

หน่วยการเรียนรู้ที่ 1

ความรู้เบื้องต้นเกี่ยวกับ

การเขียนโปรแกรมคอมพิวเตอร์



คอมพิวเตอร์เป็นเพียงแค่อุปกรณ์ที่ทำงานตามคำสั่งที่รับมาจากผู้ใช้ ดังนั้นผู้ใช้จึงจำเป็นต้องบอกคอมพิวเตอร์ว่าต้องการให้เครื่องคอมพิวเตอร์ทำอะไร โดยการสั่งงานให้เครื่องคอมพิวเตอร์ทำงานที่ต้องการวิธีการสั่งงานดังกล่าวต้องอาศัยโปรแกรม ซึ่งก็คือกลุ่มของโปรแกรมหรือกลุ่มของคำสั่งที่ใช้ภาษาต่างๆ ในการเขียนโปรแกรมขึ้นมา ซึ่งมีมากมาย เช่น ภาษา Fortran , Cobol , Lisp , PL / 1 , Basic , Pascal , C , Ada , C++ , Perl , Java เป็นต้น แต่ภาษาที่นำมาให้นักเรียนเรียนในวิชาการเขียนโปรแกรมภาษา C คือโปรแกรมภาษา C นั้นเอง ซึ่งเป็นรากฐานที่ดีที่สุดสำหรับผู้เริ่มต้นการเขียนโปรแกรมคอมพิวเตอร์ รวมไปถึงการใช้สัญลักษณ์ Flowchart แทนคำสั่ง โดยไม่อิงถึงคำสั่งในภาษาใดภาษาหนึ่ง ช่วยให้การเขียนโปรแกรมง่ายขึ้น

ก่อนจะเข้าสู่การเขียนโปรแกรมภาษา C นักเรียนควรมีความรู้พื้นฐานเกี่ยวกับการเขียนโปรแกรมก่อน โดยมีเนื้อหาต่างๆ ดังนี้

1. องค์ประกอบของระบบคอมพิวเตอร์

โดยทั่วไปการนำคอมพิวเตอร์มาช่วยในการทำงานต่าง ๆ นั้น ถ้าเฉพาะตัวเครื่องและอุปกรณ์หรือ "ฮาร์ดแวร์"(Hardware) นั้น จะไม่สามารถทำงานได้ เพราะเครื่องยังไม่มีสิ่งที่จะมาจัดการหรือควบคุมระบบต่าง ๆ ดังนั้นจึงต้องมีโปรแกรมที่ใช้ควบคุมระบบและใช้งาน ซึ่งเราเรียกว่า "ซอฟต์แวร์"(Software) แต่เนื่องจากคอมพิวเตอร์เป็นสิ่งที่ไม่มีชีวิตจิตใจเราจึงต้องพึ่งพามนุษย์ หรือ "พีเพิลแวร์" (Pepleware) มาสั่งการให้คอมพิวเตอร์ทำงานตามกระบวนการ (Procedure) ดังนั้นสรุปได้ว่าองค์ประกอบที่จะทำให้เครื่องคอมพิวเตอร์ทำงานได้นั้น ประกอบด้วย 5 ส่วน คือ

1.1 ฮาร์ดแวร์ (Hardware)

สิ่งที่มองเห็นและจับต้องสัมผัสได้ทั้งหมดที่เกี่ยวข้องกับคอมพิวเตอร์ ฮาร์ดแวร์จะไม่สามารถทำงานด้วยตัวเองเดี่ยว ๆ ได้ จะต้องนำมาลงหรือติดตั้งโปรแกรมหรือซอฟต์แวร์ที่เขียนขึ้นเพื่อให้ทำงานได้

1.2 ซอฟต์แวร์ (Software)

ซอฟต์แวร์หรือโปรแกรมคอมพิวเตอร์หรือชุดคำสั่งที่เขียนขึ้น เป็นองค์ประกอบที่สำคัญประการที่สอง ซึ่งก็คือลำดับขั้นตอนของคำสั่งที่จะสั่งงานให้ฮาร์ดแวร์ทำงาน เพื่อประมวลผลข้อมูลให้ได้ผลลัพธ์ตามความต้องการของการใช้งาน ซอฟต์แวร์แบ่งออกเป็น 2 ประเภทคือ

1. ซอฟต์แวร์ระบบปฏิบัติการ (Operating System Software-OS) หมายถึง ซอฟต์แวร์หรือโปรแกรมที่ควบคุมการทำงานทั้งหมดของเครื่อง ได้แก่ Windows , Dos , Linux , Unix , Mac
2. ซอฟต์แวร์ประยุกต์ (Application Software) หมายถึงโปรแกรมที่เขียนขึ้นมาเพื่อสั่งให้เครื่องคอมพิวเตอร์ทำงานเฉพาะด้าน เช่น โปรแกรมระบบบัญชี โปรแกรมออกแบบ โปรแกรมสำเร็จรูปต่างๆ

1.3 ข้อมูล (Data)

ข้อมูล เป็นองค์ประกอบที่สำคัญอีกประการหนึ่งของระบบสารสนเทศ อาจจะเป็นตัวชี้ความสำเร็จหรือความล้มเหลวของระบบได้ เนื่องจากจะต้องมีการเก็บข้อมูลจากแหล่งกำเนิด ข้อมูลจะต้องมีความถูกต้อง มีการกลั่นกรองและตรวจสอบแล้วเท่านั้นจึงจะมีประโยชน์

1.4 บุคลากร หรือพีเพิลแวร์ (People ware)

บุคลากรเป็นองค์ประกอบสำคัญในความสำเร็จของระบบสารสนเทศ บุคลากรมีความรู้ความสามารถทางคอมพิวเตอร์มากเท่าใด โอกาสที่จะใช้งานระบบสารสนเทศและระบบคอมพิวเตอร์ได้เต็มศักยภาพและคุ้มค่ายิ่งมากขึ้นเท่านั้น บุคลากรที่เกี่ยวข้องกับคอมพิวเตอร์มีหลายประเภท ซึ่งแต่ละประเภทมีหน้าที่และความรับผิดชอบแตกต่างกันไปดังนี้

1. **ผู้ใช้งานคอมพิวเตอร์ (User)** หมายถึง ผู้ใช้งานคอมพิวเตอร์ทั่วไป สามารถทำงานตามหน้าที่ในหน่วยงานนั้น ๆ เช่น พิมพ์งาน ป้อนข้อมูลเข้าเครื่องคอมพิวเตอร์ การส่งจดหมายอิเล็กทรอนิกส์ เป็นต้น
2. **ผู้ดูแลและซ่อมบำรุงเครื่องคอมพิวเตอร์ (Supporter or Technician)** หมายถึง ผู้คอยดูแล ตรวจสอบสภาพของเครื่องคอมพิวเตอร์เพื่อให้มีสภาพพร้อมที่จะทำงานได้ตลอดเวลา กลุ่มนี้จะเรียนรู้เทคนิค การดูแล รักษา การซ่อมแซม การต่อเชื่อม ตลอดจนการใช้งานโปรแกรมต่าง ๆ ค่อนข้างดี
3. **ผู้เขียนโปรแกรมคอมพิวเตอร์ (Programmer :โปรแกรมเมอร์)** หมายถึง ผู้เขียนโปรแกรมตามที่ผู้ออกแบบและผู้วิเคราะห์ระบบคอมพิวเตอร์เป็นผู้กำหนด เพื่อให้ได้โปรแกรมที่ตรงตามวัตถุประสงค์การใช้งานในองค์กร กลุ่มนี้จะศึกษามาทางด้านภาษาคอมพิวเตอร์โดยเฉพาะ สามารถเขียนคำสั่งคอมพิวเตอร์โดยภาษาต่างๆ ได้ และเป็นนักพัฒนาโปรแกรมให้คนอื่นเอาไปใช้
4. **ผู้ออกแบบและวิเคราะห์ระบบคอมพิวเตอร์ (System Analysis)** หมายถึง ผู้ที่มีหน้าที่พิจารณาว่า องค์กรควรจะใช้คอมพิวเตอร์ในลักษณะใดจึงจะเหมาะสม เกิดประโยชน์สูงสุด เป็นผู้ออกแบบโปรแกรมก่อนส่งงานไปให้โปรแกรมเมอร์ทำงานในส่วนต่อไป
5. **ผู้บริหารระบบคอมพิวเตอร์ (System Manager or Administrator)** คือ ผู้ที่มีหน้าที่บริหาร และดูแลทรัพยากรทุกชนิดที่เกี่ยวข้องกับคอมพิวเตอร์ให้เกิดประโยชน์สูงสุด

1.5 กระบวนการทำงาน (Documentation/Procedure)

เป็นขั้นตอนการทำงานเพื่อให้ได้ ผลลัพธ์หรือข้อสนเทศจากคอมพิวเตอร์ ในการทำงานกับคอมพิวเตอร์จำเป็นที่จะต้องให้ผู้ใช้งานเข้าใจขั้นตอนการทำงาน ต้องมีระเบียบปฏิบัติให้เป็นแบบเดียวกัน

2. ภาษาคอมพิวเตอร์ (Computer Programming Language)

ภาษาเป็นตัวแทนการสื่อสารระหว่าง 2 สิ่งหรือหลายๆสิ่งเพื่อให้เกิดความหมายและความเข้าใจตรงกัน เช่น มนุษย์ใช้คำพูดสื่อสารกันก็ถือว่าคำพูดนั้นเป็นภาษาหรืออาจใช้มือในการสื่อสารระหว่างคนที่งูเป็นไขพูดไม่ได้หรือแม้แต่ดนตรีก็ถือว่าเป็นภาษาชนิดหนึ่งที่เป็นสากลเพราะคนชาติใดมาฟังก็จะให้ความรู้สึกเดียวกัน

ในทางคอมพิวเตอร์นั้นก็ต้องทำการพัฒนาภาษาที่จะสื่อสารระหว่างมนุษย์กับเครื่องคอมพิวเตอร์แต่ด้วยเหตุที่เครื่องคอมพิวเตอร์เป็นอุปกรณ์อิเล็กทรอนิกส์ซึ่งมีเฉพาะวงจรการเปิดและปิดทำให้เครื่องคอมพิวเตอร์สื่อสารโดยใช้เลขฐานสองเท่านั้นเรียกภาษาที่ใช้เฉพาะเลขฐานสองในคอมพิวเตอร์ว่าภาษาเครื่อง (Machine Language) การที่มนุษย์จะเรียนรู้ภาษาเครื่องนั้นยากมากเพราะนอกจากจะต้องศึกษาถึงอุปกรณ์นั้นอีกด้วยซึ่งจะทำให้การใช้งานคอมพิวเตอร์เป็นเรื่องยุ่งยากจึงมีผู้คิดค้นภาษาคอมพิวเตอร์ขึ้นเพื่อทำหน้าที่ในการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์กับมนุษย์โดยผู้ใช้จะสามารถติดต่อกับเครื่องคอมพิวเตอร์โดยผ่านทางภาษาคอมพิวเตอร์ (Computer Programming Language)

2.1 ชนิดของภาษาคอมพิวเตอร์

จากการที่มีภาษาจำนวนมากมายนั้นทำให้ต้องกำหนดระดับของภาษาคอมพิวเตอร์เพื่อช่วยในการแบ่งประเภทของภาษาเหล่านั้นการกำหนดว่าเป็นภาษาระดับต่ำหรือภาษาระดับสูงจะขึ้นอยู่กับภาษานั้นใกล้เคียงกับภาษาเครื่องคอมพิวเตอร์ (ใกล้เคียงกับรหัส 0 และ 1 เรียกว่า ภาษาระดับต่ำ) หรือว่าใกล้เคียงกับภาษาที่มนุษย์ใช้ (ใกล้เคียงกับภาษาอังกฤษเรียกว่า ภาษาระดับสูง)

1. ภาษาระดับต่ำ (Low Level Language) ได้แก่

1.1 ภาษาเครื่อง (Machine Language) เป็นภาษาระดับต่ำที่สุดเพราะใช้เลขฐานสองแทนข้อมูล (0 และ 1) และคำสั่งต่างๆทำให้การเขียนโปรแกรมยุ่งยากมาก

ตัวอย่าง แสดงคำสั่งของภาษาเครื่องมีดังนี้

ถ้าเราต้องการสั่งให้เครื่องทำงานตามคำสั่ง $9 + 3$ แสดงได้ดังนี้

การบวกแทนด้วยรหัส 10101010

เลข 9 เปลี่ยนเป็นเลขฐานสอง 00001001

เลข 3 เปลี่ยนเป็นเลขฐานสอง 00000011

ดังนั้น คำสั่ง $9 + 3$ เขียนเป็นภาษาเครื่องได้ดังนี้

00001001 10101010 00000011 -----> ภาษาเครื่อง

9 + 3 -----> ภาษามนุษย์และภาษาคอมพิวเตอร์

1.2 ภาษาแอสเซมบลี (Assembly Language) ภาษาแอสเซมบลีใช้รหัสเป็นคำแทนคำสั่งภาษาเครื่องทำให้นักเขียนโปรแกรมสามารถเขียนโปรแกรมได้ง่ายขึ้นคือใช้สัญลักษณ์แทนเลข 0 และ 1 ของภาษาเครื่องซึ่งสัญลักษณ์ที่ใช้จะเป็นคำสั่งสั้นๆ ทำให้การเขียนโปรแกรมง่ายขึ้นกว่าภาษาเครื่อง แต่ก็ยังคงยุ่งยากมากในการจำคำสั่งทั้งหมด

ตัวอย่างที่ แสดงคำสั่งของภาษาแอสเซมบลีดังนี้

ถ้าเราต้องการสั่งให้เครื่องทำงานตามคำสั่ง $9 + 3$ แสดงได้ดังนี้

```
MOV AX, 9
```

```
MOV BX, 3
```

```
ADD AX, BX
```

2. ภาษาระดับสูง (High Level Language)

ภาษาระดับสูงจะใช้คำในภาษาอังกฤษแทนคำสั่งต่างๆรวมทั้งสามารถใช้นิพจน์ทางคณิตศาสตร์ได้ด้วย ทำให้นักเขียนโปรแกรมสามารถใช้เวลามุ่งไปในการศึกษาถึงทางแก้ปัญหาเท่านั้นไม่ต้องเป็นกังวลว่าคอมพิวเตอร์จะทำงานอย่างไรอีกต่อไปภาษาระดับสูงนี้ถือว่าเป็นภาษายุคที่สาม (third-generation language) ซึ่งทำให้เกิดการประมวลผลข้อมูลเพิ่มมากขึ้นและมีผู้หันมาใช้คอมพิวเตอร์กันมากขึ้น

3. ภาษาระดับสูงมาก (Very high-level Language)

เป็นภาษายุคที่ 4 (fourth-generation language) หรือ 4GLs จะเป็นภาษาที่ใช้เขียนโปรแกรมได้สั้นกว่าภาษาในยุคก่อนๆการทำงานบางอย่างสามารถใช้เพียง 5 ถึง 10 บรรทัดเท่านั้นในขณะที่ถ้าเขียนด้วยภาษาอาจต้องใช้ถึง 100 บรรทัดโดยพื้นฐานแล้วภาษาในยุคที่ 4 นี้มีคุณสมบัติที่แยกจากภาษายุคก่อนๆอย่างชัดเจน กล่าวคือผู้เขียนโปรแกรมเพียงแต่กำหนดว่าต้องการให้โปรแกรมทำอะไรบ้างก็สามารถเขียนโปรแกรมได้ทันทีโดยไม่ต้องทราบว่าจะทำได้อย่างไรทำให้การเขียนโปรแกรมสามารถทำได้ง่ายและรวดเร็ว

ภาษาในยุคที่ 4 นี้ยังมีภาษาที่ใช้สำหรับเรียกดูข้อมูลจากฐานข้อมูลได้เรียกว่าภาษาเรียกค้นข้อมูล (query language) การแสดงรายงานจากฐานข้อมูลภาษาเรียกค้นข้อมูลที่เป็นมาตรฐานเรียกว่า SQL (Structured Query Language)

4. ภาษาธรรมชาติ (Nature Language)

เป็นภาษายุคที่ 5 (fifth generation language) หรือ 5GLs ธรรมชาติ หมายถึง ธรรมชาติของมนุษย์คือไม่ต้องสนใจถึงคำสั่งหรือลำดับของข้อมูลที่ต้องการผู้ใช้เพียงแต่พิมพ์สิ่งที่ต้องการลงในเครื่องคอมพิวเตอร์เป็นคำหรือประโยคตามที่ผู้ใช้เข้าใจซึ่งจะทำให้มีรูปแบบของคำสั่งหรือประโยคที่แตกต่างกันออกไปได้มากมายเพราะผู้ใช้แต่ละคนอาจจะใช้ประโยคต่างกันใช้คำศัพท์ต่างกันหรือแม้กระทั่งบางคนอาจจะใช้ศัพท์แสลงก็ได้คอมพิวเตอร์จะพยายามแปลคำหรือประโยคเหล่านั้นตามคำสั่งแต่ถ้าไม่สามารถแปลให้เข้าใจได้ก็จะมีคำถามกลับมาถามผู้ใช้เพื่อยืนยันความถูกต้องภาษาธรรมชาติจะใช้ระบบฐานความรู้ (knowledge base system) ช่วยในการแปลความหมายของคำสั่งต่างๆ

5. ภาษาโปรแกรมเชิงวัตถุ (Object-Oriented Programming Language)

นักเขียนโปรแกรมบางคนคิดว่า การเขียนโปรแกรมขนาดใหญ่บางครั้งก็เป็นงานที่หนักและเสียเวลามากจึงได้พยายามคิดหาวิธีที่จะทำให้การเขียนโปรแกรมนั้นง่ายขึ้นและสามารถเขียนได้อย่างรวดเร็วทำให้เกิดเทคนิคการโปรแกรมเชิงวัตถุ (Object-Oriented Programming) หรือ OOP เพื่อช่วยลดความยุ่งยากของการเขียนโปรแกรมโปรแกรมเชิงวัตถุที่ได้รับความนิยมสูงได้แก่ JAVA , Visual Basic , C++

2.2 การเลือกใช้ภาษาคอมพิวเตอร์

เนื่องจากในปัจจุบันทุกๆปีจะมีภาษาคอมพิวเตอร์เกิดขึ้นมากมายและภาษาต่างๆจะมีจุดดีและจุดด้อยแตกต่างกันไปผู้ใช้จึงจำเป็นต้องทำการคัดเลือกภาษาที่จะนำมาใช้งานอย่างระมัดระวังเนื่องจากเมื่อศึกษาและพัฒนาซอฟต์แวร์ด้วยภาษาใดภาษาหนึ่งแล้วการเปลี่ยนไปใช้ภาษาอื่นในภายหลังจะเป็นเรื่องที่ยากลำบากอย่างยิ่งทั้งในเรื่องของค่าใช้จ่ายและเวลาที่ต้องสูญเสียไปรวมทั้งปัญหาของบุคลากรที่ต้องศึกษาหาความชำนาญใหม่อีกด้วย

ในการเลือกภาษาคอมพิวเตอร์ที่จะนำมาใช้สิ่งที่ควรพิจารณาคือ

1. ในหน่วยงานหนึ่งๆควรจะใช้ภาษาคอมพิวเตอร์ภาษาเดียวกันเพราะการดูแลรักษาซอฟต์แวร์ที่พัฒนาขึ้นตลอดจนการจัดหาบุคลากรจะกระทำให้ง่ายกว่า
2. ในการเลือกภาษาควรเลือกโดยดูจากคุณสมบัติหรือข้อดีของภาษานั้นๆ เป็นหลัก
3. ถ้าโปรแกรมที่เขียนขึ้นนั้นต้องนำไปทำงานบนเครื่องต่างๆกันควรเลือกภาษาที่สามารถใช้งานได้บนทุกเครื่องเพราะจะทำให้เขียนโปรแกรมเพียงครั้งเดียวเท่านั้น
4. ผู้ใช้ควรจำกัดภาษาคอมพิวเตอร์ที่จะใช้ไม่ควรติดตั้งตัวแปลภาษาคอมพิวเตอร์ทุกภาษาบนเครื่องทุกเครื่อง
5. ภาษาคอมพิวเตอร์ที่เลือกใช้จะถูกจำกัดโดยนักเขียนโปรแกรมที่มีอยู่เพราะควรใช้ภาษาที่มีผู้รู้อยู่บ้าง

ตารางแสดงตัวอย่างภาษาคอมพิวเตอร์และการใช้งาน

ภาษาคอมพิวเตอร์	การใช้งาน
BASIC	เพื่อใช้สอนวิชาการเขียนโปรแกรมสำหรับผู้เริ่มศึกษาการเขียนโปรแกรมภาษาคอมพิวเตอร์
COBOL	เป็นภาษาสำหรับใช้ในงานธุรกิจบนเครื่องขนาดใหญ่
FORTRAN	เป็นภาษาระดับสูงภาษาแรกของโลกใช้สำหรับงานด้านคณิตศาสตร์วิทยาศาสตร์วิศวกรรมศาสตร์ที่ต้องการการคำนวณมากๆ
Pascal	เพื่อใช้สอนวิชาการเขียนโปรแกรมสำหรับผู้เริ่มศึกษาการเขียนโปรแกรมภาษาคอมพิวเตอร์
C	เป็นภาษาระดับสูง ใช้สอนวิชาการเขียนโปรแกรมสำหรับผู้เริ่มศึกษาการเขียนโปรแกรมภาษาคอมพิวเตอร์
C++	สำหรับผู้ผลิตซอฟต์แวร์แบบ OOP
ALGOL	เริ่มต้นได้รับการออกแบบให้เป็นภาษาสำหรับงานทางวิทยาศาสตร์และต่อมามีการพัฒนาต่อเป็นภาษา PL/I และ Pascal
APL	ออกแบบโดยบริษัท IBM เป็นภาษาที่โต้ตอบกับผู้ใช้ทันทีเหมาะสำหรับจัดการกับกลุ่มของข้อมูลที่สัมพันธ์กันในรูปแบบตาราง
LISP	ถูกออกแบบมาให้ใช้กับข้อมูลที่ไม่ใช้ตัวเลขซึ่งอาจเป็นสัญลักษณ์พิเศษหรือตัวอักษรก็ได้นิยมใช้ในด้านปัญญาประดิษฐ์ (Artificial Intelligence)
LOGO	นิยมใช้ในโรงเรียนเพื่อสอนทักษะการแก้ปัญหาให้กับนักเรียน
PL/I	ถูกออกแบบมาให้ใช้กับงานทั้งทางด้านวิทยาศาสตร์และธุรกิจ
PROLOG	นิยมใช้มากในงานด้านปัญญาประดิษฐ์จัดเป็นภาษาธรรมชาติภาษาหนึ่ง
RPG	ถูกออกแบบมาให้ใช้กับงานทางธุรกิจจะมีคุณสมบัติในการสร้างโปรแกรมสำหรับพิมพ์รายงาน

2.3 ตัวแปลภาษา (Translator)

ในการพัฒนาซอฟต์แวร์คอมพิวเตอร์นั้นโปรแกรมเมอร์จะเขียนโปรแกรมในภาษาคอมพิวเตอร์แบบต่างๆ ตามแต่ความชำนาญของแต่ละคนโปรแกรมที่ได้จะเรียกว่าโปรแกรมต้นฉบับหรือซอร์สโปรแกรม (source program) ซึ่งมนุษย์จะอ่านโปรแกรมต้นฉบับนี้ได้แต่คอมพิวเตอร์จะไม่เข้าใจคำสั่งเหล่านั้นเนื่องจากคอมพิวเตอร์เข้าใจแต่ภาษาเครื่อง (Machine Language) ซึ่งประกอบขึ้นจากรหัสฐานสองเท่านั้นจึงต้องมีการใช้โปรแกรมตัวแปลภาษาคอมพิวเตอร์ (Translator) ในการแปลภาษาคอมพิวเตอร์ภาษาต่างๆไปเป็นภาษาเครื่องโปรแกรมที่แปลจากโปรแกรมต้นฉบับแล้ว เรียกว่า ออบเจคโปรแกรม (object program) ซึ่งจะประกอบด้วยรหัสคำสั่งที่คอมพิวเตอร์สามารถเข้าใจและนำไปปฏิบัติได้ต่อไป

ตัวแปลภาษาที่มีการใช้ในปัจจุบันสามารถแบ่งได้เป็น 3 ตัวดังนี้

1. แอสเซมเบลเลอร์ (Assembler)

เป็นตัวแปลภาษาแอสเซมบลีซึ่งเป็นภาษาระดับต่ำให้เป็นภาษาเครื่อง

2. คอมไพเลอร์ (Compiler)

จะทำการแปลโปรแกรมทั้งโปรแกรมให้เป็นภาษาเครื่องที่เดียวการแปลนี้จะเป็นการตรวจสอบไวยากรณ์ของภาษาถ้ามีข้อผิดพลาดทางไวยากรณ์ของภาษาเกิดขึ้นก็จะแจ้งให้ทราบ

3. อินเทอร์พรีเตอร์ (Interpreter)

จะทำการแปลโปรแกรมภาษาชั้นสูงทีละคำสั่ง ทีละบรรทัดระหว่างการแปลเกิดพบข้อผิดพลาดที่บรรทัดใดก็จะฟ้องให้ทำการแก้ไขที่บรรทัดนั้นทันที

ตัวแปลภาษาทุกตัวจะทำการแปลภาษาคอมพิวเตอร์ให้เป็นภาษาเครื่อง (ตัวเลข 0 กับ 1) เพื่อให้เครื่องคอมพิวเตอร์เข้าใจและทำงานตามคำสั่งได้

3. การพัฒนาโปรแกรม

ขั้นตอนหรือวิธีการพัฒนาโปรแกรมประกอบด้วย 6 ขั้นตอนดังนี้

1. การวิเคราะห์ปัญหา(Problem Analysis) การวิเคราะห์ปัญหา ประกอบด้วยขั้นตอนต่างๆ ดังนี้

1. กำหนดวัตถุประสงค์ของงาน เพื่อพิจารณาว่าโปรแกรมต้องทำการประมวลผลอะไรบ้าง
2. พิจารณาข้อมูลนำเข้า เพื่อให้ทราบว่าต้องนำข้อมูลอะไรเข้าคอมพิวเตอร์ ข้อมูลมีคุณสมบัติเป็นอย่างไร ตลอดจนถึงลักษณะและรูปแบบของข้อมูลที่จะนำเข้า
- 3 พิจารณาการประมวลผล เพื่อให้ทราบว่าโปรแกรมมีขั้นตอนการประมวลผลอย่างไรและมีเงื่อนไขการประมวลผลอะไรบ้าง
4. พิจารณาข้อมูลนำออก เพื่อให้ทราบว่าข้อมูลอะไรที่จะแสดง ตลอดจนรูปแบบและสื่อที่จะใช้ในการแสดงผล

2. การออกแบบโปรแกรม(Design)

การออกแบบขั้นตอนการทำงานของโปรแกรมเป็นขั้นตอนที่ใช้เป็นแนวทางในการลงรหัสโปรแกรม อาจใช้เครื่องมือต่างๆ ช่วยในการออกแบบ เช่น คำสั่งจำลอง (Pseudocode) หรือ ผังงาน (Flow chart) การออกแบบโปรแกรมนั้นไม่ต้องพะวงกับรูปแบบคำสั่งภาษาคอมพิวเตอร์ แต่มุ่งความสนใจไปที่ลำดับขั้นตอนในการประมวลผลของโปรแกรมเท่านั้น

3. การเขียนโปรแกรมด้วยภาษาคอมพิวเตอร์(Programming)

การเขียนโปรแกรมเป็นการนำเอาผลลัพธ์ของการออกแบบโปรแกรม มาเปลี่ยนเป็นโปรแกรมภาษาคอมพิวเตอร์ภาษาใดภาษาหนึ่ง ผู้เขียนโปรแกรมจะต้องให้ความสนใจต่อรูปแบบคำสั่งและกฎเกณฑ์ของภาษาที่ใช้เพื่อให้การประมวลผลเป็นไปตามผลลัพธ์ที่ได้ออกแบบไว้

4. การทดสอบและแก้ไขโปรแกรม(Testing)

การทดสอบโปรแกรมเป็นการนำโปรแกรมที่ลงรหัสแล้วเข้าคอมพิวเตอร์ หรือการติดตั้งโปรแกรมเพื่อตรวจสอบการทำงานของโปรแกรมว่าถูกต้องหรือไม่ ถ้าพบว่ายังไม่ถูกต้องก็แก้ไขให้ถูกต้องต่อไป

5. การทำเอกสารประกอบโปรแกรม(Documentation)

การทำเอกสารประกอบโปรแกรมเป็นงานที่สำคัญของการพัฒนาโปรแกรม เอกสารประกอบโปรแกรมช่วยให้ผู้ใช้โปรแกรมเข้าใจวัตถุประสงค์ ข้อมูลที่จะต้องใช้กับโปรแกรม ตลอดจนผลลัพธ์ที่จะได้จากโปรแกรม การทำโปรแกรมทุกโปรแกรมจึงควรต้องทำเอกสาร

6. การบำรุงรักษาโปรแกรม(Maintenance)

ต้องมีผู้คอยควบคุมดูแลและคอยตรวจสอบการทำงาน การบำรุงรักษาโปรแกรมจึงเป็นขั้นตอนที่ผู้เขียนโปรแกรมต้องคอยเฝ้าดูและหาข้อผิดพลาดของโปรแกรมในระหว่างที่ผู้ใช้ใช้งานโปรแกรม และปรับปรุงโปรแกรมเมื่อเกิดข้อผิดพลาดขึ้น

4. การจำลองความคิดในการเขียนโปรแกรม

การจำลองความคิดเพื่อวางแผนขั้นตอนในการเขียนโปรแกรมทางคอมพิวเตอร์เรียกได้อีกอย่างว่า อัลกอริทึม (Algorithm) หรือขั้นตอนวิธี

อัลกอริทึม เป็นลำดับของคำสั่งที่คอมพิวเตอร์จะปฏิบัติตามเพื่อแก้ปัญหาให้กับเราโดยจะทำคำสั่งเรียงกันตามลำดับก่อนหลังจะไม่ข้ามขั้นผลลัพธ์ของแต่ละขั้นตอนผลลัพธ์ที่ได้ของขั้นตอนหนึ่งจะส่งต่อไปยังขั้นตอนถัดไปและส่งต่อกันไปเช่นนี้ตามลำดับขั้นจนถึงคำสั่งสุดท้ายจึงจะได้ผลลัพธ์ที่เสร็จสมบูรณ์

เครื่องมือที่ใช้ในการจำลองความคิดประกอบด้วย 2 ลักษณะคือ

1. ข้อความคำบรรยายหรือรหัสเทียม (pseudo code)

เป็นการเขียนเค้าโครงด้วยการบรรยายเป็นภาษามนุษย์ที่ใช้สื่อสารกันเพื่อให้ทราบถึงขั้นตอนการทำงานของการเขียนโปรแกรมแต่ละตอนในบางครั้งอาจใช้คำสั่งของภาษาที่ใช้เขียนโปรแกรมก็ได้

หลักการทั่วไปในการเขียนรหัสเทียม

1. สัญลักษณ์ที่ใช้ในการดำเนินการทางคณิตศาสตร์ต่างๆจะถูกใช้งานตามปกติคือ “+” สำหรับการบวก “-” สำหรับการลบ “*” สำหรับการคูณและ “/” สำหรับการหาร

2. ชื่อข้อมูลแทนจำนวนที่จะถูกดำเนินการ

3. การกำหนดค่าให้กับชื่อข้อมูล เช่น เมื่อเราต้องการกำหนดให้ข้อมูล pi มีค่าเท่ากับ 3.14 สามารถเขียนได้ด้วยข้อความ pi=3.14

ในการกำหนดค่าทางคอมพิวเตอร์ด้านซ้ายของเครื่องหมายมักใช้แทนที่ *เก็บข้อมูล* และด้านขวาแทน *ข้อมูลที่ต้องการนำไปเก็บ* (ดังนั้นหากใช้ข้อความว่า 3.14=pi ถือว่าไม่ถูกต้องตามความหมายนี้)

4. คำสงวนบางคำที่ใช้ในภาษาระดับสูงทั่วไปอาจถูกนำมาใช้เช่น Read หรือ Enter สำหรับการรับข้อมูลเข้าและ Write หรือ Print สำหรับการแสดงข้อมูลออก

5. การเพิ่มหรือลดระยะย่อหน้าอย่างเหมาะสมเพื่อแสดงระดับของขั้นตอนการทำงานในโครงสร้างควบคุมการทำงานในกลุ่มเดียวกัน

ตัวอย่างที่ 1 การวางแผนไปโรงเรียน

การจำลองความคิดด้วยรหัสเทียม

เริ่มต้น

ตื่นนอน

อาบน้ำแต่งตัว

รับประทานอาหารเช้า

ไปโรงเรียน

จบ



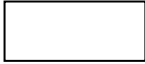
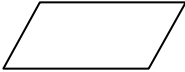




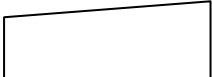
2. สัญลักษณ์หรือ ผังงาน (Flowchart)

เป็นเครื่องมือชนิดหนึ่งที่ใช้รูปภาพแสดงการไหลของข้อมูลในระบบตั้งแต่แรกจนได้ผลลัพธ์ตามต้องการจะแสดงการทำงานของโปรแกรมโดยละเอียดในแต่ละขั้นตอนผังงานโปรแกรมเป็นสิ่งจำเป็นสำหรับผู้เขียนโปรแกรมเพราะต้องใช้เป็นแนวทางในการเขียนโปรแกรมและเมื่อโปรแกรมเกิดข้อผิดพลาดการเข้าไปวิเคราะห์ผังงานโปรแกรมจะทำได้ง่ายกว่าการเข้าไปวิเคราะห์ตัวโปรแกรมโดยตรง

ประโยชน์ของผังงาน

1. ช่วยอธิบายลำดับขั้นตอนการทำงานของโปรแกรม
2. ทำให้ตรวจสอบข้อผิดพลาดของโปรแกรมได้ง่าย
3. ทำให้ผู้อื่นสามารถศึกษาการทำงานของโปรแกรมและแก้ไขโปรแกรมได้ง่าย

สัญลักษณ์ที่ใช้ในการเขียนผังงานและหน้าที่

	หมายถึง	เริ่มต้น/จบงาน
	หมายถึง	การตัดสินใจเลือก
	หมายถึง	แทนการกำหนดค่าหรือคำนวณค่า
	หมายถึง	อ่านข้อมูลเข้าหรือแสดงข้อมูลออก
	หมายถึง	แสดงผลลัพธ์ทางเครื่องพิมพ์
	หมายถึง	แสดงผลลัพธ์บนจอภาพ
	หมายถึง	จุดเชื่อมต่อ
	หมายถึง	ลูกศรแสดงทิศทาง
	หมายถึง	การป้อนข้อมูลเข้าด้วยมือ

ทำความรู้จักกับสัญลักษณ์สำคัญที่ใช้บ่อย

1. สัญลักษณ์การคำนวณ Process

ความหมายของ Process คือ การกำหนดค่าหรือการคำนวณค่า



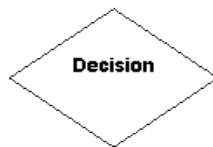
ตัวอย่าง แสดงการกำหนดค่าให้ตัวแปร Total



ตัวอย่าง แสดงการคำนวณค่าผลรวมของตัวแปร Total

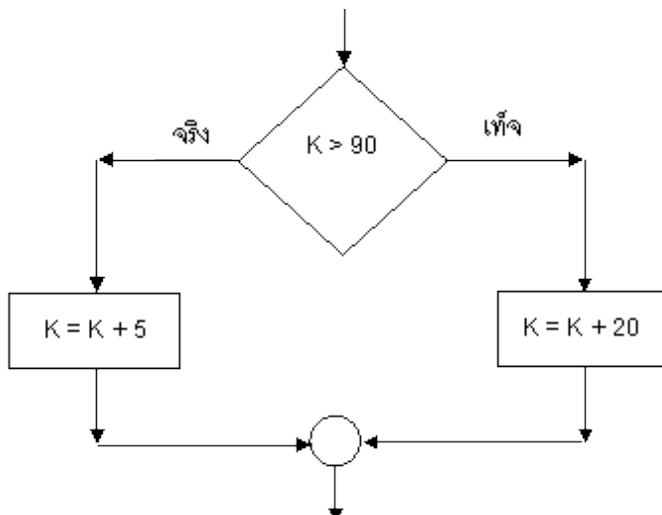


2. สัญลักษณ์การตัดสินใจ Decision



ความหมายของ Decision คือ การทดสอบเงื่อนไขหรือเปรียบเทียบเงื่อนไข เพื่อตัดสินใจทำงานอย่างใดอย่างหนึ่งตามที่ต้องการ

ตัวอย่าง แสดงการเปรียบเทียบเงื่อนไข 1 ทางเลือก



**เปรียบเทียบเงื่อนไข $K > 90$
จริงหรือไม่**

- ถ้าจริง (true) ให้คำนวณ $K = K + 5$ แล้วออกจากเงื่อนไข
- ถ้าไม่จริง (false) ให้คำนวณ $K = K + 20$ แล้วออกจากเงื่อนไข

ตัวอย่างแสดงการจำลองความคิดแบบรหัสเทียม (pseudo code)และเป็นสัญลักษณ์ (Flowchart)

ตัวอย่างที่ 1 การเดินทางไปโรงเรียน

รหัสเทียม (pseudo code)

เริ่มต้น

ตื่นนอน

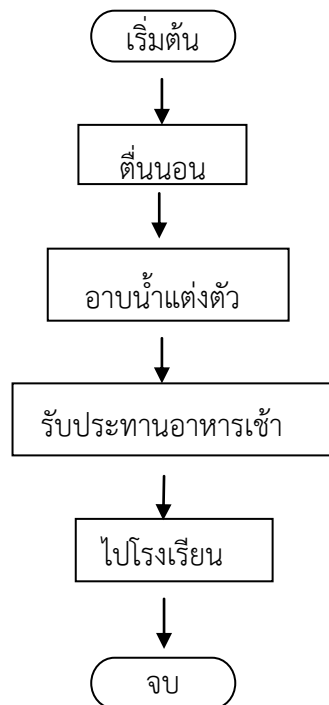
อาบน้ำแต่งตัว

รับประทานอาหารเช้า

ไปโรงเรียน

จบ

สัญลักษณ์ (Flowchart)



ตัวอย่างที่ 2 การคำนวณพื้นที่รูปวงกลม

การจำลองความคิดด้วยรหัสเทียม

เริ่มต้น

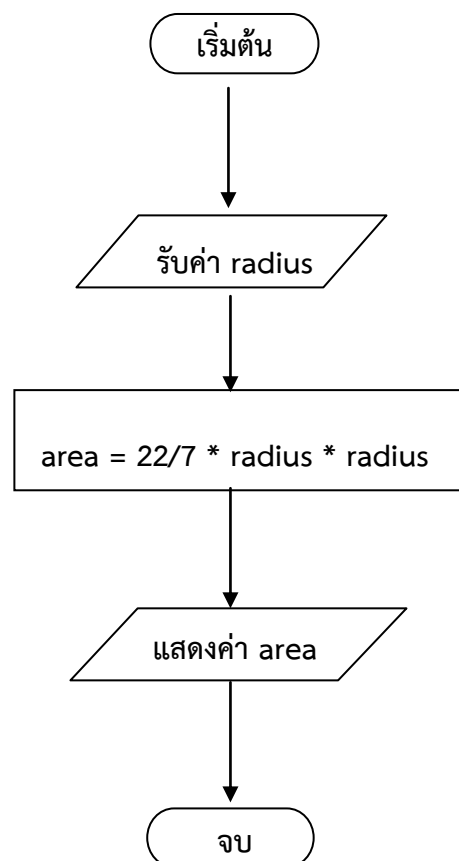
รับค่า radius

คำนวณพื้นที่ด้วยสูตร $area = 22/7 * radius * radius$

แสดงค่า area

จบ

การจำลองความคิดเป็นสัญลักษณ์



ตัวอย่างที่ 3 การจำลองความคิดในการหาผลบวก 1,2,3,4,5,... จนถึง 20
(นั่นคือจะหาค่า $1+2+3+4+5+\dots+20$)

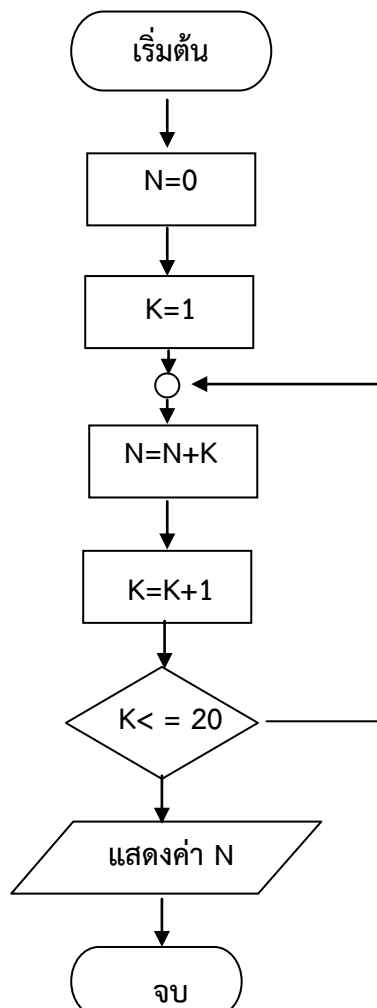
การจำลองความคิดด้วยรหัสเทียม

เริ่มต้น

1. กำหนดให้ N มีค่าเริ่มต้นเป็น 0
2. กำหนดให้ K มีค่าเริ่มต้นเป็น 1
3. นำค่า K มารวมกับค่า N เดิมได้ผลลัพธ์เท่าไรเก็บไว้ที่ N
4. นำค่า 1 มารวมกับค่า K เดิมได้ผลลัพธ์เท่าไรเก็บไว้ที่ K
5. เปรียบเทียบค่า K กับ 20 ถ้า K น้อยกว่าหรือเท่ากับ 20 ให้วนกลับไปทำในขั้น 3 และทำคำสั่งถัดลงมาตามลำดับ แต่ถ้า K มากกว่า 20 ให้แสดงคำตอบ

จบ

การจำลองความคิดเป็นสัญลักษณ์



ตัวอย่างที่ 4 การสร้างผังงานแสดงการคำนวณภาษี

การจำลองความคิดด้วยรหัสเทียม

เริ่มต้น

รับค่า ราคาสินค้า

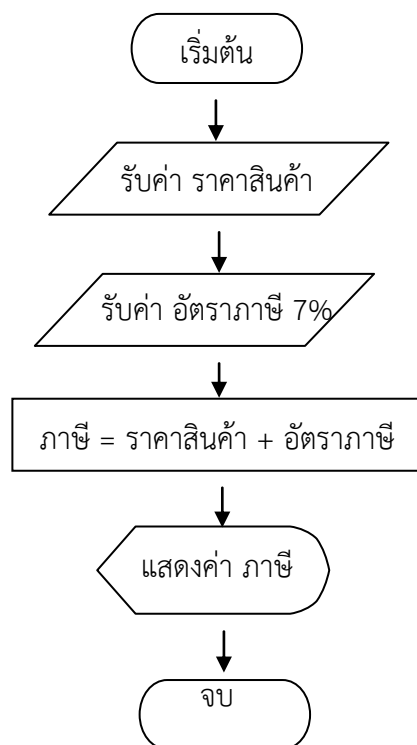
รับค่า อัตราภาษี ร้อยละ 7

คำนวณภาษีที่ต้องจ่าย โดยใช้สูตร $\text{ภาษี} = \text{ราคาสินค้า} + \text{อัตราภาษี}$

แสดงค่าภาษี

จบ

การจำลองความคิดเป็นสัญลักษณ์



กิจกรรมที่ 1.1 องค์ประกอบของระบบคอมพิวเตอร์

คำชี้แจง ให้นักเรียนตอบคำถามต่อไปนี้ให้ถูกต้องสมบูรณ์ที่สุด

1. องค์ประกอบของระบบคอมพิวเตอร์ประกอบด้วยกี่ส่วน อะไรบ้าง

.....
.....
.....
.....
.....

2. ยกตัวอย่างซอฟต์แวร์ระบบปฏิบัติการ (OS) ที่นักเรียนรู้จักมาอย่างน้อย 5 โปรแกรม

.....
.....
.....
.....
.....

3. ยกตัวอย่างซอฟต์แวร์ประยุกต์หรือโปรแกรมสำเร็จรูปที่นักเรียนเคยใช้และชอบมาอย่างน้อย 5 โปรแกรม

.....
.....
.....
.....
.....

4. ผู้เขียนโปรแกรมหรือผู้พัฒนาโปรแกรมมีชื่อเรียกอีกอย่างหนึ่งว่าอะไร

.....
.....

5. จากองค์ประกอบระบบของคอมพิวเตอร์ทั้งหมด ตัวนักเรียนเองจัดอยู่ในองค์ประกอบข้อใด

.....
.....

ชื่อ-นามสกุล.....ห้อง.....เลขที่.....

กิจกรรมที่ 1.2 ภาษาคอมพิวเตอร์

คำชี้แจง ให้นักเรียนตอบคำถามต่อไปนี้ให้ถูกต้องสมบูรณ์ที่สุด

1.บอกความหมายของ ภาษาคอมพิวเตอร์

.....
.....
.....
.....
.....

2. ภาษาเครื่องใช้เลขฐานสองแทนข้อมูล ตัวเลขฐานสองประกอบด้วยตัวเลขจำนวนกี่ตัว อะไรบ้าง

.....
.....
.....

3.ภาษา C จัดอยู่ในภาษาระดับใด

.....
.....
.....

4. ตัวแปลภาษาทำหน้าที่อะไร จงอธิบาย

.....
.....
.....

5.คอมไพเลอร์(Complier) ทำงานอย่างไร

.....
.....
.....
.....

ชื่อ-นามสกุล.....ห้อง.....เลขที่.....

กิจกรรมที่ 1.3 ความรู้เบื้องต้นเกี่ยวกับการพัฒนาโปรแกรม

คำชี้แจง ให้นักเรียนตอบคำถามต่อไปนี้ให้ถูกต้องสมบูรณ์ที่สุด

1. ขั้นตอนการพัฒนาโปรแกรมประกอบด้วยกี่ขั้นตอน อะไรบ้าง

.....
.....
.....
.....
.....
.....

2. ขั้นตอนการวิเคราะห์ปัญหา ต้องการทราบข้อมูลอะไรบ้าง

.....
.....

3. การเขียนผังงาน หรือ Flowchart จัดอยู่ในขั้นตอนใด

.....
.....

4. เครื่องหมายที่ใช้แทนการคูณ การหาร ในการพัฒนาโปรแกรมใช้เครื่องหมายใด

.....
.....

5. การจำลองความคิดมีกี่แบบ อะไรบ้าง

.....
.....

6. จงบอกประโยชน์ของผังงานมาพอเข้าใจ

.....
.....

ชื่อ-นามสกุล.....ห้อง.....เลขที่.....

กิจกรรมที่ 1.4 ขั้นตอนการพัฒนาโปรแกรม

คำชี้แจง ให้นักเรียนนำตัวอักษรที่อยู่หน้าข้อความด้านขวามือมาใส่หน้าข้อความด้านซ้ายมือให้สอดคล้องกัน

- | | |
|--|---|
|1.การเขียนผังงาน (Flowchart) | 1. การวิเคราะห์ปัญหา (Problem Analysis) |
|2.ขั้นตอนที่ยากที่สุด | 2. การออกแบบโปรแกรม (Design) |
|3.ต้องการให้ทำอะไร | 3. การเขียนโปรแกรมด้วยภาษาคอมพิวเตอร์ (Programming) |
|4.ต้องการผลลัพธ์อะไร | 4. การทดสอบและแก้ไขโปรแกรม (Testing) |
|5.ข้อมูลพื้นฐานที่จำเป็น | 5. การทำเอกสารประกอบโปรแกรม (Documentation) |
|6.ขั้นตอนที่ผิดพลาดไม่ได้ | 6. การบำรุงรักษาโปรแกรม (Maintenance) |
|7.เหมาะสำหรับโปรแกรมเมอร์ | |
|8.ต้องมีพื้นฐานด้านภาษาโปรแกรม | |
|9.เลือกเครื่องมือที่จะนำมาแก้ปัญหา | |
|10.ตรวจสอบว่าวิธีการนี้ให้ผลลัพธ์ที่ต้องการ | |
|11.ผู้ใช้โปรแกรมไม่เข้าใจการใช้งาน | |
|12.จัดทำคู่มือการใช้โปรแกรม | |

ชื่อ-นามสกุล..... ห้อง..... เลขที่.....

กิจกรรมที่ 1.5 การจำลองความคิดแบบคำบรรยายหรือรหัสเทียม

คำชี้แจง จงเรียงลำดับขั้นตอนที่กำหนดให้ถูกต้อง

1.

ลำดับขั้นตอนที่กำหนด	การเรียงลำดับขั้นตอนที่ถูกต้อง
ไปห้องวิชาการ	
เริ่มต้น	
ถ่ายรูปทำบัตร	
จบ	
เซ็นชื่อรับบัตร	
เขียนใบคำร้อง	
พิมพ์บัตร	

2.

ลำดับขั้นตอนที่กำหนด	การเรียงลำดับขั้นตอนที่ถูกต้อง
เริ่มต้น	
อ่านหนังสือ	
สอบปลายภาค	
จบ	
คะแนนมากกว่า 50 หรือไม่	
พิมพ์สอบผ่าน	
พิมพ์สอบตก	

3.

ลำดับชั้นตอนที่กำหนด	การเรียงลำดับชั้นตอนที่ถูกต้อง
คะแนน 70-79 ได้เกรด 3	
รับข้อมูลคะแนน	
คะแนน 60-69 ได้เกรด 2	
ตรวจสอบคะแนน	
คะแนน 80 ขึ้นไปได้เกรด 4	
จบ	
คะแนน 50-59 ได้เกรด 1	
คะแนนน้อยกว่า 50 ได้เกรด 0	
พิมพ์ระดับผลการเรียน	
เริ่มต้น	

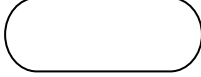

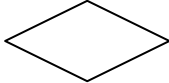


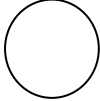



4.

ลำดับชั้นตอนที่กำหนด	การเรียงลำดับชั้นตอนที่ถูกต้อง
ลงทะเบียนขอสอบแก้ตัว	
สอบไม่ผ่าน	
ประกาศผลสอบปลายภาค	
ประกาศผลสอบแก้ตัว	
สอบแก้ตัว	
เริ่มต้น	
จบ	

กิจกรรมที่ 1.6 การเขียนผังงาน (Flowchart)

คำชี้แจง ให้นักเรียนบอกความหมายของสัญลักษณ์ผังงานให้ถูกต้อง

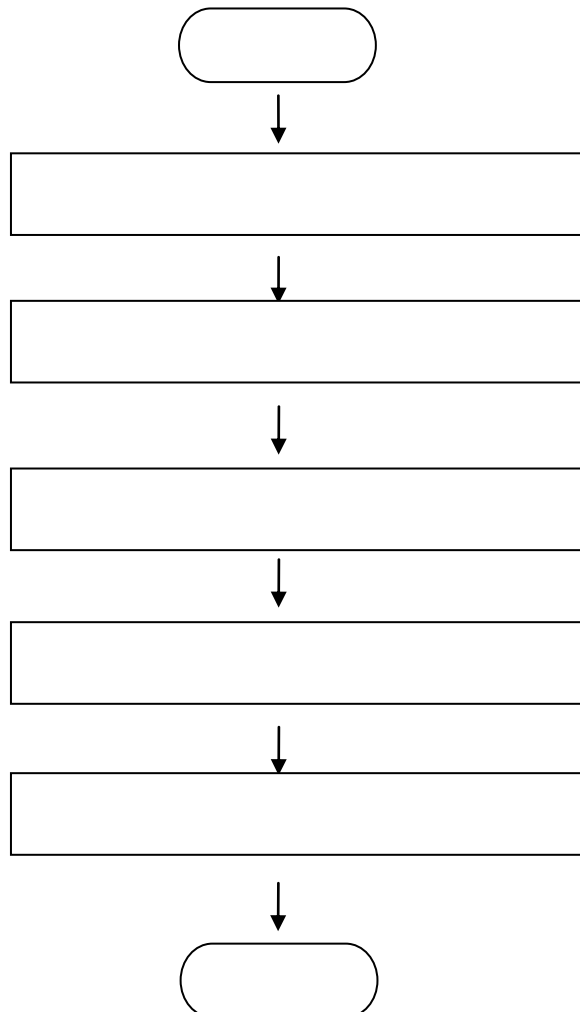
1. สัญลักษณ์ต่อไปนี้ หมายถึง

สัญลักษณ์	ความหมาย
	
	
	
	
	
	
	
	
	

2. จงเรียงลำดับขั้นตอนที่กำหนดให้ถูกต้อง

ลำดับขั้นตอนที่กำหนด	การเรียงลำดับขั้นตอนที่ถูกต้อง
ไปห้องวิชาการ	เริ่มต้น
เริ่มต้น	ไปห้องวิชาการ
ถ่ายรูปทำบัตร	เขียนใบคำร้อง
จบ	ถ่ายรูปทำบัตร
เซ็นชื่อรับบัตร	พิมพ์บัตร
เขียนใบคำร้อง	เซ็นชื่อรับบัตร
พิมพ์บัตร	จบ

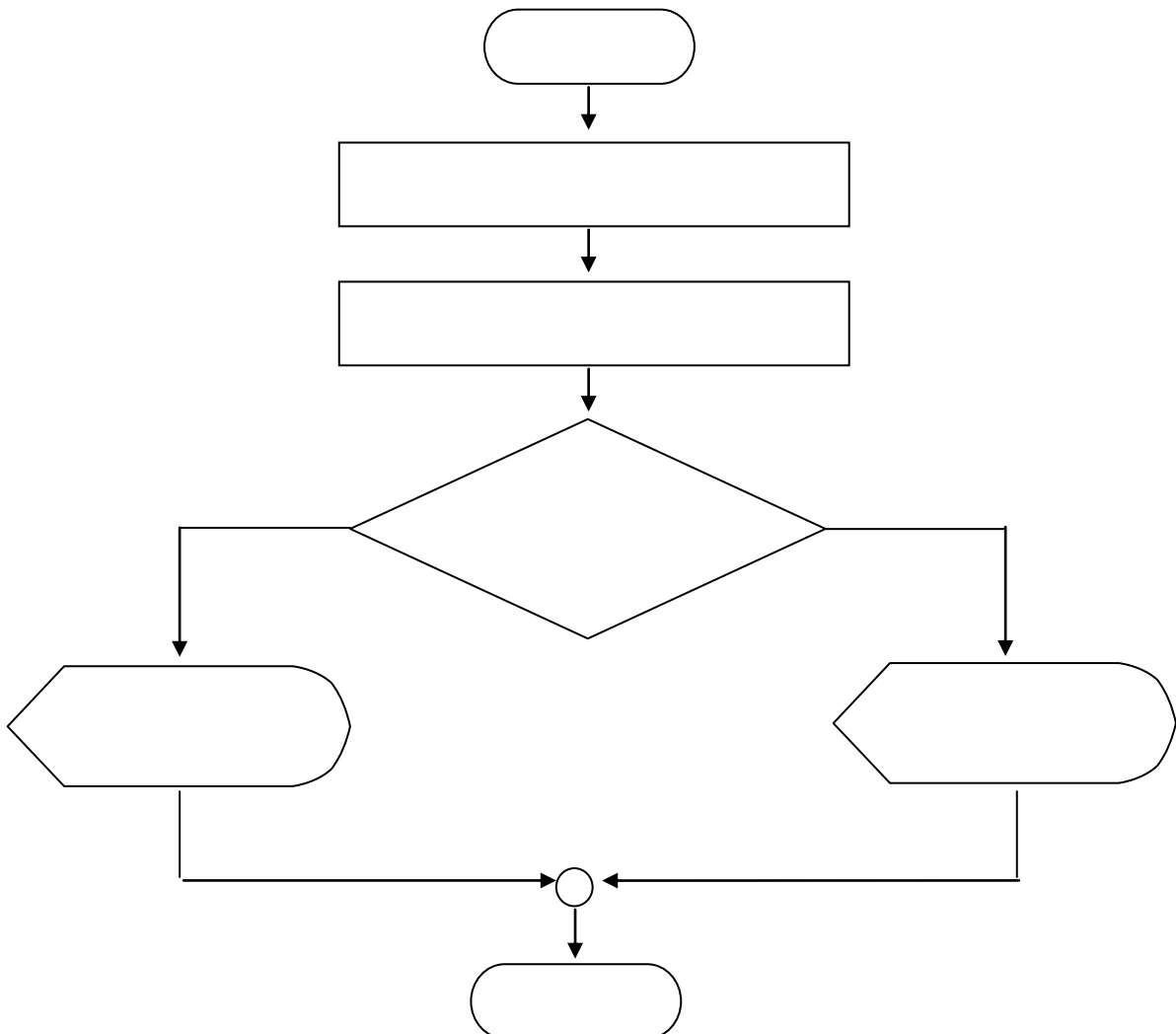
นำข้อความที่เรียงสมบูรณ์แล้วมาเขียนลงในผังงานให้ถูกต้อง



3. จงเรียงลำดับขั้นตอนที่กำหนดให้ ให้ถูกต้อง

ลำดับขั้นตอนที่กำหนด	การเรียงลำดับขั้นตอนที่ถูกต้อง
เริ่มต้น	เริ่มต้น
อ่านหนังสือ	อ่านหนังสือ
สอบปลายภาค	สอบปลายภาค
จบ	คะแนนมากกว่า 50 หรือไม่
คะแนนมากกว่า 50 หรือไม่	พิมพ์สอบผ่าน
พิมพ์สอบผ่าน	พิมพ์สอบตก
พิมพ์สอบตก	จบ

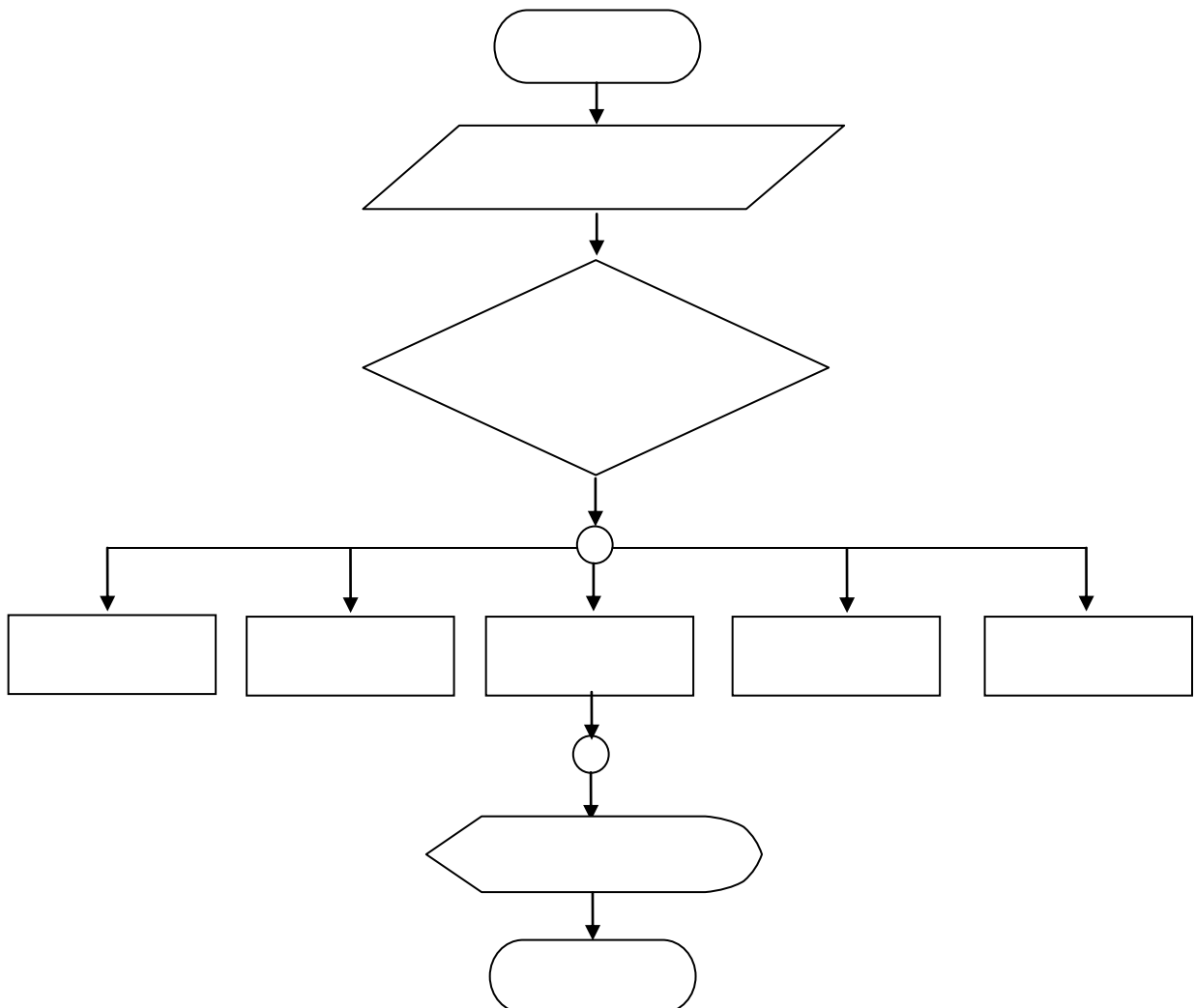
นำข้อความที่เรียงสมบูรณ์แล้วมาเขียนลงในแผนผังความคิดให้ถูกต้อง



4. จงเรียงลำดับขั้นตอนที่กำหนดให้ถูกต้อง

ลำดับขั้นตอนที่กำหนด	การเรียงลำดับขั้นตอนที่ถูกต้อง
คะแนน 70-79 ได้เกรด 3	เริ่มต้น
รับข้อมูลคะแนน	รับข้อมูลคะแนน
คะแนน 60-69 ได้เกรด 2	ตรวจสอบคะแนน
ตรวจสอบคะแนน	คะแนน 80 ขึ้นไปได้เกรด 4
คะแนน 80 ขึ้นไปได้เกรด 4	คะแนน 70-79 ได้เกรด 3
จบ	คะแนน 60-69 ได้เกรด 2
คะแนน 50-59 ได้เกรด 1	คะแนน 50-59 ได้เกรด 1
คะแนนน้อยกว่า 50 ได้เกรด 0	คะแนนน้อยกว่า 50 ได้เกรด 0
พิมพ์ระดับผลการเรียน	พิมพ์ระดับผลการเรียน
เริ่มต้น	จบ

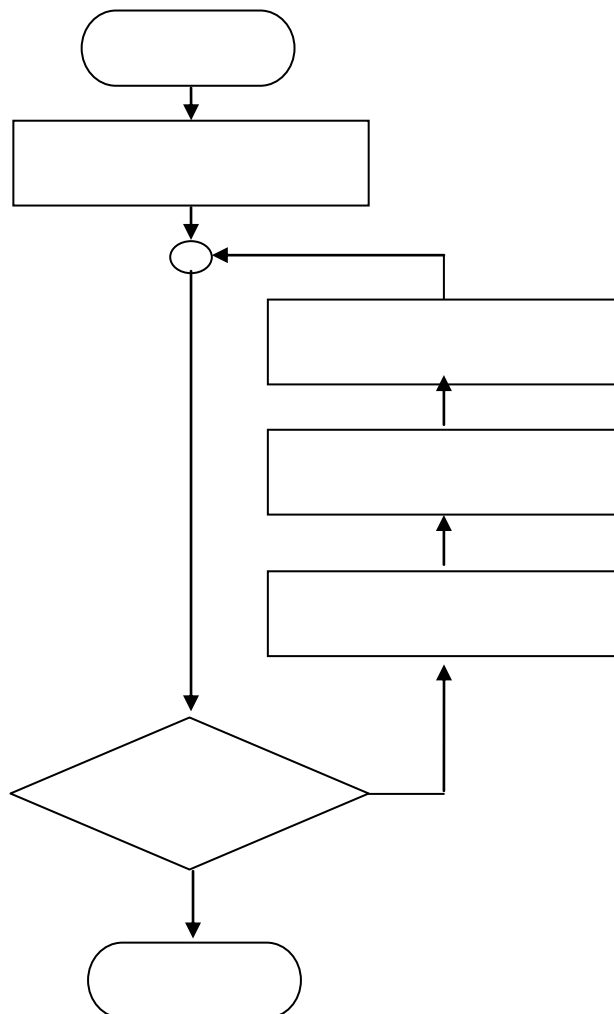
นำข้อความที่เรียงสมบูรณ์แล้วมาเขียนลงในผังงานให้ถูกต้อง



5. จงเรียงลำดับขั้นตอนที่กำหนดให้ถูกต้อง

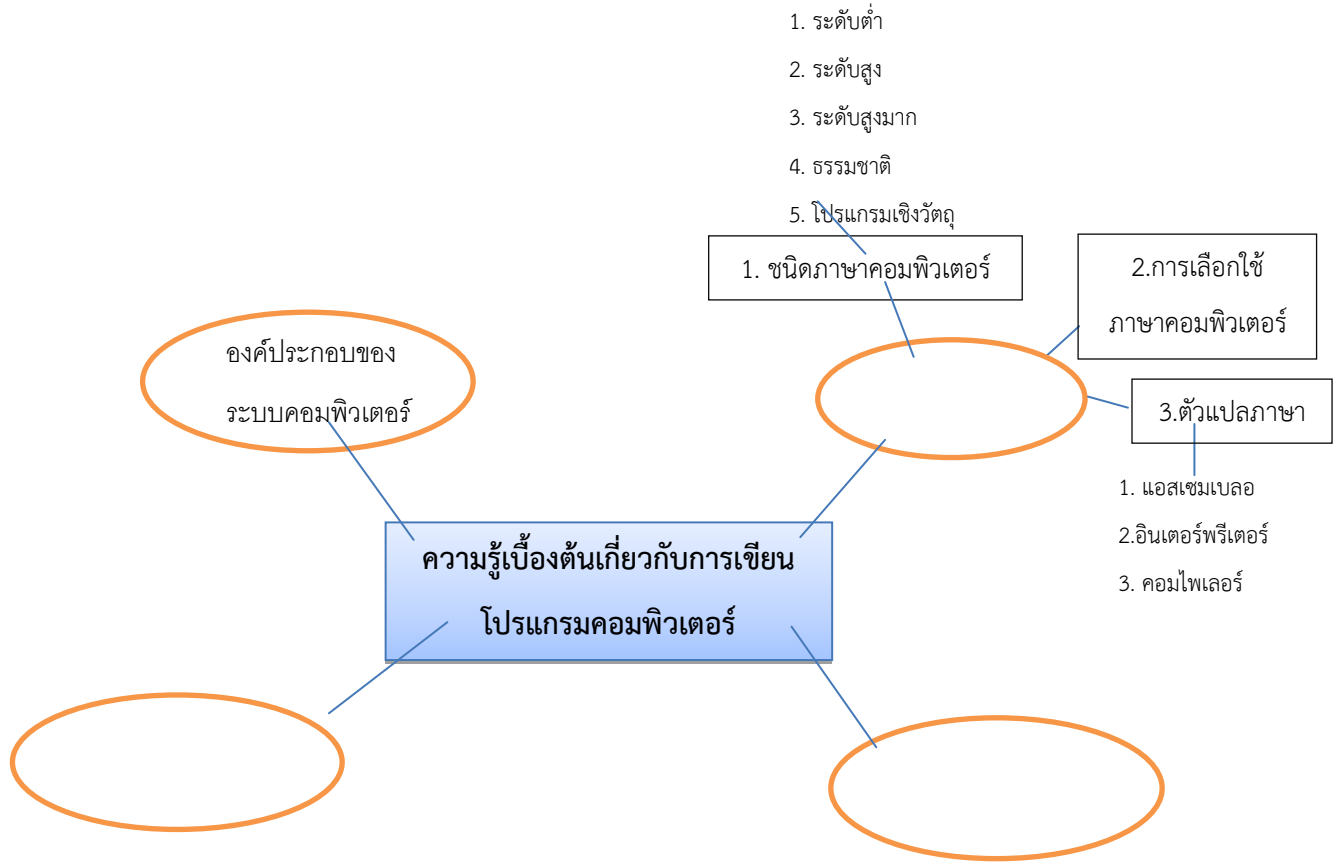
ลำดับขั้นตอนที่กำหนด	การเรียงลำดับขั้นตอนที่ถูกต้อง
ลงทะเบียนขอสอบแก้ตัว	เริ่มต้น
สอบไม่ผ่าน	ประกาศผลสอบปลายภาค
ประกาศผลสอบปลายภาค	สอบไม่ผ่าน
ประกาศผลสอบแก้ตัว	ลงทะเบียนขอสอบแก้ตัว
สอบแก้ตัว	สอบแก้ตัว
เริ่มต้น	ประกาศผลสอบแก้ตัว
จบ	จบ

นำข้อความที่เรียงสมบูรณ์แล้วมาเขียนลงในผังงานให้ถูกต้อง



กิจกรรมที่ 1.6 Mind Mapping

คำชี้แจงให้นักเรียนเขียน Mind Mapping เรื่อง ความรู้เบื้องต้นเกี่ยวกับการเขียนโปรแกรมคอมพิวเตอร์ (หน่วยที่ 1)



ชื่อ-นามสกุล.....ห้อง.....เลขที่.....