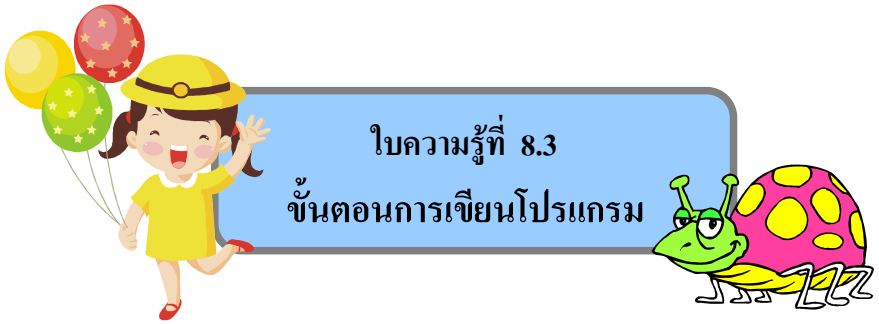


ใบความรู้ที่ 8.3
ขั้นตอนการเขียนโปรแกรม

ในการเตรียมการเพื่อดำเนินการเขียนโปรแกรมคอมพิวเตอร์ โปรแกรมเมอร์ควรปฏิบัติตามขั้นตอนพื้นฐานต่อไปนี้ ซึ่งประกอบด้วย

1. การกำหนดปัญหา (Defines the problem)
2. การพัฒนาอัลกอริทึมเพื่อแก้ปัญหา (Develops the algorithm for solution)
3. การออกแบบโปรแกรม (Designs the program)
4. การเขียนโปรแกรม (Codes the programming statements)
5. การคอมไพล์โปรแกรม (Compiles the program)
6. การแก้ไขโปรแกรม (Debugs the program)
7. การจัดทำเอกสารประกอบโปรแกรม (Document the program)

1. การกำหนดปัญหา (Defines the problem)

เป็นขั้นตอนแรกสำหรับการเขียนโปรแกรม กล่าวคือ โปรแกรมเมอร์จะต้องกำหนดว่าสิ่งที่ต้องการคืออะไร อะไรเป็นต้นเหตุของปัญหานั้น เกี่ยวข้องกับอะไร ต้องมีข้อมูลจำเข้าอะไรบ้างที่จะทำให้เกิดผลลัพธ์ตามที่ต้องการ

2. การพัฒนาอัลกอริทึมเพื่อแก้ปัญหา (Develops the algorithm for solution)

อัลกอริทึมในที่นี้หมายถึง สูตรทางคณิตศาสตร์ที่ใช้สำหรับการแก้ปัญหา เช่น ถ้าต้องมีการคำนวณค่าสินค้าที่ได้รวมภาษีมูลค่าเพิ่ม โปรแกรมเมอร์ก็จะต้องพัฒนาอัลกอริทึมการคำนวณสินค้ารวมภาษี ด้วยการนำราคาสินค้าก่อนรวมภาษีมาคูณกับอัตราภาษีมูลค่าเพิ่ม ก็จะได้จำนวนภาษามูลค่าเพิ่มของสินค้านั้น จากนั้นก็นำภาษามูลค่าเพิ่มที่ได้ไปรวมกับราคาสินค้าเพื่อจะได้ราคาสินค้าเพื่อจะได้ราคาสินค้าที่รวมภาษีไว้เรียบร้อยแล้ว เป็นต้น

3. การออกแบบโปรแกรม (Designs the program)

ในขั้นตอนนี้โปรแกรมเมอร์จะต้องทำการวางแผนการเขียนโปรแกรมด้วยการออกแบบให้มีเวลาการประมวลผล (Processing Time) ที่ใช้เวลาน้อยที่สุด ซึ่งเครื่องมือที่มักนำมาใช้งานเพื่อการนี้ก็คือผังงาน (Flowchart) ซึ่งจะช่วยให้ทราบถึงลำดับขั้นตอนการทำงานหรือการประมวลผล

4. การเขียนโปรแกรม (Codes the programming statements)

กิจกรรมต่อไปหลังจากที่ได้ผ่าน 3 ขั้นตอนแรกเป็นที่เรียบร้อยแล้ว ก็คือการลงมือเขียนโปรแกรมด้วยภาษาคอมพิวเตอร์ ซึ่งขั้นตอนนี้มักเรียกว่าการโค้ด (Coding) โดยโปรแกรมเมอร์จะใช้ประโยชน์จากผังงานที่ได้เขียนไว้ก่อนหน้ามาเป็นแนวทางในการเขียนโปรแกรมภาษา

5. การคอมไพล์โปรแกรม (Compiles the program)

โปรแกรมที่เขียนขึ้นด้วยภาษาคอมพิวเตอร์จะเรียกว่าซอร์สโค้ด (Source Code) ซึ่งโปรแกรมดังกล่าวยังไม่สามารถทำงานได้ทันที จำเป็นต้องผ่านการแปล (Translated) เป็นภาษาเครื่องที่คอมพิวเตอร์เข้าใจ ซึ่งตัวแปลภาษาก็จะมีทั้งนิดแบบคอมไพล์เลอร์และอินเตอร์พรีเตอร์

6. การทดสอบและแก้ไขโปรแกรม (Test and Debugs the program)

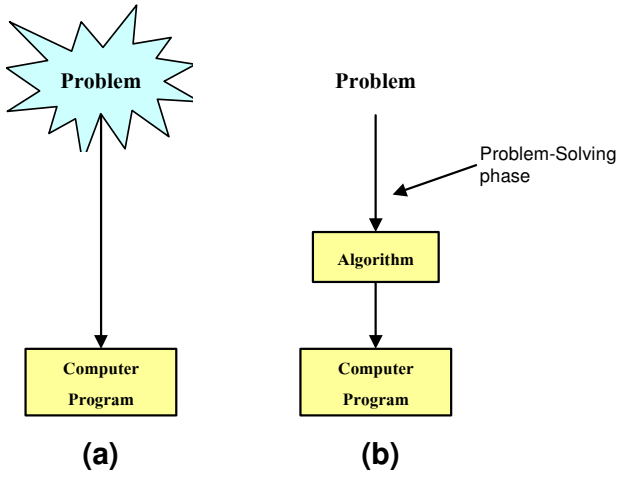
โปรแกรมคอมพิวเตอร์หลังจากที่ได้เขียนและผ่านการคอมไพล์เป็นที่เรียบร้อยแล้ว จำเป็นต้องไดรับการทดสอบโปรแกรม ซึ่งหากเกิดข้อผิดพลาดก็จะต้องดำเนินการแก้ไขให้ถูกต้อง

7. การจัดทำเอกสารประกอบโปรแกรม (Document the program)

เอกสารประกอบโปรแกรมจะช่วยอธิบายถึงจุดประสงค์ของโปรแกรม อธิบายและบอกแนวทางแก้ไขหากเกิดข้อผิดพลาดชนิดนี้ขึ้นมา ซึ่งเอกสารเหล่านี้จำเป็นต้องมีไว้เพื่อใช้ในกรณีที่เกิดข้อผิดพลาดต่าง ๆ และรวมถึงการปรับปรุงแก้ไขในอนาคต

การแก้ปัญหาโจทย์ทางคอมพิวเตอร์

การแก้ปัญหาโจทย์ทางคอมพิวเตอร์ในที่นี้จะนำขั้นตอนการเขียนโปรแกรม 3 ขั้นตอนแรกของขั้นตอนการเขียนโปรแกรมที่ได้กล่าวมาแล้วในหัวข้อข้างต้นมาประยุกต์ใช้ เพื่อจะให้นักเรียนได้รับการฝึกฝนให้มีทักษะในกระบวนการแก้ปัญหาโจทย์ทางคอมพิวเตอร์อย่างมีระบบ ด้วยการวิเคราะห์และออกแบบเพื่อนำไปพัฒนาหรือเขียนในรูปแบบของโปรแกรมภาษาต่อไป โดยพิจารณาภาพที่ 4.18 เป็นการเปรียบเทียบการพัฒนาโปรแกรม โดยภาพที่ 4.18 (a) เป็นการพัฒนาโปรแกรมจากปัญหาโดยตรง ซึ่งอาจก่อให้เกิดปัญหาตามมา กล่าวคือในระหว่างการพัฒนาอาจมีการปรับแก้อยู่เสมอ เนื่องจากอาจค้นพบปัญหาอยู่เรื่อย ๆ เพราะไม่มีการวิเคราะห์และเตรียมการมาก่อน และหากเป็นงานที่มีความซับซ้อนก็ไม่ควรนำวิธีนี้มาใช้งานเป็นอย่างยิ่ง ในขณะที่ภาพที่ 4.18 (b) นั้น จะมีขั้นตอนการแก้ปัญหา (problem-Solving Phase) ซึ่งประกอบด้วยขั้นตอนการวิเคราะห์ปัญหา การออกแบบอัลกอริทึมเพื่อแก้ไขปัญหา และการเขียนผังงานหรือรหัสจำลองเพื่อเตรียมไว้สำหรับนำไปพัฒนาเป็นโปรแกรมภาษาต่อไปได้ง่ายอย่างยิ่ง



ภาพที่ 8.18 เปรียบเทียบการพัฒนาโปรแกรม

ขั้นตอนการแก้ปัญหาโจทย์ทางคอมพิวเตอร์

ขั้นตอนการแก้ปัญหาโจทย์ทางคอมพิวเตอร์ประกอบด้วยขั้นตอนต่อไปนี้

1. กำหนดสิ่งที่โจทย์ต้องการ

เป็นขั้นตอนที่จะต้องทำการวิเคราะห์ปัญหาจากโจทย์ กล่าวคือจะต้องตีโจทย์ให้แตก และสามารถบอกได้ว่าสิ่งที่โจทย์ต้องการนั้นคืออะไร ซึ่งความต้องการของโจทย์อาจมีมากกว่าหนึ่งงานก็เป็นไปได้ กระบวนการนี้จัดได้ว่าเป็นขั้นตอนแรกที่สำคัญที่เดียว กล่าวคือหากตีโจทย์ผิดพลาด หรือไม่ครอบคลุม นั่นหมายถึงขั้นตอนที่จะกระทำต่อไปนั้น อาจไม่ตรงหรือไม่เพียงพอต่อวัตถุประสงค์ที่โจทย์ต้องการ

2. กำหนดรูปแบบของผลลัพธ์ (Output)

เป็นขั้นตอนที่ทำการออกแบบผลลัพธ์หรือรายงานที่ต้องการ โดยข้อมูลต่าง ๆ ที่แสดงบนรายงานนั้นจะต้องมีรายละเอียดครบถ้วน ซึ่งรายละเอียดข้อมูลของผลลัพธ์นั้นจะทำให้ทราบถึงข้อมูลนำเข้าว่าควรมีอะไรบ้าง

3. กำหนดข้อมูลนำเข้า (Input)

เป็นขั้นตอนที่ทำการพิจารณาว่าข้อมูลที่ต้องป้อนเข้ามาเพื่อทำการประมวลผลนั้นควรมีข้อมูลอะไรบ้างที่จำเป็นต่อการนำไปประมวลผล เพื่อให้ได้มาซึ่งผลลัพธ์ที่ได้ออกแบบไว้

4. กำหนดตัวแปร

เป็นขั้นตอนการกำหนดตัวแปรที่ใช้แทนความหมายค่าข้อมูลต่าง ๆ ซึ่งการกำหนดตัวแปรควรกำหนดชื่อตัวแปรให้สามารถสื่อความหมาย ได้ดี และถูกต้องตามกฎเกณฑ์การตั้งชื่อตัวแปร เช่น ข้อมูลเลขประจำตัว อาจตั้งชื่อตัวแปรว่า ID_NO ซึ่งสื่อความหมายได้ดีกว่าการตั้งชื่อตัวแปรชื่อว่า A เป็นต้น

5. ขั้นตอนการประมวลผล

เป็นกระบวนการที่ประกอบด้วยขั้นตอนต่าง ๆ ที่ใช้ในการแก้ปัญหา ด้วยการนำข้อมูลนำเข้าและตัวแปรต่าง ๆ ที่ได้กำหนดไว้มาทำการประมวลผล ไม่ว่าจะเป็นการคำนวณ การตรวจสอบเงื่อนไขให้ได้ผลลัพธ์ตามที่ต้องการ

6. เขียนผังงานหรือรหัสจำลอง

หลังจากที่ได้ทราบถึงขั้นตอนการประมวลผลที่ได้ออกแบบไว้แล้ว ขั้นตอนต่อไปก็คือการนำขั้นตอนการประมวลผลเหล่านั้นมาทำการเขียนเป็นผังงานหรือรหัสจำลองหรืออาจเขียนทั้งสองแบบ เพื่อนำไปใช้ประกอบการเขียนโปรแกรมต่อไป

ตัวอย่างการคิดค่าแรงอย่างง่าย

จงคำนวณค่าแรงของพนักงานรายชั่วโมง พร้อมเงินที่ต้องจ่ายแก่พนักงานแต่ละคน และจำนวนเงินทั้งสิ้นที่จะต้องจ่ายให้กับพนักงาน

1. สิ่งที่โจทย์ต้องการ

คำนวณค่าค่าแรงให้กับพนักงานแต่ละคน และยอดเงินรวมที่ต้องจ่ายทั้งสิ้น

2. รูปแบบของผลลัพธ์ (Output)

```
                         << PAYROLL  REPORT >>

ID-No    NAME                  HOURS          RATE            PAY
=====================================================================
XXXXX    XXXXXXXXXXXXXXX        999            999          99 999
XXXXX    XXXXXXXXXXXXXXX        999            999          99 999

=====================================================================
                                       TOTAL PAYMENT       999,999
                                                           ========
```

3. ข้อมูลนำเข้า (Input)

รหัสพนักงาน ชื่อพนักงาน จำนวนชั่วโมงทำงาน อัตราค่าแรงต่อชั่วโมง

4. กำหนดตัวแปรที่ใช้

id_no = รหัสพนักงาน

name = ชื่อพนักงาน

hours = จำนวนชั่วโมงทำงาน

rate = อัตราค่าแรงต่อชั่วโมง

pay = ค่าแรงที่จ่ายให้กับพนักงานแต่ละคน

total = ค่าแรงทั้งหมดที่จ่ายไป

5. ขั้นตอนการประมวลผล

5.1 กำหนดให้ค่าตัวแปร pay, total มีค่าเท่ากับศูนย์

5.2 อ่านข้อมูล id_no, name, hours, rate

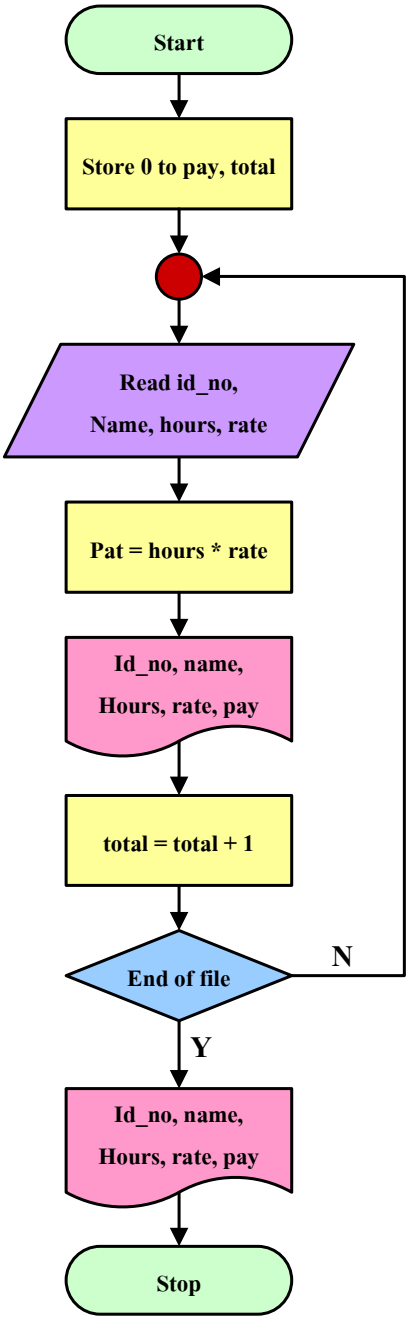
5.3 กำหนด pay = hour * rate

5.4 พิมพ์ค่า id_no, name, hours, rate, pay

5.5 กำหนด total = total + pay

5.6 กลับไปทำขั้นตอนที่ 2 จนกระทั่งจบข้อมูล

5.7 พิมพ์ค่า total

5.8 จบการทำงาน

6. เขียนผังงานหรือรหัสจำลอง