อเทียบกับการใช้ภาษาเขียนโปรแกรมแบบทางการ

(**pseudocode** is an English-like language that you can use to state your solution with more precision than you can in plain English but with less precision than is required when using a formal programming language.)

3. **การลงรหัสโปรแกรม** (Coding the Program)

- โปรแกรมเมอร์ แปล (translates) ตรรกะ จากผังงาน หรือ รหัสเทียม หรือ เครื่องมืออื่น ให้เป็นภาษาโปรแกรม
- ภาษาโปรแกรม ปฏิบัติการเชิงไวยากรณ์คล้ายกับภาษาอังกฤษ แต่ภาษาโปรแกรมมีความถูกต้องมากกว่า
- โปรแกรมเมอร์ต้องทำตามกฎต่างๆ อย่างเคร่งครัด คือ **วากยสัมพันธ์** (text syntax) ของภาษาที่ใช้
- โปรแกรมซึ่งลงรหัสแล้ว ต้อง คีย์ โดยใช้เทอร์มินอล (terminal) หรือ คอมพิวเตอร์ ส่วนบุคคล ในรูปแบบซึ่ง คอมพิวเตอร์สามารถเข้าใจได้
- โปรแกรมเมอร์ ปกติใช้ **บรรณาธิกรณ์ข้อความ** (text editor) ซึ่งเป็นโปรแกรม คล้ายโปรแกรมประมวลผลคำ เพื่อสร้างไฟล์ ซึ่งประกอบด้วยโปรแกรม อย่างไรก็ตาม ผู้เขียนโปรแกรมมือใหม่ ปกติจำเป็นต้องเขียนรหัสโปรแกรมลงบนกระดาษก่อนเป็นสิ่งแรก

4. **การทดสอบโปรแกรม** (Testing the Program)

- การทดสอบโปรแกรม เกี่ยวข้องกับขั้นตอนต่อไปนี้

การตรวจสอบด้วยมือ (Desk-checking)

- ใน desk-checking, โปรแกรมเมอร์ตามรอยในใจ หรือ ตรวจสอบตรรกะของโปรแกรม เพื่อให้เชื่อมั่นว่า ไม่มีข้อผิดพลาดในโปรแกรมและทำงานได้

  (In desk-checking, a programmer mentally traces, or checks, the logic of the program to ensure that it is error-free and workable.)

- องค์กรจำนวนมาก ทำขั้นตอนนี้เพิ่มคือ การตรวจตลอด หมายถึง กระบวนการซึ่งกลุ่มของผู้ร่วมเขียนโปรแกรม ตรวจทานโปรแกรม และให้คำแนะนำ ในวิธีของเพื่อนร่วมงาน

  (Many organizations take this phase a step further with a **walkthrough**, a process in which a group of programming peers review the program and offer suggestions in a collegial way.)

การแปลภาษา (Translating)

- ตัวแปลภาษา (translator) หมายถึงโปรแกรมซึ่ง

  (1) ตรวจสอบวากยสัมพันธ์ ของ โปรแกรมของเรา เพื่อทำให้แน่ใจว่า ภาษาโปรแกรมใช้อย่างถูกต้อง ให้ข้อความผิดพลาดวากยสัมพันธ์ทั้งหมด (all the syntax-error messages) ซึ่งเรียกว่า **diagnostics** และ

  (2) จากนั้น แปล โปรแกรมของเรา ให้เป็นรูปแบบที่คอมพิวเตอร์สามารถเข้าใจได้

- ตัวแปลภาษา จะบอกเรา ถ้าเราใช้ภาษาโปรแกรมในวิธีที่ไม่ถูกต้อง ข้อผิดพลาดชนิดนี้ เรียกว่า **ข้อผิดพลาดวากยสัมพันธ์** (syntax errors.)
- **คอมไพเลอร์** แปลโปรแกรมทั้งหมด ในครั้งเดียว

  (A **compiler** translates the entire program at one time.)
- การแปลภาษา เกียวข้องกับ โปรแกรมเริ่มต้น เรียกว่า **มอดูลต้นฉบับ** (source module) ซึ่งจะถูกแปล โดย คอมไพเลอร์ให้เป็น **มอดูลจุดหมาย** (object module)
- โปรแกรมซึ่งถูกเขียนมาแล้ว จากคลังระบบ 018 ใส่เพิมระหว่าง ขั้นตอน เชื่อม / บรรจุ ซึ่ง ผลลัพธ์ จะเป็น **มอดูลบรรจุ**

  (Prewritten programs from a system library may be added during the **link / load phase** which results in a **load module.)**
- หลังจากนั้น มอดูลบรรจุ จะถูกกระทำการ โดยคอมพิวเตอร์

  (The load module can then be executed by the computer.)

การแก้จุดบกพร่อง (Debugging)

- คำซึ่งใช้กันอย่างกว้างขวางในการเขียนโปรแกรม การแก้จุดบกพร่อง ซึ่งหมายถึง การตรวจจับหาตำแหน่ง และแก้ไขข้อผิดพลาด โดยการวิงโปรแกรม

  (A term used extensively in programming, **debugging** is detecting, locating, and correcting "bugs" (mistakes) by running the program.)
- ข้อผิดพลาดเหล่านี้ในโปรแกรม เรียกว่า **ข้อผิดพลาดตรรกะ**

  (These program mistakes are called **logic errors.)**

5. **การทำเอกสารของโปรแกรม** (Documentingthe Program)

- การทำเอกสาร หมายถึง การเขียนอธิบายในรายละเอียดของวัฏจักร์การเขียนโปรแกรม และกำหนดข้อเท็จจริง เกี่ยวกับโปรแกรมนี้ เปนกระบวนการที่จำเป็นต้องกระทำอย่างต่อเนื่อง

  (**Documentation**, a written detailed description of the programming cycle and specific facts about this program, is an ongoing, necessary process.)
- ปกติ วัตถุดิบของการทำเอกสารโปรแกรม ได้แก่ ตั้งแต่การเริ่มต้น และธรรมชาติของปัญหา รายละเอียดของโปรแกรม ผังงานและรหัสเทียม รายละเอียดของระเบียนข้อมูล รายการของโปรแกรม และผลลัพธ์ของการทดสอบโปรแกรม

  (Typical program documentation materials include the origin and nature of the problem, a description of the program, flowcharts and pseudocode, data record descriptions, program

listings, and testing results.)

· คอมเมนต์ ในโปรแกรมตัวมันเองถือว่าเป็นส่วนสำคัญอย่างหนึ่ง ของการทำเอกสาร

(Comments in the program itself are also considered an important part of documentation.)

แบบฝึกหัด 7.1 จงเลือกคำตอบที่ถูกต้องที่สุดเพียงหนึ่งตัวเลือก

1. ข้อใดคือ ลำดับขั้นตอนของการเขียนโปรแกรม

   1. วางแผนแก้ปัญหา อภิปราย ตรวจสอบ ลงรหัส จัดพิมพ์เอกสาร
   2. วางแผนแก้ปัญหา ลงรหัส ตรวจสอบ จัดพิมพ์เอกสาร อภิปราย
   3. จัดพิมพ์เอกสาร ตรวจสอบ ลงรหัส วางแผน อภิปราย
   4. อภิปรายปัญหา วางแผน ลงรหัส ตรวจสอบ จัดพิมพ์เอกสาร

2. ข้อใด **ไม่ใช่** การวางแผนแก้ปัญหา

   1. เขียนผังงาน 2. เขียนคำสั่งเทียม 3. เขียนโปรแกรม 4. ไม่มีข้อใดถูก

3. การเรียงลำดับด้วยภาพ สัญลักษณ ในขั้นตอนของการแก้ไขปัญหา เรียกว่า

   1. โปรแกรม 2. ผังงาน 3. รหัสเทียม 4. ถูกทุกข้อ

ข้อ 4 - 8 ให้ใช้ตัวเลือกข้างล่างนี้ เป็นคำตอบ

(1) (2) (3) (4)

4. ข้อใดคือ decision block
5. ข้อใดคือ connector
6. ข้อใดคือ processing block
7. ข้อใดคือ start/stop
8. ข้อใดคือ input/output
9. ขั้นตอนของกระบวนการพัฒนาโปรแกรม เรียงลำดับดังนี้

   1. defining the problem, planning, testing, documenting, coding
   2. planning, coding, defining the problem, documenting, testing
   3. coding, testing, planning, documenting, defining the problem
   4. defining the problem, planning, coding, testing, documenting

10. ข้อใดคือ ขั้นตอนแรกในการพัฒนาโปรแกรม

1. planning a solution
2. documenting the program
3. defining the program
4. testing the program

11. หลังจากแก้ปัญหาด้วยรหัสเทียมแล้ว ขั้นตอนต่อไปของกระบวนการเขียนโปรแกรม คือ

1. ทดสอบโปรแกรม
2. ลงรหัสโปรแกรม
3. ทำโปรแกรมให้เกิดผลในทางปฏิบัติ
4. แปลโปรแกรม

12. ในการเตรียมโปรแกรม, desk-checking และ translating เป็นตัวอย่างของ

1. coding
2. planning
3. testing
4. documenting

13. การทดสอบลำดับคำสั่งของโปรแกรมเพื่อให้มั่นใจว่าจะให้ผลลัพธ์ถูกต้อง เรียกว่า

1. syntax testing
2. documentation
3. logic testing
4. diagnostic testing

14. ข้อผิดพลาดในโปรแกรม ซึ่งเกี่ยวกับการใช้ภาษาโปรแกรมไม่ถูกต้อง เรียกว่า

1. diagnostics
2. attributes
3. logic error
4. syntax error

15. ข้อผิดพลาดชนิดใด ซึ่งตัวแปลโปรแกรมไม่สามารถตรวจพบได้

1. logic error
2. syntax error
3. message error
4. diagnostic error

16. คำสั่งงาน ซึ่งทำให้เกิดภาษาเดิมของคอมพิวเตอร์เครื่องหนึ่ง เรียกว่า

1. compiler
2. assembler
3. instruction set
4. editor

17. ข้อใด**ไม่ใช่** บล็อกของรหัสโปรแกรม

1. procedure
2. subroutine
3. function
4. link

18. รหัสเทียม หมายถึง

1. ข้อความสั่งเหมือนภาษาอังกฤษอธิบายขั้นตอนการแก้ปัญหา
2. เครื่องมือที่ใช้พัฒนาตรรกะของโปรแกรม
3. ภาษาเขียนอัลกอริทึม
4. ถูกทุกข้อ

19. ข้อใด **ไม่ใช่** คุณสมบัติของรหัสเทียม

1. คล้ายกับข้อความสั่งธรรมชาติ
2. คล้ายกับรหัสของโปรแกรม
3. มีข้อจำกัดน้อยกว่าภาษาโปรแกรม
4. ไม่มีกฎไวยากรณ์ใดๆ

20. ข้อใดคือโครงสร้างควบคุม ซึ่งใช้ในการเขียนโปรแกรมแบบโครงสร้าง

1. selection structure
2. object selection
3. repetition objects
4. ถูกทุกข้อ

21. โครงสร้างควบคุมชนิดใด ซึ่งนำมาใช้เมื่อทดสอบเงื่อนไขแล้วนำไปสู่ทางเลือกมากกว่าสองทางเลือก

1. iteration 2. do-while 3. do-until 4. case

22. ข้อใด **ไม่ใช่** โครงสร้างควบคุมที่ใช้ในการเขียนอัลกอริทึม

1. selection 2. sequence 3. condition 4. repetition

23. ข้อใดคือ โครงสร้างควบคุมซึ่งกระทำการคำสั่งทั้งหมดในส่วนวนซ้ำ **หนึ่งครั้งก่อน** แล้วจึงตรวจสอบเงื่อนไข

1. DOWHILE 2. DOUNTIL 3. SELECTION 4. SEQUENCE

24. โครงสร้างควบคุมที่มีการตรวจสอบเงื่อนไข**ก่อน** ที่จะตัดสินใจทำซ้ำ หรือไม่ทำซ้ำ เรียกว่า

1. DOUNTIL 2. DOWHILE 3. SELECTION 4. ถูกทุกข้อ

25. โครงสร้างชนิดใด ซึ่ง กรรมวิธีอย่างหนึ่งเกิดขึ้นทันทีหลังจากกรรมวิธีอีกอย่างหนึ่ง

1. sequence 2. selection 3. if-then-else 4. looping

26. ข้อใด **ไม่ใช่** คุณสมบัติของการเขียนโปรแกรมแบบโครงสร้าง

1. เขียนง่าย 2. อ่านง่าย 3. แก้ไขที่ผิดง่าย 4. มีลักษณะเป็นสากล

27. แนวคิดของแต่ละโครงสร้าง จะมีเพียงหนึ่ง

1. entry point 2. exit point 3. loop 4. ข้อ 1 และข้อ 2

28. ผังงานซึ่งแสดงการไหล (flow) ของข้อมูลผ่านระบบคอมพิวเตอร์ เรียกว่า

1. logic flowchart 2. system flowchart 3. syntax diagram 4. structure chart

29. โปรแกรมแปลคำสั่ง ได้แก่

1. assembler 2. interpreter 3. translator 4. compiler

30. โปรแกรมซึ่งเขียนด้วยภาษาระดับสูง เรียกว่า

1. source program 2. object program
3. compiled program 4. application program

31. โปรแกรมซึ่งเขียนด้วยภาษาระดับสูงและถูกแปลให้เป็นโปรแกรมภาษาเครื่องที่มีความหมายเหมือนกัน โปรแกรมชุดหลังนี้ เรียกว่า

1. source program 2. translator program
3. object program 4. interpreter program

32. แนวคิดของแต่ละโครงสร้าง จะมีเพียงหนึ่ง

1. structured 2. object-oriented
3. low-level 4. portable

33. ชุดของคำสั่งซึ่งโปรแกรมเมอร์ พิมพ์ไปในคอมพิวเตอร์ เรียกว่า ..... โปรแกรม

1. object code 2. source code 3. link code 4. definition

34. ขั้นตอนเชิงตรรกะของโปรแกรม ซึ่งถูกแทนที่ด้วยการรวมกันของสัญลักษณ์และข้อความ เรียกว่า

1. pseudocode 2. program flowchart
3. desk checking 4. structured walkthrough

35. ชุดของคำสั่งซึ่งโปรแกรมเมอร์พิมพ์เข้าไปในคอมพิวเตอร์ เรียกว่า

1. object code 2. source code 3. link code 4. definition

36. ข้อใดคือโปรแกรม ซึ่งแปลงผัน (converts) ข้อความสั่งหนึ่งคำสั่งของภาษาระดับสูง ให้เป็นภาษาเครื่องและกระทำการทันที่ทำทีละคำสั่ง

1. assembler 2. interpreter 3. translator 4. implementer

37. โปรแกรม ซึ่งแปลงผันโดยคอมไพเลอร์ เรียกว่า และโปรแกรมภาษาเครื่องซึ่งได้มา เรียกว่า.....

1. object, source 2. data, information 3. source, object 4. information, data

38. ข้อใดคือ ตัวแปลโปรแกรม (translator)

1. compiler 2. assembler 3. interpreter 4. ถูกทุกข้อ

39. ข้อใดคือ ตัวแปลโปรแกรม

1. compiler 2. assembler 3. interpreter 4. translator

40. “Spaghetti code” หมายถึง

1. โปรแกรมที่เขียนยาก 2. โปรแกรมที่มีข้อความสั่ง GOTO จำนวนมาก
3. โปรแกรมที่แก้ไขที่ผิดไม่ได้ 4. ถูกทุกข้อ