ใบความรู้ที่3

หลักการแก้ปัญหา

**ภาษาโปรแกรม** (Programming Languages) ที่มีการคิดค้นขึ้นมาใช้กับคอมพิวเตอร์นั้นมีหลายพัน ภาษา แต่ภาษาที่เป็นที่รู้จักและเป็นที่นิยมใช้ทั่วไปนั้นอาจจะมีเพียงหลายสิบภาษา เช่น โคบอล (COBOL) ปาสคาล (Pascal) เดลไฟล์ (Delphi) วิชวลเบสิก (Visual Basic) ซี (C) จาวา (Java) เป็นต้น ซึ่งแต่ละภาษา สร้างขึ้นด้วยวัตถุประสงค์ที่แตกต่างกันและมีจุดเด่นของภาษาที่ต่างกัน ภาษาซี(C Programming Language) เป็นภาษาเชิงโครงสร้างที่มีการออกแบบโปรแกรมในลักษณะโมดูลที่มีจุดเด่นในเรื่องของประสิทธิภาพการ ทำงานที่เร็ว มีความยืดหยุ่นในการเขียนโปรแกรมสูง เนื่องจากมีผู้ผลิตคอมไพเลอร์เพื่อใช้แปลภาษาซีหลายบริษัท ตัวอย่างต่าง ๆ ที่นำเสนอในหนังสือเล่มนี้ เป็นตัวอย่างที่นำเสนอโดยใช้คอมไพเลอร์ของ Turbo C เวอร์ชัน 3.0 ของบริษัทบอร์แลนด์ โดยพยายามเขียนใน รูปแบบที่เป็นมาตรฐานหากผู้อ่านนำไปใช้กับคอมไพเลอร์ของบริษัทอื่นจะได้มีการปรับแก้ไม่มากนัก เพื่อให้ผู้อ่านได้เห็นภาพการพัฒนาโปรแกรมเชิงโครงสร้างอย่างชัดเจน

**ประวัติความเป็นมา** ภาษาซีได้รับการพัฒนาขึ้นโดยเดนนิส ริทชี (Dennis Ritchie) ขณะที่ทำงานอยู่ที่เบลแล็บบอราทอรี (Bell Laboratories) โดยพัฒนาขึ้นจากหลักการพื้นฐานของภาษาบี (B) และบีซีพีแอล (BCPL) ในช่วงปี ค.ศ. 1971 ถึง1973 แต่ได้เพิ่มชนิดข้อมูลและความสามารถอื่น ๆ ให้มากขึ้น และนำภาษาซีไปใช้พัฒนาระบบปฏิบัติ การยูนิกซ์ (UNIX) บนเครื่องคอมพิวเตอร์ DEC PDP-11 ภาษาซีเป็นที่นิยมใช้อย่างแพร่หลายในช่วงต้นทศวรรษ ที่1980 จนกระทั่งมีความพยายามกำหนดมาตรฐานของภาษาเพื่อให้สามารถใช้ภาษาซีได้บนเครื่องคอมพิวเตอร ์ใด ๆ ในปี ค.ศ. 1983 โดย ANSI (The American National Standards Institute) ได้ตั้งคณะกรรมการ X3J11 เพื่อร่างมาตรฐานดังกล่าว และได้รับการตรวจสอบและยอมรับโดย ANSI และ ISO (The International Standards Organization) โดยมีการตีพิมพ์มาตรฐานของภาษาซีในป ีค.ศ. 1990 จากความมีประสิทธิภาพ และสามารถทำงานบนเครื่องคอมพิวเตอร์ใด ๆ ของภาษาซีจึงได้มีการนำภาษาซีไปใช้ในการพัฒนาระบบปฏิบัติ การต่าง ๆ และใช้เป็นต้นแบบของภาษาอื่น ๆ ที่สำคัญในปัจจุบัน เช่น ซีพลัสพลัส (C++) จาวา (Java) เป็นต้น

**ขั้นตอนการพัฒนาโปรแกรม**

ขั้นตอนการพัฒนาโปรแกรมประกอบด้วย

1. การวิเคราะห์ปัญหา

การวิเคราะห์ปัญหา ประกอบด้วยขั้นตอนต่างๆ ดังนี้

1.กำหนดวัตถุประสงค์ของงาน เพื่อพิจารณาว่าโปรแกรมต้องทำการประมวลผลอะไรบ้าง

2.พิจารณาข้อมูลนำเข้า เพื่อให้ทราบว่าจะต้องนำข้อมูลอะไรเข้าคอมพิวเตอร์ ข้อมูลมีคุณสมบัติเป็นอย่างไร ตลอดจนถึงลักษณะและรูปแบบของข้อมูลที่จะนำเข้า

3.พิจารณาการประมวลผล เพื่อให้ทราบว่าโปรแกรมมีขั้นตอนการประมวลผลอย่างไรและมีเงื่อนไปการประมวลผลอะไรบ้าง

4.พิจารณาข้อสนเทศนำออก เพื่อให้ทราบว่ามีข้อสนเทศอะไรที่จะแสดง ตลอดจนรูปแบบและสื่อที่จะใช้ในการแสดงผล

2.  การออกแบบโปรแกรม

การออกแบบขั้นตอนการทำงานของโปรแกรมเป็นขั้นตอนที่ใช้เป็นแนวทางในการลงรหัสโปรแกรม ผู้ออกแบบขั้นตอนการทำงานของโปรแกรมอาจใช้เครื่องมือต่างๆ ช่วยในการออกแบบ อาทิเช่น คำสั่งลำลอง (Pseudocode) หรือ ผังงาน (Flow chart) การออกแบบโปรแกรมนั้นไม่ต้องพะวงกับรูปแบบคำสั่งภาษาคอมพิวเตอร์ แต่ให้มุ่งความสนใจไปที่ลำดับขั้นตอนในการประมวลผลของโปรแกรมเท่านั้น

3.  การเขียนโปรแกรมด้วยภาษาคอมพิวเตอร์

การเขียนโปรแกรมเป็นการนำเอาผลลัพธ์ของการออกแบบโปรแกรม มาเปลี่ยนเป็นโปรแกรมภาษาคอมพิวเตอร์ภาษาใดภาษาหนึ่ง ผู้เขียนโปรแกรมจะต้องให้ความสนใจต่อรูปแบบคำสั่งและกฎเกณฑ์ของภาษาที่ใช้เพื่อให้การประมวลผลเป็นไปตามผลลัพธ์ที่ได้ออกแบบไว้ นอกจากนั้นผู้เขียนโปรแกรมควรแทรกคำอธิบายการทำงานต่างๆ ลงในโปรแกรมเพื่อให้โปรแกรมนั้นมีความกระจ่างชัดและง่ายต่อการตรวจสอบและโปรแกรมนี้ยังใช้เป็นส่วนหนึ่งของเอกสารประกอบ

4.  การทดสอบและแก้ไขโปรแกรม

การทดสอบโปรแกรมเป็นการนำโปรแกรมที่ลงรหัสแล้วเข้าคอมพิวเตอร์ เพื่อตรวจสอบรูปแบบกฎเกณฑ์ของภาษา และผลการทำงานของโปรแกรมนั้น ถ้าพบว่ายังไม่ถูกก็แก้ไขให้ถูกต้องต่อไป ขั้นตอนการทดสอบและแก้ไขโปรแกรม อาจแบ่งได้เป็น 3 ขั้น

1.  สร้างแฟ้มเก็บโปรแกรมซึ่งส่วนใหญ่นิยมนำโปรแกรมเข้าผ่านทางแป้นพิมพ์โดยใช้โปรแกรมประมวลคำ

2.  ใช้ตัวแปลภาษาคอมพิวเตอร์แปลโปรแกรมที่สร้างขึ้นเป็นภาษาเครื่อง โดยระหว่างการแปลจะมีการตรวจสอบความถูกต้องของรูปแบบและกฎเกณฑ์ในการใช้ภาษา ถ้าคำสั่งใดมีรูปแบบไม่ถูกต้องก็จะแสดงข้อผิดพลาดออกมาเพื่อให้ผู้เขียนนำไปแก้ไขต่อไป ถ้าไม่มีข้อผิดพลาด เราจะได้โปรแกรมภาษาเครื่องที่สามารถให้คอมพิวเตอร์ประมวลผลได้

3.  ตรวจสอบความถูกต้องของการประมวลผลของโปรแกรม โปรแกรมที่ถูกต้องตามรูปแบบและกฎเกณฑ์ของภาษา แต่อาจให้ผลลัพธ์ของการประมวลผลไม่ถูกต้องก็ได้ ดังนั้นผู้เขียนโปรแกรมจำเป็นต้องตรวจสอบว่าโปรแกรมประมวลผลถูกต้องตามต้องการหรือไม่ วิธีการหนึ่งก็คือ สมมติข้อมูลตัวแทนจากข้อมูลจริงนำไปให้โปรแกรมประมวลผลแล้วตรวจสอบผลลัพธ์ว่าถูกต้องหรือไม่ ถ้าพบว่าไม่ถูกต้องก็ต้องดำเนินการแก้ไขโปรแกรมต่อไป การสมมติข้อมูลตัวแทนเพื่อการทดสอบเป็นสิ่งที่มีความสำคัญเป็นอย่างมาก ลักษณะของข้อมูลตัวแทนที่ดีควรจะสมมติทั้งข้อมูลที่ถูกต้องและข้อมูลที่ผิดพลาด เพื่อทดสอบว่าโปรแกรมที่พัฒนาขึ้นสามารถครอบคลุมการปฏิบัติงานในเงื่อนไขต่างๆ ได้ครบถ้วน นอกจากนี้อาจตรวจสอบการทำงานของโปรแกรมด้วยการสมมติตัวเองเป็นคอมพิวเตอร์ทีจะประมวลผล แล้วทำตามคำสั่งทีละคำสั่งของโปรแกรมนั้นๆ วิธีการนี้อาจทำได้ยากถ้าโปรแกรมมีขนาดใหญ่ หรือมีการประมวลผลที่ซับซ้อน

5.  การทำเอกสารประกอบโปรแกรม

การทำเอกสารประกอบโปรแกรมเป็นงานที่สำคัญของการพัฒนาโปรแกรม เอกสารประกอบโปรแกรมช่วยให้ผู้ใช้โปรแกรมเข้าใจวัตถุประสงค์ ข้อมูลที่จะต้องใช้กับโปรแกรม ตลอดจนผลลัพธ์ที่จะได้จากโปรแกรม การทำโปรแกรมทุกโปรแกรมจึงควรต้องทำเอกสารกำกับ เพื่อใช้สำหรับการอ้างอิงเมื่อจะใช้งานโปรแกรมและเมื่อต้องการแก้ไขปรับปรุงโปรแกรม เอกสารประกอบโปรแกรมที่จัดทำ ควรประกอบด้วยหัวข้อต่อไปนี้

1.  วัตถุประสงค์

2.  ประเภทและชนิดของคอมพิวเตอร์และอุปกรณ์ที่ใช้ในโปรแกรม

3.  วิธีการใช้โปรแกรม

4.  แนวคิดเกี่ยวกับการออกแบบโปรแกรม

5.  รายละเอียดโปรแกรม

6.  ข้อมูลตัวแทนที่ใช้ทดสอบ

7.  ผลลัพธ์ของการทดสอบ

6.  การบำรุงรักษาโปรแกรม

เมื่อโปรแกรมผ่านการตรวจสอบตามขั้นตอนเรียบร้อยแล้ว และถูกนำมาให้ผู้ใช้ได้ใช้งาน ในช่วงแรกผู้ใช้อาจจะยังไม่คุ้นเคยก็อาจทำให้เกิดปัญหาขึ้นมาบ้าง ดังนั้นจึงต้องมีผู้คอยควบคุมดูแลและคอยตรวจสอบการทำงาน การบำรุงรักษาโปรแกรมจึงเป็นขั้นตอนที่ผู้เขียนโปรแกรมต้องคอยเฝ้าดูและหาข้อผิดพลาดของโปรแกรมในระหว่างที่ผู้ใช้ใช้งานโปรแกรม และปรับปรุงโปรแกรมเมื่อเกิดข้อผิดพลาดขึ้น หรือในการใช้งานโปรแกรมไปนานๆ ผู้ใช้อาจต้องการเปลี่ยนแปลงการทำงานของระบบงานเดิมเพื่อให้เหมาะกับเหตุการณ์ นักเขียนโปรแกรมก็จะต้องคอยปรับปรุงแก้ไขโปรแกรมตามความต้องการของผู้ใช้ที่เปลี่ยนแปลงไปนั่นเอง