แผนบริหารการสอนประจำบทที่ 1

โปรแกรมคอมพิวเตอร์

หัวข้อประจำบท

1. ระบบคอมพิวเตอร์
2. โปรแกรมคอมพิวเตอร์
3. ภาษาคอมพิวเตอร์
4. ตัวแปลภาษาคอมพิวเตอร์
5. วิวัฒนาการรูปแบบของโปรแกรมภาษาคอมพิวเตอร์

วัตถุประสงค์เชิงพฤติกรรม

1. เพื่อให้ผู้เรียนมีความรู้ความเข้าการทำงานของระบบคอมพิวเตอร์
2. เพื่อให้ผู้เรียนมีความรู้เข้าใจ ความหมาย และการทำงานโปรแกรมคอมพิวเตอร์
3. เพื่อให้ผู้เรียนเข้าใจถึงภาษาคอมพิวเตอร์ และสามารถบอกความแตกต่างระหว่างภาษาคอมพิวเตอร์ในระดับต่าง ๆ ได้
4. เพื่อให้ผู้เรียนมีความรู้ความเข้าใจในตัวแปลภาษา และสามารถบอกความแตกต่างระหว่างตัวแปลภาษาแต่ละประเภทได้
5. เพื่อให้ผู้เรียนมีความรู้ความเข้าใจและบอกความแตกต่างของรูปแบบโปรแกรมภาษาคอมพิวเตอร์ แบบไม่มีโครงสร้าง แบบโครงสร้าง และแบบเชิงวัตถุได้

วิธีการสอนและกิจกรรม

1. ผู้สอนบรรยายในชั้นเรียน ตามหัวข้อเนื้อหาในเอกสารประกอบการสอน วิชาเทคโนโลยีการเขียนโปรแกรมคอมพิวเตอร์
2. ผู้เรียนทำแบบฝึกหัดเพื่อส่งเสริมความเข้าใจ กระบวนการคิด และการแก้ปัญหา

สื่อการเรียนการสอน

1. เอกสารประกอบการสอน วิชา เทคโนโลยีการเขียนโปรแกรมคอมพิวเตอร์
2. โปรแกรม PowerPoint

การวัดผลและการประเมินผล

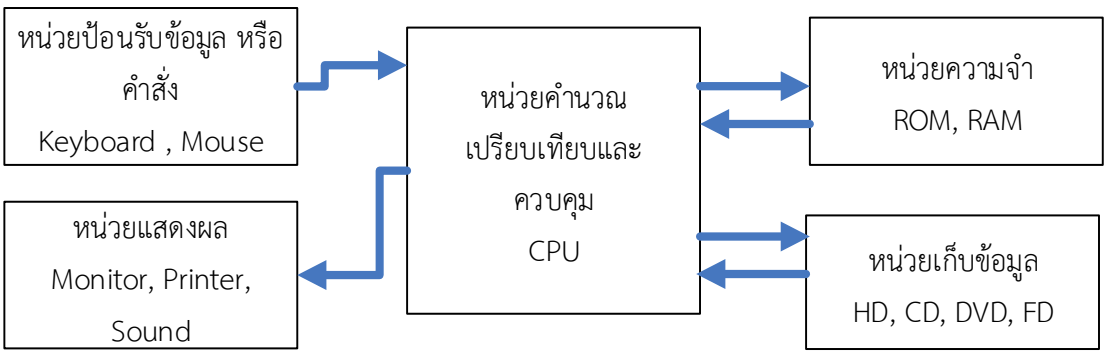
1. สังเกตจากการอภิปราย การตอบคำถาม และซักถามระหว่างเรียน
2. การทำแบบฝึกหัดท้ายบท

บทที่ 1
โปรแกรมคอมพิวเตอร์

ในการทำงานของระบบคอมพิวเตอร์ประกอบไปด้วย การทำงานประสานกันระหว่างตัวเครื่องคอมพิวเตอร์ เรียกว่า “ฮาร์ดแวร์” (Hardware) และส่วนที่เป็นคำสั่ง สั่งงานให้คอมพิวเตอร์ทำงาน เรียกว่า ซอฟต์แวร์“Software” ซึ่งเป็นชุดคำสั่งที่เกิดจากการเขียนจากโปรแกรมภาษา เรียกว่า “โปรแกรมคอมพิวเตอร์” โปรแกรมคอมพิวเตอร์มีรูปแบบและการใช้งานความแตกต่างกันไปตามวิวัฒนาการและจุดประสงค์ของการเขียนโปรแกรมคอมพิวเตอร์ ทั้งนี้ทั้งนั้นเพื่อตอบสนองและสนับสนุนการทำงานของระบบที่ซับซ้อนในปัจจุบัน

1.1 ระบบคอมพิวเตอร์

คอมพิวเตอร์ หมายถึง เครื่องกลที่สามารถรับข้อมูลได้ โดยผ่านอุปกรณ์ต่าง ๆ ที่เชื่อมต่อกับคอมพิวเตอร์ และนำข้อมูลที่ได้ไปประมวลผลทีละขั้นตอน และผลลัพธ์ที่ได้เป็นรูปแบบสัญญาณทางอิเล็กทรอนิกส์ แปลงเป็นข้อมูลแสดงให้มนุษย์ได้รับรู้บนเครื่องมือหรืออุปกรณ์ต่าง ๆ (Evans, 2011) ซึ่งระบบของคอมพิวเตอร์สามารถแบ่งส่วนประกอบการทำงานได้ดังภาพที่ 1.1



ภาพที่ 1.1 ส่วนประกอบของระบบคอมพิวเตอร์

จากภาพที่ 1.1 ส่วนประกอบการทำงานของระบบคอมพิวเตอร์ ประกอบไปด้วย หน่วยป้อนรับข้อมูลหรือคำสั่ง ทำหน้าที่ในการรับข้อมูลจากอุปกรณ์ที่รับข้อมูลหรือคำสั่ง ส่งไปยังหน่วยประมวลผลกลางเพื่อทำการควบคุมการประมวลผล ซึ่งการประมวลผลแบ่งออกเป็น 2 ประเภท คือ คำนวณและเปรียบเทียบ เมื่อประมวลเรียบร้อย ข้อมูลนั้นจะถูกส่งไปยังหน่วยแสดงผล เพื่อแสดงผล

ในอุปกรณ์ต่าง ๆ ต่อไป และในระหว่างที่มีการทำงาน ข้อมูลต่าง ๆ อาจถูกเก็บไว้ในหน่วยความจำเพื่อใช้งานชั่วคราว เมื่อปิดเครื่องข้อมูลก็จะหายไป แต่อาจมีข้อมูลอีกประเภทที่ต้องการเก็บไว้ใช้งานในภายหลัง จะถูกเก็บไว้ในหน่วยเก็บข้อมูล

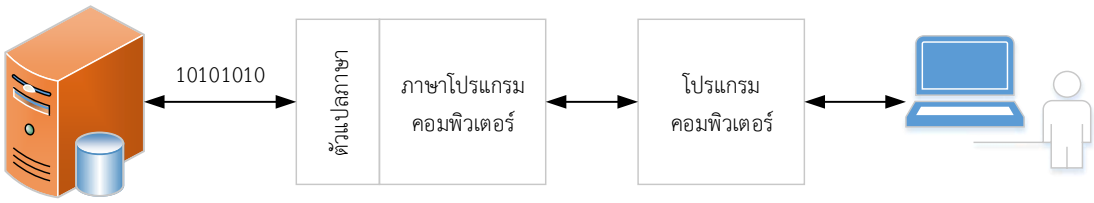
1.2 โปรแกรมคอมพิวเตอร์

การทำงานของระบบคอมพิวเตอร์นั้น คอมพิวเตอร์จะสามารถทำงานได้จะต้องมีข้อมูลภายในระบบ ซึ่งข้อมูลเหล่านี้จะถูกควบคุมการทำงานในการรับส่งข้อมูลผ่าน “โปรแกรมคอมพิวเตอร์ (Computer Programming)” ซึ่งมีผู้ให้ความหมายของโปรแกรมคอมพิวเตอร์ ดังต่อไปนี้

โปรแกรมคอมพิวเตอร์ หมายถึง ชนิดของตัวอักษรที่ถูกสร้างขึ้นโดย “นักพัฒนาโปรแกรม (Programmer)” จุดมุ่งเหมายเพื่อให้คอมพิวเตอร์ทำงานได้ตามที่ต้องการ และโปรแกรมยังเป็นข้อมูลที่อยู่ในหน่วยความจำของคอมพิวเตอร์ (Haverbeke, 2014)

โปรแกรมคอมพิวเตอร์ หมายถึง การสร้างคำสั่ง เพื่อให้เครื่องคอมพิวเตอร์ทำงานที่เป็นประโยชน์ต่อผู้ใช้งาน และด้วยคอมพิวเตอร์สามารถประมวลผลได้เป็นจำนวนมาก จึงมีการนำเอาไปใช้ในงานต่าง ๆ เช่น ด้านวิศวกรรม ด้านวิทยาศาสตร์ และด้านงานศิลปะ เป็นต้น (Evans, 2011)

จากผู้ให้ความหมายของโปรแกรมคอมพิวเตอร์ สามารถสรุปการทำงานของโปรแกรมคอมพิวเตอร์ ได้ดังภาพที่ 1.2



ภาพที่ 1.2 การทำงานของโปรแกรมคอมพิวเตอร์

จากภาพที่ 1.2 อธิบายการทำงานของโปรแกรมคอมพิวเตอร์ ซึ่งเริ่มต้นจากการสั่งงานคอมพิวเตอร์ให้ทำงานตามที่ต้องการผ่านโปรแกรมคอมพิวเตอร์ ซึ่งโปรแกรมคอมพิวเตอร์นั้นถูกสร้างขึ้นด้วยโปรแกรมภาษาคอมพิวเตอร์ โดยโปรแกรมภาษาคอมพิวเตอร์จะมีตัวแปลภาษา ซึ่งจะทำหน้าที่ในการแปลงโปรแกรมที่ถูกพัฒนาขึ้นให้เป็นภาษาที่สามารถติดต่อพูดคุยกับคอมพิวเตอร์ให้สามารถเข้าใจ และทำงานได้ตามคำสั่งที่ได้สั่งงานเข้าไป

1.3 ภาษาคอมพิวเตอร์

ภาษาคอมพิวเตอร์ (Computer Language) หมายถึง ภาษาที่ใช้ในการออกแบบเพื่อใช้ในการอ่าน และเขียนของมนุษย์ สร้างเป็นโปรแกรมขึ้น เพื่อให้คอมพิวเตอร์สามารถทำงานตามที่ต้องการได้ (Evans, 2011) โปรแกรมภาษาคอมพิวเตอร์เป็นสัญลักษณ์ที่มนุษย์คิดค้นขึ้นมาเพื่อใช้แทนคำสั่งสื่อสารสั่งงานระหว่างมนุษย์กับเครื่องคอมพิวเตอร์ หรืออุปกรณ์เชื่อมต่อเป็นเครือข่ายกับคอมพิวเตอร์ ซึ่งมีวิวัฒนาการตั้งแต่ยุคเริ่มแรกจนถึงปัจจุบัน แบ่งออกเป็น 2 ประเภท ดังนี้

1.3.1 ภาษาระดับต่ำ

ภาษาระดับต่ำ (Programming Low Level Language) มีลักษณะการใช้คำสั่งเป็นชุดของเลขฐานสอง หรือใช้สัญลักษณ์เป็นภาษาอังกฤษ ร่วมกับเลขฐานอื่น ๆ เพื่อใช้ในการสั่งงานคอมพิวเตอร์ ภาษาในยุคนี้มีภาพรวมของภาษา จะมีความยากในการเขียนชุดคำสั่ง บุคคลที่สามารถเขียนคำสั่งควบคุมการทำงานของเครื่องคอมพิวเตอร์ ต้องมีผู้มีความรู้ความชำนาญในด้านของอุปกรณ์อิเล็กทรอนิกส์เป็นพื้นฐาน ซึ่งในภาษาในระดับนี้แบ่งออกเป็น 2 ภาษา ประกอบด้วยดังนี้

1.3.1.1 ภาษาเครื่อง (Machine Language) หมายถึง เครื่องคอมพิวเตอร์นั้นจะถูกขับเคลื่อนได้โดยโปรแกรมที่เรียกว่า “รหัสเครื่อง (Machine Code)” โปรแกรมรหัสเครื่องนี้เกิดจากการเรียงกันของคำสั่ง แต่ละคำสั่งประกอบไปด้วยบิต (เลข 0 กับ 1) ที่ซึ่งจะถูกแปลความหมายโดยเครื่องและปฏิบัติตามขอบเขตคำสั่งนั้น รหัสเครื่องเป็นคำสั่งยุคแรกของการสั่งงานคอมพิวเตอร์ (Watt and Brown, 2000) ลักษณะการทำงานดังตารางที่ 1.1 ต่อไปนี้

ตารางที่ 1.1 การสั่งงานคอมพิวเตอร์ด้วยรหัสเครื่อง

การทำงาน	รหัสเครื่อง
ดึงข้อมูลจากหน่วยความจำตำแหน่ง address 366	0000 0001 0110 1110
ใส่ข้อมูลตัวเลข 2 จำนวน ลงไปใน Register 1 และ Register 2	0100 0000 0001 0010
กระโดดไปที่คำสั่ง 13 ถ้าผลลัพธ์ของค่าก่อนนี้เป็น 0	1100 0000 0000 1101

รหัสเครื่องสามารถใช้งานได้ง่ายกับเครื่อง โดยวิธีการโหลดเข้าไปในเครื่องและสามารถทำงานได้ทันที นั่นหมายความว่า เครื่องคอมพิวเตอร์นั้นจะสามารถเข้าใจรหัสเครื่องได้ทันที่เมื่อมีการโปรแกรมเข้าไปในระบบ ดังนั้นสามารถกล่าวได้ว่า ภาษาคอมพิวเตอร์ ก็คือ ภาษาเครื่องนั้นเอง

จากที่กล่าวมาดังกล่าว คอมพิวเตอร์จะติดต่อสื่อสารหรือพูดคุยในลักษณะของรหัสเครื่องที่เป็นเลขฐานสอง ทำให้ยากต่อการอ่าน การเขียน และการแก้ไข นักโปรแกรมคอมพิวเตอร์ต้องอ้างอิง

ตำแหน่งที่อยู่ในหน่วยความจำจริงเพื่อใช้ในการเก็บข้อมูลหรือคำสั่ง และในการโปรแกรมจะต้องอยู่ในรูปของบิตเรียงกัน ทำให้นักโปรแกรมคอมพิวเตอร์ลำบากในการใช้งาน

1.3.1.2 ภาษาแอสเซมบลี หรือภาษาสัญลักษณ์ (Assembly / Symbolic Language) หมายถึง ภาษาระดับต่ำ ที่เรียกว่า “นิวมอนิกโค้ด” (Mnemonic code) ถูกนำไปใช้แทนภาษาเครื่อง เพื่อให้ง่ายต่อการเขียนโปรแกรม (Liang, 2010) ภาษาเอสเซม บลีถูกพัฒนาขึ้นมาใหม่ต่อจากภาษาเครื่อง สืบเนื่องมาจากความยากในการเขียนชุดคำสั่งงานด้วยเลขฐานสอง จึงมีผู้คิดค้นพัฒนาเป็นรูปแบบของภาษาสัญลักษณ์แทนคำสั่งเลขฐานสองมาแทน

1.3.2 ภาษาระดับสูง

ภาษาระดับสูง (Programming High Level Language) คือ โปรแกรมภาษาคอมพิวเตอร์ที่ถูกพัฒนาขึ้น คำสั่งคล้ายกับภาษาอังกฤษ ง่ายต่อการเรียนรู้ และการเขียนโปรแกรม (Liang, 2015) ภาษานี้พัฒนาต่อเนื่องมาจากภาษาระดับต่ำ ลักษณะภาษาระดับสูงมีการกำหนดสัญลักษณ์การใช้งานคำสั่งเป็นรูปแบบของอักษรภาษาอังกฤษ ที่ใกล้เคียงกับข้อความที่ใช้ในการสื่อสารกันของมนุษย์ จึงทำให้ผู้เขียนโปรแกรมสามารถเขียนคำสั่ง สั่งงานควบคุมการทำงานคอมพิวเตอร์ได้ง่ายขึ้น

ภาษาระดับสูงมีผู้คิดค้นพัฒนาขึ้นมามากมายหลายภาษา สาเหตุนั่นก็คือ ความแตกต่างกันลักษณะของระบบงาน ที่จะนำภาษาคอมพิวเตอร์ไปพัฒนาเป็นโปรแกรม เพื่อให้คอมพิวเตอร์ทำงานตามวัตถุประสงค์ที่ต้องการ ตัวอย่างภาษาระดับสูงมีดังต่อไปนี้

1.3.2.1 ภาษาฟอร์แทน (FORTRAN : FORmular TRANslator) คือ ภาษาที่นำไปใช้ในด้านการคำนวณทางคณิตศาสตร์ และแก้ปัญหางานทางวิทยาศาสตร์ พัฒนาขึ้นมาโดยบริษัท ไอบีเอ็ม (IBM)

1.3.2.2 ภาษาเบสิก (BASIC : Beginner’s All-purpose Symbolic Instruction Code) คือ ภาษาที่ออกแบบให้ง่ายต่าการศึกษาเรียนรู้ ภาษานี้ใช้ในงานต่าง ๆ ได้อย่างกว้างขวางทั้งงานทางธุรกิจและงานอื่น ๆ พัฒนาขึ้นมาใช้งานโดยสถาบันมาตรฐานแห่งชาติของสหรัฐอเมริกา (American National Standards Institute : ANSI)

1.3.2.3 ภาษาโคบอล (COBOL : Common Business Oriented Language) คือ ภาษาที่ถูกออกแบบเพื่อประยุกต์ใช้ในงานด้านธุรกิจ ส่วนใหญ่งานถูกนำไปใช้ คือ ประมวลผลข้อมูลในเชิงธุรกิจ

1.3.2.4 ภาษาปาสคาล (Pascal) คือ ภาษาที่ถูกพัฒนาขึ้นโดยตั้งชื่อเป็นเกียรติกับ Blaise Pascal พัฒนาขึ้นมาใช้ในงานด้านการคำนวณทั่วไป ทั้งงานทางวิทยาศาสตร์ ธุรกิจ หรือวิศวกรรม เช่นกัน พัฒนาโดย นายเดนนิส ริชชี่ (Dennis Ritchie) ได้รับการสนับสนุนจากบริษัทเอทีแอนท์ที (AT&T) นิยมใช้ระบบปฏิบัติการยูนิกซ์ (UNIX)

1.3.2.5 ภาษาเอดา (ADA) คือ ภาษาที่ถูกพัฒนาขึ้นตามโครงการป้องกันประเทศของทหาร ประเทศสหรัฐอเมริกา ภาษาเอดา ตั้งชื่อให้เกียรติแก่ นางออกัสตา เอดา (Augusta Ada) เป็นภาษาหลักใช้ในการทางทหาร โดยการสนับสนุนของกระทรวงกลาโหมของประเทศสหรัฐอเมริกา

1.3.2.6 ภาษาวิชวลเบสิค (Visual Basic) และเดลไฟ (Delphi) คือ ภาษาที่มีการใช้รูปแบบส่วนประสานกราฟิกกับผู้ใช้ (Graphic User Interface) ทำให้สามารถสร้างโปรแกรมประยุกต์ต่าง ๆ ได้รวดเร็วขึ้น

1.3.2.7 ซีพลัสพลัส (C++) คือ ภาษาที่นิยมใช้ในการพัฒนาโปรแกรมระบบ (System Software) เช่น พัฒนาตัวแปลภาษาคอมไพเลอร์ และระบบปฏิบัติการ (Operating System) ระบบปฏิบัติการถูกสร้างขึ้นด้วยภาษา ซีพลัสพลัส ที่เห็นได้ชัดที่สุดก็ ระบบปฏิบัติการของวินโดว์ ของไมโครซอฟท์

1.3.2.8 ภาษาจาวา(Java) คือ ภาษาที่ถูกพัฒนาขึ้นด้วยบริษัทSun Microsystems ถูกใช้งานอย่างกว้างขวาง เพื่อใช้ในการสร้างกรอบการใช้งานโปรแกรมบนอินเทอร์เน็ต

1.4 ตัวแปลภาษาคอมพิวเตอร์

การเขียนโปรแกรมในปัจจุบัน โปรแกรมภาษาจะเป็นภาษาที่ใกล้เคียงกับภาษามนุษย์ แต่เครื่องคอมพิวเตอร์ไม่สามารถเข้าใจภาษานี้ได้ จึงจำเป็นจะต้องมีตัวกลางที่จะทำหน้าที่ในการแปลภาษาให้เป็นภาษาเครื่องที่คอมพิวเตอร์เข้าใจ เรียกว่า “ตัวแปลภาษา (Translator Program)” ซึ่งจะทำหน้าที่ในการแปลงโปรแกรมให้เป็นเลขฐานสอง หรือภาษาเครื่องนั่นเอง ตัวแปลภาษามีรายละเอียดการทำงานดังต่อไปนี้

1.4.1 ลักษณะการทำงานของตัวแปลภาษาคอมพิวเตอร์

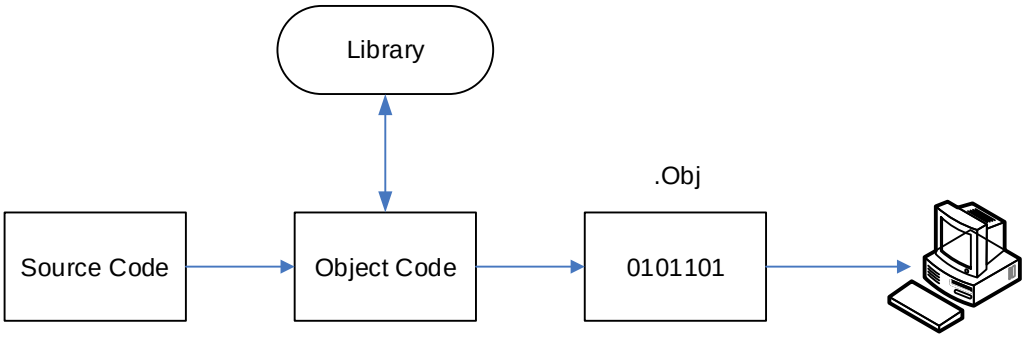
การเขียนโปรแกรมด้วยภาษาแอสเซมบลี และภาษาระดับสูง โปรแกรมที่ถูกเขียนขึ้นจะเรียกว่า “โปรแกรมต้นฉบับ (Source Program)” เมื่อมีการบันทึกโปรแกรมต้นฉบับ เครื่องคอมพิวเตอร์จะกำหนดชนิดของโปรแกรม ตามโปรแกรมภาษาที่ใช้ในการสร้างโดยอัตโนมัติ เช่น ภาษาซี จะถูกกำหนดให้โปรแกรมต้นฉบับไฟล์มีนามสกุล .C ดังนั้น แต่ละโปรแกรมภาษาคอมพิวเตอร์จะมีตัวแปลภาษา ที่ทำหน้าที่แปลรหัสคำสั่งที่ใช้ในรูปแบบคำส่งที่ภาษากำหนดไว้ให้เป็นเลขฐานสอง (Object Program) ซึ่งเป็นภาษาที่เครื่องคอมพิวเตอร์เข้าใจ และนำไปปฏบัติงานตามที่ต้องการได้ เมื่อมีการแปลโปรแกรมต้นเสร็จสิ้น เครื่องคอมพิวเตอร์จะทำการสร้างโปรแกรมเพิ่มขึ้นอีก 1 โปรแกรมโดยอัตโนมัติ ให้มีชนิดของโปรแกรมเป็นชนิด .OBJ .EXE หรือ .COM เพื่อนำไปใช้งานในรูปแบบของภาษาเครื่อง

ตัวแปลภาษา ได้มีการนำมาใช้นับตั้งแต่ภาษาเอสเซมบลี จนมาถึงภาษาระดับสูงในปัจจุบัน สามารถแบ่งประเภทของตัวแปลภาษาได้ดังต่อไปนี้

1.4.2 ประเภทของตัวแปลภาษาคอมพิวเตอร์

ตัวแปลภาษา แบ่งตามวิวัฒนาการของการพัฒนาโปรแกรมภาษาคอมพิวเตอร์ แบ่งออกได้ 3 ประเภทดังต่อไปนี้

1.4.2.1 แอสเซมเบลอร์ (Assembler) คือ ตัวแปลภาษาที่ทำหน้าที่ในการแปลงภาษาแอสเซมบลี ให้เป็นภาษาเครื่อง (Liang, 2015) โดยการใช้ภาษาแอสเซมบลีจะให้ แอสเซมเบลอร์แปลภาษาให้อยู่ในรูปของรหัสคำสั่ง .OBJ ภาษาแอสเซมบลีค่อนข้างมีความยุ่งยากในการใช้งาน เพราะการเขียนโปรแกรมเป็นรูปแบบของภาษาสัญลักษณ์รวมกับเลขฐาน ซึ่งเมื่อเปรียบเทียบกับการใช้ภาษาระดับสูง เช่น ภาษา C หรือภาษา BASIC ซึ่งมีการใช้งานง่ายกว่า แต่ผลดีของภาษาแอสเซมบลี คือ การทำงานของโปรแกรมเร็วกว่า และขนาดของตัวโปรแกรมมีขนาดเนื้อที่น้อยกว่าโปรแกรมที่สร้างจากภาษาอื่นมาก จึงนิยมใช้ภาษานี้เมื่อต้องการประหยัดเวลาทำงานของเครื่องคอมพิวเตอร์ และเพิ่มประสิทธิภาพของโปรแกรม โครงสร้างการทำงานตัวแปลภาษาแอสเซมเบลอร์ มีรายละเอียดดังภาพที่ 1.3

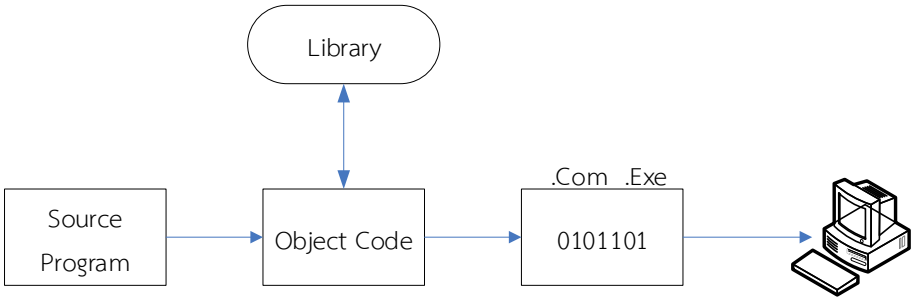


ภาพที่ 1.3 การทำงานของตัวแปลภาษาแอสเซมเบลอร์

1.4.2.2 ตัวแปลภาษาคอมไพเลอร์ (Complier) หมายถึง โปรแกรมคอมพิวเตอร์ที่ถูกสร้างด้วยโปรแกรมอีกโปรแกรม ซึ่งคอมไพเลอร์จะทำหน้าที่ในการแปลข้อมูลโปรแกรมที่ถูกสร้างขึ้นในภาษาระดับสูง ใช้งานง่ายสำหรับมนุษย์ ให้เป็นโปรแกรมภาษาเครื่อง ซึ่งคอมพิวเตอร์จะประมวลผลได้ (Evans, 2011)

ตัวแปลภาษาคอมไพเลอร์ทำหน้าที่แปลโปรแกรมต้นฉบับทั้งโปรแกรมให้เป็น“รหัสออบเจ็ค (Object Code)” ในระหว่างการแปลรหัส หากพบข้อผิดพลาดของคำสั่งกรณีใช้ผิดกฎเกณฑ์รูปแบบคำสั่งที่กำหนดไว้ของแต่ละภาษาจะแสดงข้อความแจ้งข้อผิดพลาดที่เกิดขึ้น และหยุดการแปล นักเขียนโปรแกรมต้องแก้ไขข้อผิดพลาดนั้น และทำการแปลใหม่ จะกระทั่งไม่มีข้อผิดพลาดของโปรแกรม จึงได้โปรแกรมใหม่ที่เรียกว่า “โปรแกรมออบเจ็ค” หลังจากนั้นจึงนำโปรแกรมออบเจ็คที่ได้ไปทำการเชื่อมโยง (Link) เข้ากับระบบหรือไลบรารี (Library) แปลงโปรแกรมให้เป็น

ไฟล์โปรแกรมที่สามารถพูดคุยกับคอมพิวเตอร์ได้ (.com หรือ .exe) และนำไฟล์ที่ได้ไปสั่งให้เครื่องคอมพิวเตอร์ทำงานตามคำสั่งต่อไปดังภาพที่ 1.4



ภาพที่ 1.4 การทำงานของตัวแปลภาษาคอมไพเลอร์

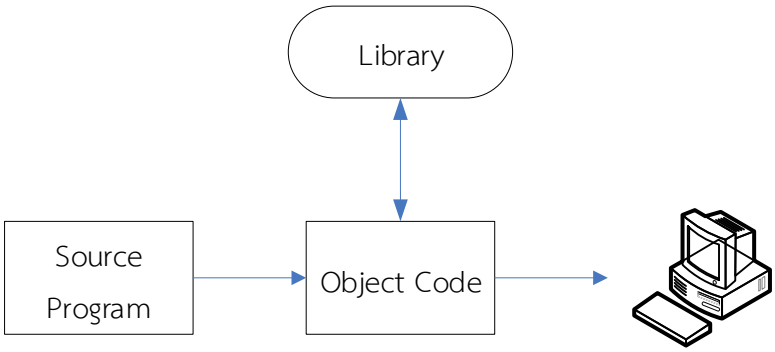
การทำงานของลักษณะตัวแปลภาษาแบบคอมไพเลอร์ จากภาพที่ 1.4 สามารถแยกประเภทข้อดีและข้อเสียของการทำงานตัวแปลภาษาคอมไพเลอร์ได้ดังต่อไปนี้

1) ข้อดี คือ เมื่อผ่านการคอมไพล์ จะได้ไฟล์ที่เป็นออบเจ็คต์โค้ด เมื่อมีการใช้งานโปรแกรมนี้อีกครั้ง สามารถเรียกไฟล์ที่เป็นออบเจ็คต์โค้ดไปสั่งงานคอมพิวเตอร์ได้ทันที โดยไม่ต้องไปผ่านกระบวนการคอมไพล์ใหม่อีกครั้ง

2) ข้อเสีย คือ จะแสดงผลลัพธ์ที่ได้มีความล่าช้า เพราะทุกครั้งที่มีการประมวลผล เพื่อแสดงผลลัพธ์ที่ได้ของโปรแกรม จะต้องผ่านขบวนการตรวจสอบความถูกต้องทุกบรรทัดของโปรแกรม

1.4.2.3 ตัวแปลภาษาอินเทอร์พรีเตอร์ (Interpreter) คือ ตัวแปลภาษาที่เป็นเครื่องมือในการแปลภาษาระดับสูง ให้เป็นภาษาเครื่อง หลักการทำงาน ก็คือตัวแปลภาษาประเภทนี้จะแปลภาษาทีละบรรทัด หากบรรทัดนั้นถูกต้องตามหลักของภาษา ก็จะประมวลผลชุดคำสั่ง และนำไปแสดงผลลัพธ์ แต่ถ้าหากบรรทัดใดมีข้อผิดพลาด จะหยุดการทำงานบรรทัดนั้น และจะข้ามกระโดดไปทำงานในบรรทัดถัดไป (Evans, 2011) การทำงานลักษณะนี้จะเห็นผลลัพธ์ที่ชุดคำสั่งถูกต้องและส่วนที่ผิดพลาดพร้อมกัน

ตัวแปลภาษาแบบอินเทอร์พรีเตอร์ โดยโปรแกรมภาษาคอมพิวเตอร์ต่าง ๆ ที่ใช้ตัวแปลภาษาแบบอินเทอร์พรีเตอร์ จะมี ไลบลารี เก็บคำสั่งของโปรแกรมนั้นๆ เอาไว้ เมื่อมีการแปลคำสั่ง จะทำการตรวจสอบคำสั่งกับไลบลารี เพื่อความตรวจสอบความถูกต้องของคำสั่งดังภาพที่ 1.5



ภาพที่ 1.5 การทำงานของตัวแปลภาษาอินเทอร์พรีเตอร์

การทำงานของลักษณะตัวแปลภาษาแบบอินเทอร์พรีเตอร์ จากภาพที่ 1.5 สามารถแยกประเภทข้อดีและข้อเสียของการทำงานตัวแปลภาษาอินเทอร์พรีเตอร์ได้ดังต่อไปนี้

1) ข้อดี คือ สามารถสั่งแสดงผลการทำงานได้ทันที โดยไม่ต้องตรวจสอบชุดคำสั่งทั้งหมด หรือตรวจสอบโปรแกรมทั้งที่โปรแกรมยังไม่เสร็จสมบูรณ์ ปัจจุบันที่เห็นมากที่สุดก็คือ กลุ่มภาษาสคริปต์ ที่ทำงานบนเว็บ

2) ข้อเสีย คือ เนื่องจากไม่มีโปรแกรมที่ถูกแปลรหัสเก็บไว้ หากมีการประมวลผลใหม่ โดยเฉพาะโปรแกรมมีความยาวมาก จะต้องเริ่มอ่านคำสั่งจากจุดเริ่มของโปรแกรม จึงทำให้การประมวลผลได้ช้า

1.5 วิวัฒนาการรูปแบบของโปรแกรมภาษาคอมพิวเตอร์

โปรแกรมภาษาคอมพิวเตอร์ได้มีวิวัฒนาการ ตั้งแต่ต้องใช้งานจากภาษาเครื่อง มาเป็นภาษาที่ใกล้เคียงกับภาษามนุษย์ในปัจจุบันหรือเรียกว่าเป็นภาษาระดับสูง. มีรูปแบบของการเขียนโปรแกรมรายละเอียดดังต่อไปนี้

1.5.1 การเขียนโปรแกรมแบบไม่มีโครงสร้าง (Non-Structural Programming)

ในการเขียนโปรแกรมแบบไม่มีโครงสร้าง เป็นยุคเริ่มแรกของการสร้างชุดคำสั่งในการสั่งงานคอมพิวเตอร์ การสั่งงานคอมพิวเตอร์จะใช้สั่งโดยตรง ซึ่งยุคเริ่มแรกจะมีการใช้คอมพิวเตอร์ยังไม่หลากหลายเนื่องจากยากต่อการใช้งาน ลักษณะของการใช้เป็นลักษณะของภาษาเครื่อง และภาษาสัญลักษณ์ จากภาพที่ 1.6 แสดงการสร้างชุดคำสั่งในการสั่งงานคอมพิวเตอร์ด้วยภาษาเครื่อง เป็นโปรแกรมในการบวกตัวเลข 2 ตัวเลข ซึ่งจะเห็นได้ว่าการสั่งงานคอมพิวเตอร์จะใช้เลขฐานสองทั้งหมด โดยการสั่งงานจะต้องเขียนคำสั่งที่ต้องการลงไปโดยตรง

2+2=4
0 0 1 0 0 0 0 1 0 0 0 0 1 1 1 1 0 0 1 0
0 0 1 0 0 0 0 1 1 0 0 0 1 1 1 1 0 0 1 0
0 0 1 0 0 0 0 1 0 0 1 1 1 1 1 1 1 1 0 0
0 0 1 0 0 0 0 1 0 0 1 1 1 1 0 1 1 1 0 0
0 0 1 0 1 0 0 1 1 0 0 0 1 1 1 1 0 0 1 0
1 1 1 1 0 0 1 0 0 1 1 1 0 1 1 1 0 1 1 1
1 1 1 1 0 0 1 0 0 1 1 1 1 0 1 1 1 1 0 1
1 1 1 1 1 1 0 0 0 1 0 1 1 0 1 0 1 1 0 1
1 1 0 1 1 1 0 0 0 1 0 1 1 0 1 0 1 1 1 1
1 1 1 1 0 0 1 0 0 1 1 1 0 0 1 1 1 1 0 1

ภาพที่ 1.6 การสร้างชุดคำสั่งด้วยภาษาเครื่อง

ภาษาสัญลักษณ์มีลักษณะของการใช้คำสั่งคล้ายกับภาษาเครื่อง โดยเมื่อต้องการสั่งงานคอมพิวเตอร์ก็สามารถสร้างชุดคำสั่ง สั่งงานได้โดยตรง แต่ลักษณะของคำสั่งที่ใช้นั้นจะแตกต่างจากภาษาเครื่องนั่นคือ จะใช้สัญลักษณ์รวมกับเลขฐานต่าง ๆ แทนชุดของเลขฐานสองทั้งหมด ทั้งนี้ชุดคำสั่งด้วยด้วยภาษาสัญลักษณ์แทนเลขฐานสอง ก็เพื่อสะดวกในการจดจำและการใช้งาน ซึ่งจะสามารถใช้งานคอมพิวเตอร์ได้ง่ายขึ้น แต่ว่าชุดคำสั่งเหล่านี้จะไม่สั่งงานคอมพิวเตอร์ได้โดยตรง จะต้องไปผ่านกระบวนการแปลเป็นภาษาเครื่องเสียก่อน ภาษาที่มีลักษณะการทำงานแบบนี้ คือ ภาษาแอสเซมบลี ซึ่งจะใช้ตัวแปลภาษาแอสเซมเบลอร์ ทำการแปลให้เป็นภาษาเครื่องที่คอมพิวเตอร์เข้าใจ ดังตารางที่ 1.2 แสดงการเขียนโปรแกรมแอสเซมบลี ให้แสดงข้อความ “hello world”

ตารางที่ 1.2 ตัวอย่างการเขียนโปรแกรมด้วยภาษาเอสเซมบลี

```
org 100h ; set video mode
mov ax, 3 ; text mode 80x25, 16 colors, 8 pages (ah=0, al=3)
int 10h ; do it!
        ; cancel blinking and enable all 16 colors:
mov ax, 1003h
mov bx, 0
int 10h
        ; set segment register:
```

```
    mov ax, 0b800h
    mov ds, ax
                ; print "hello world"
                ; first byte is ascii code, second byte is color code.
    mov [02h], 'H'
    mov [04h], 'e'
    mov [06h], 'l'
    mov [08h], 'l'
    mov [0ah], 'o'
    mov [0ch], ','
    mov [0eh], 'W'
    mov [10h], 'o'
    mov [12h], 'r'
    mov [14h], 'l'
    mov [16h], 'd'
    mov [18h], '!'
                ; color all characters:
    mov cx, 12 ; number of characters.
    mov di, 03h ; start from byte after 'h'
    c: mov [di], 32004101001h ; light red (1100) on yellow (1110)
    add di, 2 ; skip over next ascii code in vga memory.
    loop c
                ; wait for any key press:
    mov ah, 0
    int 16h
ret
```

ที่มา ภาษาเอสเซมบลี (2558)

1.5.2 การเขียนโปรแกรมแบบมีโครงสร้าง

การเขียนโปรแกรมแบบไม่มีโครงสร้างพบว่าการใช้คำสั่งนั่น จะเป็นคำสั่งที่เป็นภาษาเครื่องหรือใกล้เคียง ซึ่งภาษาเหล่านี้จะสามารถสื่อสารกับคอมพิวเตอร์ได้อย่างรวดเร็ว ถึงแม้จะมีการดำเนินการแปลชุดคำสั่งในส่วนของภาษาสัญลักษณ์ แต่การประมวลผลก็ยังมีความรวดเร็วในการทำงาน เป็นข้อดีของการพัฒนาชุดคำสั่งแบบไม่มีโครงสร้าง แต่ในทางกลับกัน ในการสร้างชุดคำสั่งเหล่านี้ การป้อนชุดคำสั่ง จะถูกดำเนินการด้วยมนุษย์ ซึ่งการจดจำคำสั่งเหล่านี้เป็นเรื่องยากมากในการที่จะจดจำ ดังนั้นการใช้งานคอมพิวเตอร์ทั่วไปในยุคเริ่มแรก ก็จะมีเฉพาะกลุ่มบุคคลที่มีความเกี่ยวข้องโดยตรงกับคอมพิวเตอร์เท่านั้นที่จะใช้งานได้ เมื่อวิวัฒนาการของระบบคอมพิวเตอร์ได้พัฒนาเพิ่มมากขึ้น มนุษย์มีการใช้งานคอมพิวเตอร์เพิ่มมาก จึงมีการคิดค้นวิธีการที่จะสั่งงานคอมพิวเตอร์ให้ได้ง่ายขึ้น มีการสร้างชุดคำสั่งที่มีภาษาใกล้เคียงกับภาษามนุษย์ ให้สามารถสั่งงานคอมพิวเตอร์ได้ง่ายมากยิ่งขึ้น คือ การเขียนโปรแกรมแบบมีโครงสร้าง (Structural Programming)

การเขียนโปรแกรมแบบมีโครงสร้าง มีการพัฒนาขึ้นมาใช้งานมากมายหลายโปรแกรมภาษาคอมพิวเตอร์ ซึ่งแต่ละโปรแกรมภาษาถูกพัฒนาให้มีการใช้งานในแต่ละด้านแตกต่างกันออกไป เช่น ด้านธุรกิจ ด้านการทหาร ด้านนักวิทยาศาสตร์ ด้านงานวิจัย เป็นต้น ซึ้งชุดคำสั่งเหล่านี้ถูกสร้างขึ้นโดยเฉพาะกลุ่ม วิธีการพัฒนาหรือชุดคำสั่งต่าง ๆ จะไม่เหมือนกันเพราะผู้พัฒนาต่างกัน แต่โครงสร้างในการดำเนินการพัฒนาทุกโปรแกรมภาษาระดับสูง ก็จะมีลักษณะคล้ายกันนั่นก็คือ จะมีโครงสร้าง ระเบียบ กฎเกณฑ์ในการสร้างชุดคำสั่งเหล่านั่น ซึ่งลักษณะแบบนี้เป็นการพัฒนาโปรแกรมแบบมีโครงสร้าง เช่น ให้หาผลลัพธ์จากการคูณของตัวเลข 2 จำนวน ด้วยภาษาซี ดังตารางที่ 1.3

ตารางที่ 1.3 การสร้างชุดคำสั่งแบบมีโครงสร้างด้วยภาษาซี

บรรทัดที่	ชุดคำสั่ง
1	#include <stdio.h>
2	main() {
3	int num1,num2,result;
4	printf("Enter first number :"};
5	scanf ("%d", &num1);
6	printf("Enter second number :"};
7	scanf ("%d", &num2);
8	result = num1*num2;
9	printf("Result is %d",result); return;
10	}

จากตารางที่ 1.3 การพัฒนาชุดคำสั่งโดยโปรแกรมภาษซี ซึ่งเป็นลักษณะของโปรแกรมภาษาแบบมีโครงสร้าง ซึ่งมีวิธีการ และกฎระเบียบในการพัฒนาชุดคำสั่ง โดยก่อนการสร้างชุดคำสั่งสั่งงานคอมพิวเตอร์ จะต้องมีการเรียกใช้ฟังก์ชั่นเพื่อใช้ต่าง ๆ ในโปรแกรมด้วย #include ตามด้วย ไลบลารี่ (Library) ที่เก็บคำสั่งต่าง ๆ เหล่านั้นไว้ เช่น คำสั่ง printf และ scanf จะถูกเก็บเอาไว้ในไลบลารี่ stdio.h โปรแกรมหลัก (เมื่อมีการประมวลผลชุดคำสั่ง จะประมวลผลส่วนนี้เป็นส่วนแรก) จะต้องเขียนคำสั่ง main() และชุดคำสั่งที่สร้างขึ้นต้องอยู่ระหว่างสัญลักษณ์ “{” และ “}” ซึ่งกำหนดขอบเขตการทำงานของโปรแกรมหลัก

ตารางที่ 1.4 การสร้างชุดคำสั่งแบบมีโครงสร้างภาษาปาสคาล

บรรทัดที่	ชุดคำสั่ง
1	Program multiply
2	uses crt;
3	var num1,num2,result : interger;
4	begin
5	clrscr;
6	result :=num1*num2;
7	writeln(‘Result is : ’,result);
8	readln;
9	end.

เพื่อให้เกิดความเข้าใจที่เพิ่มมากขึ้น จากโปรแกรมตารางที่ 1.4 การสร้างชุดคำสั่งแบบมีโครงสร้างด้วยภาษาปาสคาล เป็นลักษณะของโปรแกรมแบบมีโครงสร้างอีกโปรแกรมภาษา คือ ภาษาปาสคาล ซึ่งการใช้งานภาษาในลักษณะกลุ่มเดียวกันกับภาษาซี โดยการมีโครงสร้าง และกฎในการสร้างชุดคำสั่งดังนี้ นั่นคือ โปรแกรมที่สร้างขึ้นจะต้องมีชื่อของโปรแกรม โดยการตั้งชื่อจะต้องใส่ชื่อหลังคำสั่ง Program การสร้างชุดคำสั่งต่าง ๆ จะต้องมีการเรียกใช้คำสั่งในไลบลารี่หลังคำสั่ง uses (บรรทัดที่ 2) ซึ่งโปรแกรมภาษาปาสคาล จะมีไลบราลี่ เหมือนกับภาษาซี ซึ่งเก็บฟังก์ชันการใช้งานต่าง ๆ เมื่อมีการใช้งาน เช่น writeln (บรรทัดที่ 7) โปรแกรมภาษาก็จะไปเรียกคำสั่งนี้ที่ถูกเก็บไว้ในไลบราลี่ crt ขึ้นมาใช้งานได้ทันที

โปรแกรมภาษาแบบมีโครงสร้างโดยทั่วไปจะมีโปรแกรมหลัก โดยโปรแกรมภาษาปาสคาลจะเริ่มต้นด้วยคำสั่ง begin และปิดชุดคำสั่ง ด้วยคำสั่ง end คำสั่ง begin คือคำสั่งที่ใช้ในการเปิดโปรแกรมหลัก หากมีโปรแกรมย่อย โปรแกรมจะเรียกใช้โปรแกรมย่อยอีกครั้ง

จากตัวอย่างโปรแกรมตัวอย่างในข้างต้น จะเห็นได้ว่าลักษณะของโปรแกรมภาษแต่ละโปรแกรมภาษา ไม่เหมือนกัน ขึ้นอยู่กับผู้พัฒนาซึ่งจะเป็นผู้คิดค้นต้นแบบขึ้นมา ผู้ใช้เมื่อจะใช้งานโปรแกรมภาษาจะต้องศึกษา และทำความเข้าใจ โครงสร้าง กฎระเบียบและวิธีการในการสร้างชุดคำสั่งแต่ละโปรแกรมภาษา แต่ไม่ว่าจะเป็นโปรแกรมภาษาใด จะสังเกตได้ว่า คำสั่งที่ใช้ในแต่ละโปรแกรมภาษาจะเป็นคำสั่งที่ใกล้เคียงกับภาษามนุษย์ ทำให้มนุษย์สามารถใช้งานคอมพิวเตอร์ได้ง่ายขึ้น แต่ในทางกับกัน คอมพิวเตอร์ไม่ได้คุยกันเป็นภาษามนุษย์ ดังนั้นชุดคำสั่งเหล่านี้จะสั่งงานโดยตรงไม่ได้ จะต้องผ่านกระบวนการแปลภาษาให้เป็นภาษาเครื่องเสียก่อน ซึ่งในภาษาระดับสูงนั้น จะมีตัวแปลภาษา 2 ประเภทด้วยกัน คือ คอมไพเลอร์ และอินเทอร์พรีเตอร์ เมื่อโปรแกรมภาษาระดับสูงต้องผ่านกระบวนการแปลภาษา จะทำให้เกิดการสูญเสียเวลาในการแปลผล ผลกระทบที่ได้ที่จะแสดงให้เห็นกับผู้ใช้งานนั่นก็คือ เมื่อมีการเขียนโปรแกรมที่มีขนาดใหญ่และมีความซับซ้อนมาก การประมวลสั่งงานของคอมพิวเตอร์และการแสดงผลลัพธ์ที่จะช้าลง

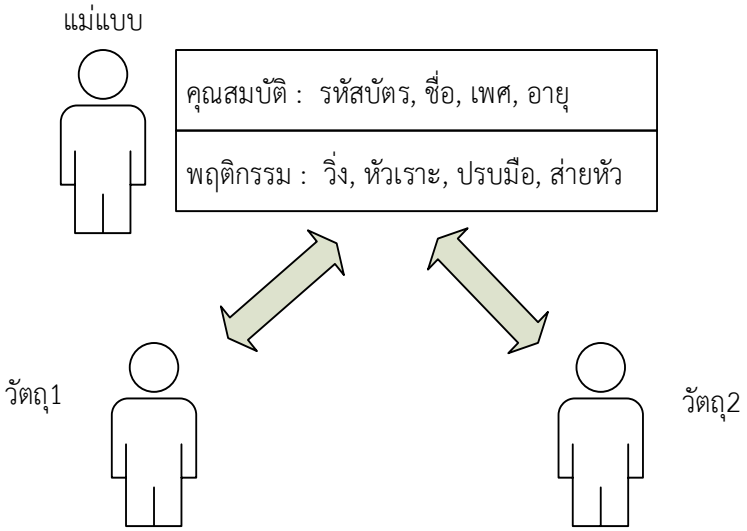
1.5.3 การเขียนโปรแกรมแบบเชิงวัตถุ

การพัฒนาโปรแกรมแบบไม่มีโครงสร้าง และแบบมีโครงสร้าง จะเห็นได้ว่ามีข้อแตกต่างกัน แต่ละโครงสร้างจะมีข้อดี และข้อเสียแตกต่างกัน แต่ด้วยวิวัฒนาการของการพัฒนาระบบคอมพิวเตอร์ทั้งในด้านฮาร์ดแวร์ ซอร์ฟแวร์ และระบบเครือข่าย มีความเจริญก้าวหน้าไปอย่างรวดเร็วมากในปัจจุบัน เช่น มีความรวดเร็วในการประมวลผลสูง สามารถเก็บข้อมูลได้ปริมาณมาก ๆ ระบบเครือข่ายมีการส่งข้อมูลระหว่างอุปกรณ์ต่าง ๆ ได้อย่างรวดเร็ว เป็นต้น ปัญหาต่าง ๆ ในอดีตจึงลดน้อยลงไป

แนวคิดของด้านการเขียนโปรแกรม จากที่กล่าวมาในข้างต้น จะเห็นได้ว่าในการเขียนโปรแกรม แนวทางการเขียนโปรแกรมภาษา จะถูกจัดทำเฉพาะกลุ่มเท่านั้น หลักการเขียนหรือโครงสร้างของโปรแกรมภาษาจะถูกออกแบบตามเฉพาะกลุ่มเท่านั้น ดังนั้นเมื่อมีการเปลี่ยนโปรแกรมภาษา จะต้องทำการศึกษาจากโปรแกรมภาษาใหม่ เพราะโครงสร้าง หรือการใช้คำสั่งต่าง ๆ ไม่เหมือนกันจึงทำให้เกิดปัญหาและอุปสรรคในการเขียนโปรแกรม จากปัญหาดังกล่าวในข้างต้นเขียนโปรแกรมภาษาในปัจจุบัน จึงมีแนวคิดของการที่จะเขียนโปรแกรมภาษาเป็นลักษณะของการใช้ทรัพยากรร่วมกัน เพื่อให้เกิดประโยชน์สูงสุด นั่นคือ เมื่อพัฒนาโปรแกรมภาษาหนึ่ง ก็จะสามารถนำชุดคำสั่งในโปรแกรมภาษานั่นนำมาใช้ในอีกโปรแกรมภาษาหนึ่งได้ จึงเกิดเป็นลักษณะของแนวคิดที่เรียกว่า “เชิงวัตถุ”

การเขียนโปรแกรมแบบเชิงวัตถุ (Object Oriented Programming) คือ การแปลงกระบวนการที่ได้จากการวิเคราะห์ตามหลักการเชิงวัตถุ มาดำเนินการเขียนโปรแกรม (Phillips, 2010) ซึ่งเป็นการเขียนโปรแกรมให้เข้าใกล้การดำเนินไปตามสภาพแวดล้อมที่เป็นจริงของโลก กล่าวคือ เป็นการมองทุกอย่างในการเขียนโปรแกรมเป็นดังการดำรงอยู่ของสิ่งต่าง ๆ บนโลกนี้นั่นเอง โดยทุกสิ่ง ซึ่งกล่าวได้ว่าสิ่งบนโลกนี้มักเกิดจากแม่แบบ (Class) ซึ่งแม่แบบประกอบไปด้วย

สิ่งที่จำเป็นอยู่ 2 ส่วน คือ คุณสมบัติ (Properties) และพฤติกรรม (Behavior หรือ Action) ยกตัวอย่าง เช่น ถ้ามองถึง “มนุษย์” เป็นแม่แบบ ทุก ๆ คนบนโลกนี้เกิดจากแม่แบบมนุษย์ ซึ่งวัตถุ (Object) ที่เกิดจากแม่แบบมนุษย์ หรือถ้าจะมองย้อนกลับไป แม่แบบก็เกิดจากลุ่มของวัตถุนั้นเอง ในแม่แบบมนุษย์จะประกอบไปด้วยส่วนที่เป็นคุณสมบัติ เช่น ความสูง น้ำหนัก เพศ เป็นต้น และส่วนที่เป็นพฤติกรรม เช่น วิ่ง หัวเราะ ปรบมือ เป็นต้น ซึ่งคุณสมบัติที่เกิดขึ้นจะมาจากคุณสมบัติของแม่แบบนั้นที่มี ซึ่งสามารถอธิบายลักษณะของแม่แบบและวัตถุได้ดังภาพที่ 1.7



คุณสมบัติ	พฤติกรรม
รหัสบัตร : 2131261548974 ชื่อ : สมหมาย ใจดี เพศ : ชาย อายุ : 27	วิ่ง หัวเราะ ปรบมือ ส่ายหัว

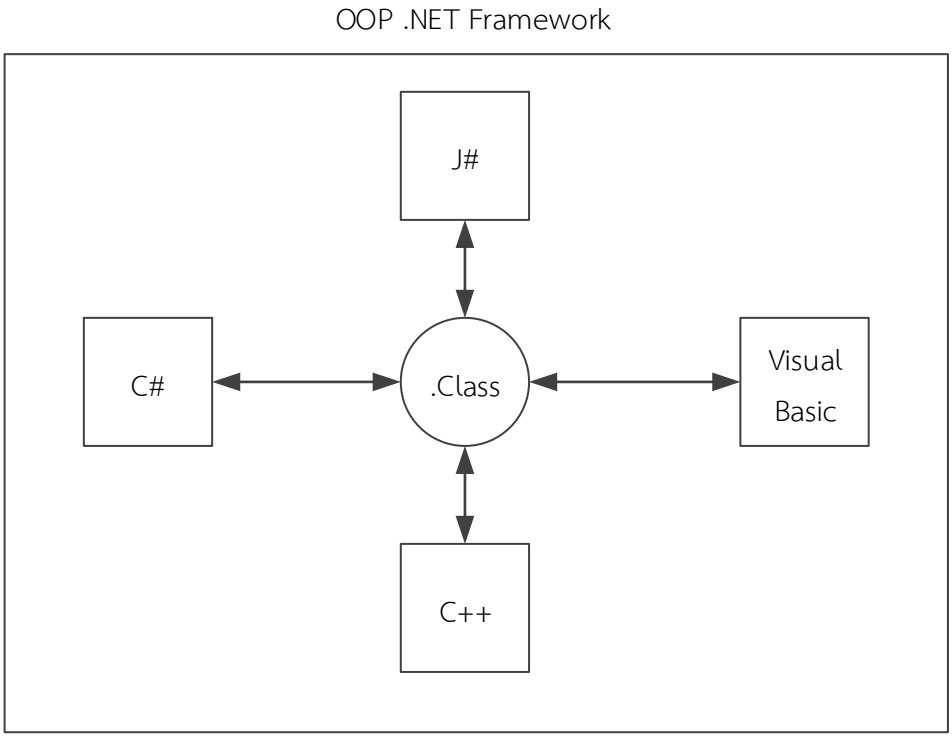
คุณสมบัติ	พฤติกรรม
รหัสบัตร : 1234569878954 ชื่อ : สมใจ โชคดี เพศ : หญิง อายุ : 24	วิ่ง หัวเราะ ปรบมือ ส่ายหัว

ภาพที่ 1.7 แม่แบบและวัตถุตามแนวคิดเชิงวัตถุ

จากภาพที่ 1.7 จะเห็นว่า วัตถุ1 และ วัตถุมีคุณสมบัติและพฤติกรรมเหมือนกับแม่แบบมนุษย์ แต่คุณสมบัติของทั้งสองวัตถุนั้น ได้ถูกกำหนดต่อมาให้แตกต่างกัน ทำให้ผลของพฤติกรรมนั้นแตกต่างกันไปตามข้อมูลของคุณสมบัติที่เปลี่ยนแปลงไป เช่น วัตถุ1 นั้นมี น้ำหนัก 48 กก. และเป็นผู้หญิง ซึ่งมีคุณสมบัติแตกต่างกับ Obj2 ที่มีความสูง 179 ซม. น้ำหนัก 65 กก. และเป็นเพศชาย กล่าวคือ ในสภาวะเป็นจริง ความแตกต่างของทั้งสองวัตถุนั้นถูกกำหนดจากข้อมูลภายในคุณสมบัติ ซึ่งทำผลของพฤติกรรมนั้นแตกต่างกัน แม้จะเกิดจากแม่แบบเดียวกันก็ตาม

คุณลักษณะพิเศษของแนวความคิดแบบเชิงวัตถุ เป็นแนวคิดที่นำแก้ไขปัญหา จากแนวคิดแบบเดิม ซึ่งสามารถจำแนกคุณลักษณะพิเศษ ได้ดังต่อไปนี้

1) คุณสมบัติด้านการใช้ทรัพยากรร่วมกัน (Sharing Resource) หมายถึง ในการพัฒนาโปรแกรมภาษาแบบเชิงวัตถุนั้น ไม่ว่าจะพัฒนาด้วยโปรแกรมภาษาใดก็ตาม ผู้พัฒนาสามารถที่จะนำเอา ชุดคำสั่งที่พัฒนาต่างโปรแกรมภาษากัน สามารถนำมาใช้งานร่วมกันได้ เช่น การใช้งานเชิงวัตถุใน .NET Framework ซึ่งสามารถอธิบายได้ ดังภาพที่ 1.8



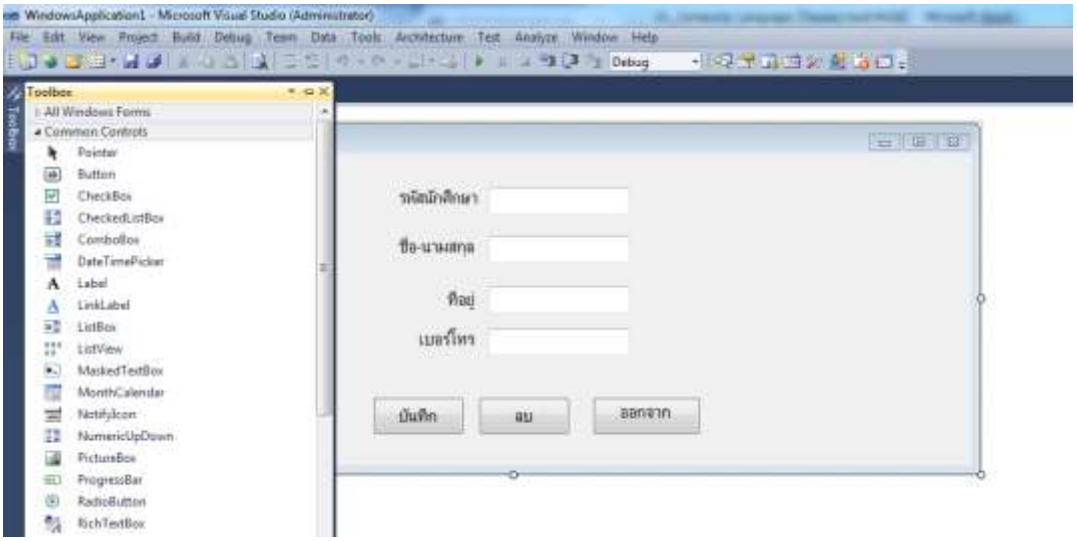
ภาพที่ 1.8 การใช้ทรัพยากรร่วมกันของกลุ่มโปรแกรมภาษาเชิงวัตถุ

การใช้ทรัพยากรร่วมกันของกลุ่มโปรแกรมภาษาเชิงวัตถุ นั่นคือ เมื่อมีการสร้างชุดคำสั่งจากโปรแกรมภาษาใดก็ตามที่เป็นโปรแกรมภาษาที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุ เมื่อมีการแปลภาษาจากตัวแปลภาษา ไฟล์ที่ได้จากการแปลภาษาจะเป็นไฟล์แบบเดียวกันนั่นคือไฟล์นามสกุล class โปรแกรมภาษาต่าง ๆ ก็สามารถนำไฟล์นี้ไปเรียกใช้งานได้ทันที ซึ่งถือว่าเป็นประโยชน์มาก เพราะเมื่อมีการเรียกใช้งานที่เป็นแบบเดิมทุก ๆ ครั้ง ไม่ต้องจำเป็นต้องมาสร้างชุดคำสั่งใหม่ ๆ

2) คุณสมบัติปรับปรุงแก้ไข (Validation Data) หมายถึง ในการปรับปรุงแก้ไข ข้อมูลต่าง ๆ ที่อยู่ภายในคลาส สามารถที่จะปรับปรุงแก้ไขได้ ซึ่งจะมีผลกระทบต่อค่าต่าง ๆ ที่อยู่ภายในคลาส ซึ่งการเปลี่ยนแปลงต่าง ๆ จะมีผลกระทบเฉพาะในคลาสของตนเอง ดังนั้นเมื่อมีข้อผิดพลาดต่าง ๆ ที่เกิดขึ้นภายในคลาส ก็จะสามารถดำเนินตรวจสอบข้อผิดพลาดได้ง่ายเพิ่มมากยิ่งขึ้น

ลักษณะของการนำหลักการแนวคิดเชิงวัตถุไปใช้ สามารถแบ่งลักษณะการใช้งานได้ดังต่อไปนี้

1) การใช้โปรแกรมเชิงวัตถุ จะเป็นลักษณะของโปรแกรมภาษาที่มีการแบ่งกลุ่มเครื่องมือต่าง ๆ ไว้ใช้งาน ซึ่งแต่ละกลุ่ม ก็คือแต่ละคลาส ซึ่งเวลาที่จะนำมาใช้งาน ก็ต้องเลือกคลาสก่อน แล้วนำไปการจัดตามหน้าต่างของโปรแกรมที่ออกแบบไว้ โปรแกรมภาษาในปัจจุบันจะมีการออกแบบเป็นลักษณะแบบนี้ ยกตัวอย่างโปรแกรมเช่น Visual Studio .NET, Dreamweaver, Delphi เป็นต้น ดังภาพที่ 1.9



ภาพที่ 1.9 การใช้โปรแกรมเชิงวัตถุ

จากภาพที่ 1.9 แสดงลักษณะที่มีการการใช้โปรแกรมเชิงวัตถ ซึ่งขั้นตอนการใช้งานเริ่มจากเลือกเครื่องมือในกลุ่มคลาส เช่น การเลือกคลาสที่เป็น TextBox ซึ่งมีหน้าที่ในการรับข้อมูล และคลาส Button จะเป็นคลาสปุ่มที่ทำหน้าที่ในการยอมรับคำสั่งแล้วนำไปประมวลผล นำคลาสเหล่านี้วางลงบนโปรแกรม สิ่งที่อยู่ในโปรแกรมจะเรียกว่าวัตถุ ซึ่งสามารถกำหนดลักษณะของวัตถุได้ เช่น ความยาว สี ขนาดตัวหนังสือ เป็นต้น

2) การเขียนโปรแกรมเชิงวัตถุ ในลักษณะมุมมองอีกแบบหนึ่ง ที่ใช้เทคนิควิธีการตามแนวความคิดของเชิงวัตถุ ในการเขียนโปรแกรม ซึ่งในปัจจุบันนี้เป็นที่นิยมมาก และแทบทุกโปรแกรมภาษาระดับสูง ได้มีการพัฒนาโปรแกรมภาษาคอมพิวเตอร์ให้สามารถรองรับการเขียนโปรแกรมเชิงวัตถุ เช่น Java C++ C# php เป็นต้น ดังตารางที่ 1.5 แสดงการเขียนโปรแกรมตามแนวคิดเชิงวัตถุด้วยภาษา php

ตารางที่ 1.5 การเขียนโปรแกรมเชิงวัตถุด้วยภาษา php เวอร์ชั่น 7

บรรทัดที่	ชุดคำสั่ง
1	<?php
2	class human
3	{
4	public $name;
5	var $sex;
6	public $height;
7	public $age;
8	public function run()
9	{
10	echo "you can runnig";
11	}
12	public function jump()
13	{ echo "you can jumping"; }
14	}
15	$fon = new human();
16	$fon -> name="ฝน";
17	$fon -> sex="หญิง";
18	$fon -> height=168;
19	$fon -> age=28;
20	$fon -> run();
21	$fon -> jump();
22	echo "ชื่อ : ".$fon->name." ";
23	echo "เพศ : ".$fon->sex." ";
24	echo "ความสูง : ".$fon->height." ";
25	echo "อายุ : ".$fon->age." ";
26	$ton = new human();
27	$ton -> name="ต้น";
28	$ton -> sex="ชาย";
29	$ton -> height=179;

```
30  $ton -> age=30;
31  echo "ชื่อ : ".$ton->name."<br>";
32  echo "เพศ : ".$ton->sex."<br>";
33  echo "ความสูง : ".$ton->height."<br>";
34  echo "น้ำหนัก : ".$ton->weight."<br>";
35  echo "อายุ : ".$ton->age."<br>";
36  $fon -> run();
37  $fon -> jump();
38  ?>
```

จากโปรแกรมตารางที่ 1.5 แม่แบบของคลาสก็คือ คลาส human ประกอบไปด้วยออบเจ็กต์ 2 ออบเจ็กต์คือ $fon และ $ton ในคลาสนี้ประกอบไปด้วยคุณสมบัติ $name, $sex, $height, $weight และ $age และประกอบไปด้วยการกระทำ run() และ jump() ซึ่งเป็นลักษณะโครงสร้างของการเขียนโปรแกรมเชิงวัตถุ

สรุป

โปรแกรมคอมพิวเตอร์ มีหน้าที่ในการสั่งให้คอมพิวเตอร์ทำงานตามที่ต้องการ โดยโปรแกรมคอมพิวเตอร์จะถูกสร้างขึ้นด้วยภาษาต่าง ๆ มีอยู่ 3 ประเภท คือ ภาษาเครื่อง ภาษาระดับกลาง และภาษาระดับสูง ซึ่งแต่ละภาษาจะมีคุณสมบัติ ข้อดีและข้อเสียแตกต่างกันออกไป โดยที่ภาษาในเครื่องหรือภาษาระดับกลาง จะมีความยากในการสั่งงานคอมพิวเตอร์ แต่จะมีความรวดเร็วในการทำงานและใช้หน่วยความจำน้อย ส่วนภาษาระดับสูงนั้นจะต้องใช้ตัวแปลภาษา ดังนั้นจะสั่งงานง่ายเพราะเป็นภาษาที่ใกล้เคียงกับภาษามนุษย์ แต่ต้องผ่านตัวแปลภาษา จึงจะมีสั่งงานให้คอมพิวเตอร์ทำงานได้ ภาษาของคอมพิวเตอร์คือ ภาษาเครื่อง ดังนั้นเมื่อโปรแกรมภาษาคอมพิวเตอร์ใดไม่ใช่ภาษาเครื่องจะต้องมีการใช้ตัวแปลภาษา เพื่อให้สามารถสั่งงานคอมพิวเตอร์ได้ โปรแกรมแรกที่ใช้ในการแปลภาษา คือ ตัวแปลภาษาแอสเซมเบลอร์ ใช้แปลภาษาเอสเซมบลี ต่อมาโปรแกรมภาษาคอมพิวเตอร์ถูกพัฒนาขึ้นมาจำนวนมาก ซึ่งแบ่งตัวแปลภาษาออกเป็น 2 กลุ่ม คือ คอมไพเลอร์และอินเทอร์พรีเตอร์ การเขียนโปรแกรมที่ถูกพัฒนาขึ้นมีโครงสร้างการทำงานของโปรแกรมแบ่งออกเป็น 3 ประเภทคือ การเขียนโปรแกรมแบบไม่มีโครงสร้าง การเขียนโปรแกรมแบบมีโครงสร้าง และการเขียนโปรแกรมแบบเชิงวัตถุ

แบบฝึกหัด

1. ให้นักผู้เรียนบอกความหมายของโปรแกรมคอมพิวเตอร์ และขั้นตอนการทำงานในการสั่งงานคอมพิวเตอร์
2. ให้นักผู้เรียนอธิบายการทำงานของระบบคอมพิวเตอร์ พร้อมอธิบายการทำงานแต่ละส่วนในระบบคอมพิวเตอร์
3. ให้นักผู้เรียนอธิบายวิวัฒนาการของภาษาคอมพิวเตอร์ พร้อมทั้งให้เหตุผลของการในการพัฒนาภาษาคอมพิวเตอร์
4. ให้นักผู้เรียนบอกข้อแตกต่างระหว่าง ภาษาเครื่อง ภาษาระดับกลาง และภาษาระดับสูง พร้อมทั้งอธิบายข้อดีและข้อเสียแต่ละภาษา
5. ให้นักผู้เรียนอธิบายหลักการทำงาน ของตัวแปลภาษาเอสเซมเบลอร์ คอมไพเลอร์ และอินเทอร์พรีเตอร์
6. ให้นักผู้เรียนอธิบายความแตกต่าง ข้อดี และข้อเสีย ของตัวแปลภาษาคอมไพเลอร์และอินเทอร์พรีเตอร์
7. ให้นักผู้เรียนยกตัวอย่างโปรแกรมที่ใช้ตัวแปลภาษาแบบคอมไพเลอร์ และอินเทอร์พรีเตอร์ พร้อมทั้งอธิบาย
8. ให้นักผู้เรียนอธิบายวิวัฒนาการรูปแบบของโปรแกรมภาษาคอมพิวเตอร์ มีทั้งหมดกี่ประเภท แต่ละประเภทมีแนวคิดอย่างไรบ้าง
9. ให้นักผู้เรียนโครงสร้างโปรแกรมแบบมีโครงสร้าง และแบบเชิงวัตถุ มีความแตกต่างกันอย่างไร พร้อมทั้งบอกข้อดีและข้อเสีย
10. ให้นักผู้เรียนบอกโครงสร้างภาษาแบบเชิงวัตถุ มีองค์ประกอบอะไรบ้าง พร้อมทั้งอธิบายความหมายและหน้าที่ในองค์ประกอบนั้น ๆ

เอกสารอ้างอิง

Phillips, Dusty. (2010). **Python 3 Object Oriented Programming**. Birmingham: Packet Publishing.

Evans, David. (2011). **Introduction to Computing Explorations in Language, Logic, and Machines**. Charlottesville: University of Virginia.

Haverbeke, Marjin. (2014). **Eloquent JavaScript A Modern Introduction to Programming**. Cambridge: Massachusetts Institute of Technology.

Watt, D. A., & Brown, D. F. (2000). **Programming language processors in Java: compilers and interpreters**. London: Pearson.

Liang, Y. Daniel. (2015). **Introduction to Java Programming**. 10rd ed. Georgia: Pearson.

ภาษาเอสเซมบลี. (2558). (ออนไลน์) (อ้างเมื่อวันที่ 10 พฤศจิกายน 2558). จาก https://th.wikipedia.org/wiki/ภาษาแอสเซมบลี)