	ใบความรู้ที่ 2	จำนวนชั่วโมง
	ชื่อวิชา โครงสร้างข้อมูลและอัลกอริทึม **รหัสวิชา** 30901-2001 (Data Structures and Algorithms)	5 ชั่วโมง
	ชื่อหน่วย อัลกอริทึมประสิทธิภาพของอัลกอริทึม	

1.จุดประสงค์การเรียนรู้

1. อธิบายความหมายของอัลกอริทึมได้
2. อธิบายขั้นตอนการพัฒนาอลักอริทึมได้
3. รู้วิธีการกำหนดขั้นตอนการจำลองความคิดเป็นคำพูดได้
4. วิเคราะห์โจทย์ปัญหาเพื่อพัฒนาเป็นอัลกอริทึมได้
5. เขียนอัลกอริทึมได้

2. เนื้อหาสาระ

2.1 ความหมายของอัลกอริทึม

Algorithm คือ กระบวนการแก้ปัญหาที่สามารถอธิบายออกมาเป็นขั้นตอนที่ชัดเจน เมื่อ นำเข้าอะไรแล้วจะต้องได้ผลลัพธ์เช่นไร กระบวนการนี้ประกอบด้วยจะประกอบด้วย วิธีการเป็น ขั้น ๆ และมีส่วนที่ต้องทำแบบวนซ้ำอีก จนกระทั่งเสร็จสิ้นการทำงาน Algorithm ไม่ใช่คำตอบแต่เป็นชุดคำสั่งที่ทำให้ได้คำตอบ

2.2 หลักการเขียนอัลกอริทึม

1. กระบวนการสำคัญเริ่มต้นที่จุดจุดเดียวในการมีจุดเริ่มต้นหลายที่จะทำให้กระบวนการวิธีสับสน จนในที่สุดอาจทำให้ผลลัพธ์ที่ได้ไม่ตรงกับความต้องการ หรืออาจทำให้อัลกอริทึมนั้นไม่สามารถทำงานได้เลย

2. กำหนดการทำงานเป็นขั้นเป็นตอนอย่างชัดเจน การกำหนดอัลกอริทึมที่ดีควรมีขั้นตอนที่ชัดเจนไม่คลุมเครือ เสร็จจากขั้นตอนหนึ่ง ไปยังขั้นตอนที่สองมีเงื่อนไขการทำงานอย่างไร ควรกำหนดให้ชัดเจน

3. การทำงานแต่ละขั้นตอนควรสั้นกระชับ เพราะการกำหนดขั้นตอนการทำงานให้สั้นกระชับนอกจาก จะทำให้โปรแกรมทำงานได้รวดเร็วแล้ว ยังเป็นประโยชน์ต่อผู้อื่นที่มาพัฒนาโปรแกรมต่อด้วยเพราะสามารถ ศึกษาอัลกอริทึมจากโปรแกรมที่เขียนไว้ได้ง่าย

4. ผลลัพธ์ในแต่ละขั้นตอนควรต่อเนื่องกัน การออกแบบขั้นตอนที่ดีนั้นผลลัพธ์จากขั้นตอนแรกควรเป็นข้อมูลสำหรับนำเข้า ให้กับข้อมูลในขั้นต่อไป ต่อเนื่องกันไปจนกระทั่งได้ผลลัพธ์ตามที่ต้องการ

5. การออกแบบอัลกอริทึมที่ดี ควรออกแบบให้ครอบคลุมการท างานในหลายรูปแบบ เช่น การออกแบบโดยคิดไว้ล่วงหน้าว่าหากผู้ใช้โปรแกรมป้อนข้อมูลเข้าผิดประเภท โปรแกรมจะมีการเตือนว่าผู้ใช้งานมี การ

ใส่ข้อมูลที่ผิดประเภทโดยโปรแกรมจะไม่รับข้อมูลนั้น เพื่อให้ใส่ข้อมูลใหม่อีกครั้ง เพื่อป้องกันการเกิด จุดบกพร่องของโปรแกรมได้

2.3 เครื่องมือช่วยในการเขียนอัลกอริทึม

การเขียนอัลกอริทึม เป็นการวางแผนเกี่ยวกับการแก้ปัญหา โดยจะอธิบายการทำงานที่ชัดเจนเพื่อเป็นแนวทางในการเขียนโปรแกรม ช่วยให้การเขียนโปรแกรมทำได้ง่ายขึ้น ช่วยให้โปรแกรมมีข้อผิดพลาดน้อยลง นอกจากนี้ยังช่วยตรวจสอบการทำงานของโปรแกรม ทำให้ทราบขั้นตอนการทำงานของโปรแกรมได้อย่างรวดเร็ว โดยไม่ต้องดูจากโปรแกรมจริงในการเขียนอัลกอริทึม มีเครื่องมือช่วยในการเขียนที่นิยมใช้ 3 แบบ คือ

2.3.1 Natural Language อธิบายแบบใช้ภาษาที่เราสื่อสารกันทั่วไป

เป็นการแสดงขั้นตอนการทำงานในลักษณะการบรรยายเป็นข้อความด้วยภาษาพูดใดๆ เช่น ภาษาไทย ภาษาอังกฤษ ภาษาเกาหลี ภาษาญี่ปุ่น หรือ ภาษาจีน เป็นต้น ขึ้นอยู่กับความถนัดของผู้เขียนอัลกอริทึม มักเขียนบรรยายขั้นตอนการทำงานเป็นข้อๆ เช่น

ตัวอย่าง การอธิบายการคำนวณพื้นที่รูปสามเหลี่ยมเหลี่ยม และแสดงผลการคำนวณด้วยการใช้ภาษา ธรรมชาติ

สุตรการคำนวณพื้นที่รูปสามเหลี่ยม = 0.5 *ฐาน*สูง

Area=0.5*base*high

อัลกอริทึมสามารถเขียนได้ดังนี้

1.เริ่มต้นการทำงาน
2.นำเข้าข้อมูลความยาวฐาน
3. นำเข้าข้อมูลความสูง
4.คำนวณ หาพื้นที่รูปสามเหลี่ยมด้วยสูตร พื้นที่=ความยาวฐาน*ความสูง
5.แสดงผลพื้นที่รูปสามเหลี่ยม
6. จบการทำงาน

2.3.2 Pseudo code อธิบายด้วยรหัสจำลองหรือรหัสเทียม

เนื่องจากขั้นตอนวิธีหรืออัลกอริทึม ที่มักเขียนอยู่ในรูปแบบของข้อความหรือภาษาพูดนั้น หากนำมาใช้เพื่อแก้ปัญหาโจทย์ทางคอมพิวเตอร์ ก็คงใช้งานได้ไม่ดีนัก เพราะสำนวนข้อความของผู้เขียนแต่ละคน มีความแตกต่างกัน บางครั้งอาจมีความกำกวม ส่งผลลัพธ์การออกแบบโปรแกรมที่ผิดพลาดได้ ดังนั้น เนื้อหาในหน่วยการเรียนนี้ จึงขอนำเสนออัลกอริทึมในรูปแบบของรหัสเทียม

รหัสเทียม (Pseudo Code อ่านว่า "ซูโดโค้ด") เป็นคำสั่งที่ไม่ขึ้นกับภาษาคอมพิวเตอร์ภาษาใดภาษาหนึ่ง แต่เป็นคำสั่งที่นำมาใช้ในการเขียนเลียนแบบชุดคำสั่งแบบย่อๆ เพื่อการออกแบบโปรแกรม หรือร่างเค้าโครงโปรแกรมที่จะเขียนขึ้นมา ก่อนที่จะนำไปแปลงเขียนเป็นชุดคำสั่งด้วยภาษาคอมพิวเตอร์จริงๆ ต่อๆไป

โดยรหัสเทียมนั้นจะเป็นการอธิบายลำดับขั้นตอนการทำงานที่ได้จากขั้นตอนและวิธีการประมวลผลแล้วเขียนแทนด้วยคำสั่งในรูปแบบภาษาอังกฤษและใช้ถ้อยคำที่ง่ายในการอธิบาย มีความกะทัดรัด ลักษณะจะมีความคล้ายคลึงโครงสร้างของภาษาโปรแกรมหรือภาษาคอมพิวเตอร์ ซึ่งจะช่วยให้ผู้อ่านเข้าใจความหมายก่อนการนำไปเขียนโปรแกรมจริง

1) หลักการเขียนรหัสเทียม

เมื่อมีการออกแบบอัลกอริทึมเพื่อแก้ไขปัญหาใดปัญหาหนึ่ง ขั้นตอนแรก ต้องเข้าใจในเบื้องต้นก่อนว่าคอมพิวเตอร์นั้นจะปฏิบัติงานตามชุดคำสั่งที่เขียนตามความเป็นจริง ดังนั้น หากมีการกำหนดถ้อยคำหรือประโยคในรูปแบบของรหัสเทียม ก็จะต้องมีความสอดคล้องกับการปฏิบัติงานทางคอมพิวเตอร์ขั้นพื้นฐาน เพื่อให้การแปลงชุดคำสั่งจากรหัสเทียมมาเป็นภาษาคอมพิวเตอร์ได้ง่ายยิ่งขึ้น

อย่างไรก็ตาม ในการเขียนรหัสเทียม ไม่มีการกำหนดมาตรฐานที่ระบุไว้ชัดเจนลงไป ว่าต้องทำอย่างนั้นต้องทำอย่างนี้ ดังนั้น ผู้เขียนรหัสเทียมแต่ละคน จึงอาจมีเทคนิคการเขียนที่แตกต่างกันออกไป แต่ทั้งนี้ เพื่อให้การเขียนรหัสเทียมทีแบบแผนยิ่งขึ้น จึงควรปฏิบัติตามหลักเกณฑ์ดังต่อไปนี้

1. กำหนดจุดเริ่มต้นด้วยคำว่า "Begin" และจุดสิ้นสุดด้วยคำว่า "End"
2. การเขียนประโยคคำสั่ง จะเขียนเป็นภาษาอังกฤษอย่างง่าย
3. การเขียนประโยคคำสั่งหนึ่งๆ จะเขียนต่อหนึ่งบรรทัดเท่านั้น
4. การเขียนรหัสเทียมต้องไม่ขึ้นกับภาษาใดภาษาหนึ่งในการเขียนโปรแกรม

5. การเขียนรหัสเทียมควรมีการย่อหน้า เพื่อสะดวกต่อการอ่านและตรวจสอบ

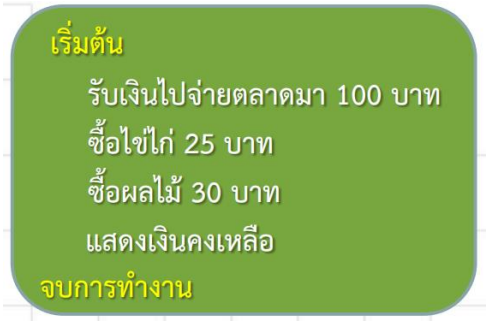
6. การเขียนรหัสเทียมจะไม่เขียนหมายเลขกำกับในแต่ละขั้นตอน

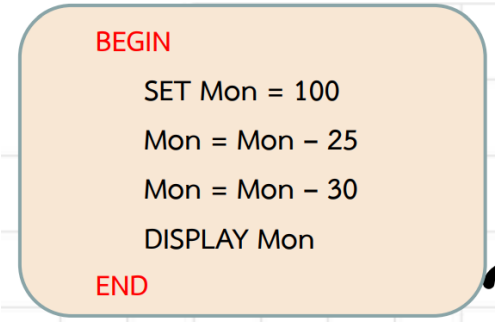
2)รูปแบบการเขียนรหัสเทียม

1. เริ่มต้นและจบท้ายขั้นตอนด้วย BEGIN...END
2. รับข้อมูลด้วย READ หรือ GET
3. แสดงผลข้อมูลด้วย PRINT, WRITE, OUTPUT, DISPLAY
4. คำนวณ + , - , * , / , ^
5. กำหนดค่าด้วย SET, =
6. เปรียบเทียบและทางเลือก IF..THEN..ELSE..ENDIF, CASE..OF..ENDCASE

7.การทำซ้ำ DO..WHILE, WHILE..DO..ENDWHILE, REPEAT..UNTIL, และ FOR.. ENDFOR

ตัวอย่างการเขียนรหัสเทียม





ที่มา http://paijit.lpru.ac.th/

2.3.3 Flowchart อธิบายด้วยแผนผัง

2.5 อัลกอริทึม

อัลกอริทึม แนวคิดอย่างมีเหตุผลที่ผู้เขียนโปรแกรมใช้ในการอธิบายวิธีการอย่างเป็นขั้นตอนตามลำดับในการที่จะพัฒนาโปรแกรมนั้น ๆ เพื่อตรวจสอบขั้นตอนต่าง ๆ ในการทำงานและความถูกต้องในแต่ละขั้นตอน

2.5.1คุณสมบัติของอัลกอริทึม

เพื่อให้ได้มาซึ่งอัลกอริทึมที่ดีสามารถนำไปใช้แก้ปัญหาได้ อัลกอริทึมจึงต้องมีคุณสมบัติพื้นฐานดังต่อไปนี้

1. อัลกอริทึมต้องไม่กำกวม อ่านแล้วเข้าใจง่าย
2. อัลกอริทึมต้องมีความถูกต้องในผลลัพธ์ที่ใช้แก้ไขปัญหาในกรณีต่าง ๆ
3. กระบวนการหรือขั้นตอนที่ระบุไว้ในอัลกอริทึมต้องมีความเรียบง่าย และเพียงพอต่อการดำเนินงานเพื่อประมวลผลในคอมพิวเตอร์ได้
4. อัลกอริทึมต้องมีจุดเริ่มต้นและจุดสิ้นสุด

2.5.2 ประโยชน์ของโครงสร้างข้อมูลและอัลกอริทึม

ประโยชน์ของโครงสร้างข้อมูลและอัลกอริทึม มีดังนี้

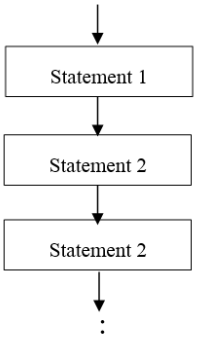
1. เพื่อจะได้เขียนอัลกอริทึมที่มีประสิทธิภาพออกมา
2. เพื่อนำโครงสร้างข้อมูลที่มีอยู่แล้วมาประยุกต์ใช้งานร่วมกับโปรแกรมคอมพิวเตอร์ที่เราเป็นผู้เขียนขึ้นมาเองได้
3. เพื่อเพิ่มประสิทธิภาพการทำงานให้กับโปรแกรมคอมพิวเตอร์
4. การออกแบบอัลกอริทึมที่ดี จะช่วยลดหน่วยความจำที่ต้องใช้ในระหว่างประมวลผลลงได้
5. อัลกอริทึมที่ดีจะช่วยให้เครื่องคอมพิวเตอร์ใช้เวลาในการประมวลผลเร็วขึ้น
6. สามารถเลือกใช้โครงสร้างข้อมูลและอัลกอริทึมได้อย่างเหมาะสม

2.5.3โครงสร้างควบคุม

อัลกอริทึมและโปรแกรมคอมพิวเตอร์ จะมีโครงสร้างควบคุม (Control Structure) การทำงานพื้นฐาน 3 รูปแบบ คือ แบบเรียงลำดับ แบบทางเลือก และแบบทำซ้ำ ซึ่งมีรายละเอียดดังนี้

1) แบบเรียงลำดับ

โครงสร้างควบคุมการทำงาน แบบเรียงลำดับ (Sequence) จะเป็นการทำงานแบบเรียงลำดับทีละขั้นตอนต่อเนื่องกันไป ตั้งแต่ขั้นตอนแรกไปจนถึงขั้นตอนสุดท้าย โดยไม่มีการข้ามขั้นตอน ดังรูป



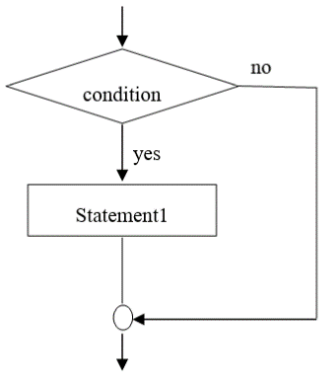
ตัวอย่างอัลกอริทึมแบบเรียงลำดับ(การทอดไข่เจียว)

1. ตอกไข่ไก่ใส่ภาชนะแล้วตีไข่
2. ปรุงรส ด้วยเครื่องปรุง
3. ตั้งกระทะบนเตา
4. เปิดแก๊ส และติดไฟ
5. ใส่น้ำมันพืช
6. นำไข่ที่ปรุงรสแล้วใส่ลงในกระทะที่ร้อน
7. ทอดจนสุก
8. ตักขึ้นใส่จานที่เตรียมไว้

2) แบบทางเลือก

โครงสร้างควบคุมการทำงาน แบบทางเลือก (Selection) จะมีการกำหนดตัวแปรขึ้นมาเพื่อสร้างเงื่อนไขในการเลือกทำงาน โดยจะมีการทดสอบค่าของตัวแปรที่ใช้เป็นเงื่อนไขเพื่อตัดสินใจว่าจะประมวลผลในส่วนใดของโปรแกรมเมื่อเงื่อนไขเป็นจริง โครงสร้างแบบนี้แบ่งออกเป็น 3 แบบ คือ

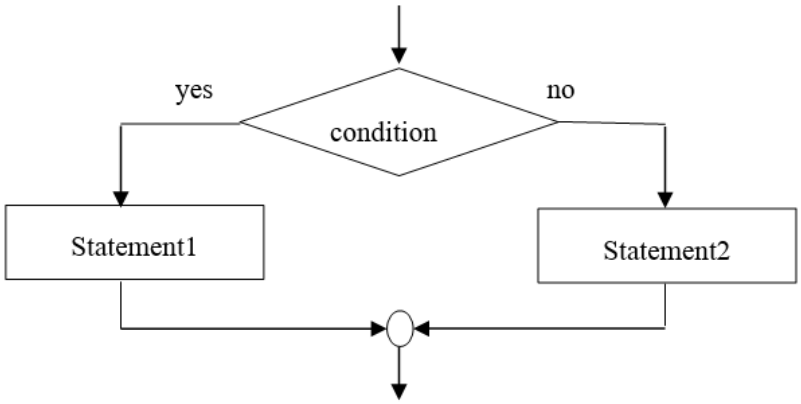
2.1 แบบมีทางเลือกเดียว (Single Alternative) การทำงานแบบนี้เมื่อเงื่อนไขเป็นจริง Statement1 จะถูกเลือกทำ แต่ถ้าเงื่อนไขเป็นเท็จก็จะข้าม Statement1 ไปทำคำสั่งถัดไป ดังรูป



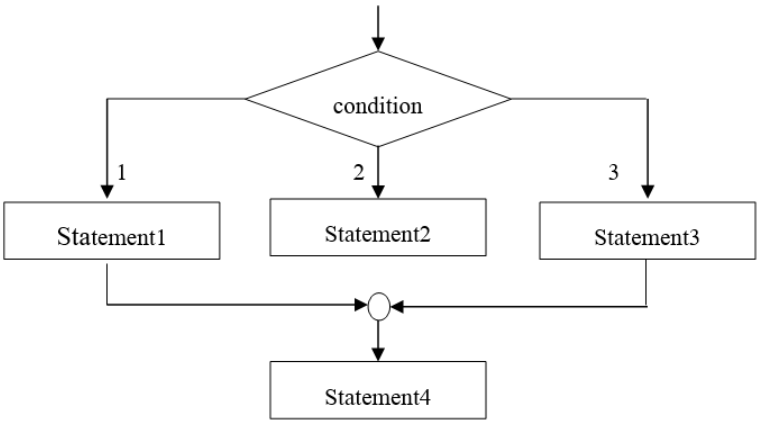
ตัวอย่างอัลกอริทึมแบบมีทางเลือกทางเดียว(ตัดเกรดวิชาคอมพิวเตอร์)

1. กรอกคะแนนสอบของนักเรียน
2. ตรวจสอบคะแนน (คะแนนที่สอบผ่าน 50 คะแนน)
3. ถ้ามากกว่า 50 คะแนน สอบผ่าน
4. ถ้าน้อยกว่า 50 คะแนน สอบตก
5. แสดงผลการสอบ

2.2. แบบมีสองทางเลือก (Double Alternative) การทำงานแบบนี้ถ้าเงื่อนไขเป็นจริง Statement1 จะถูกเลือกทำ แต่ถ้าเงื่อนไขเป็นเท็จ Statement2 จะถูกเลือกทำ เมื่อทำแต่ละทางเลือกเสร็จแล้วก็จะไปทำคำสั่งถัดลงไป

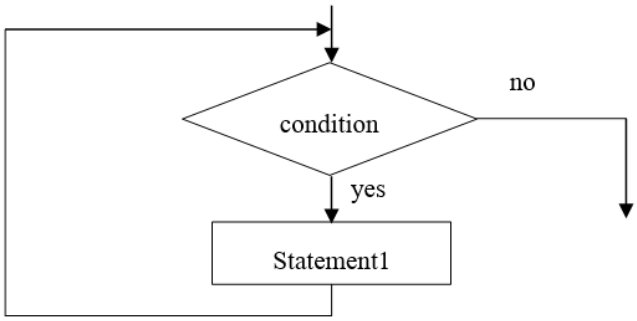


2.3. แบบมีมากกว่าสองทางเลือก (Multiple Alternative) การทำงานแบบนี้เป็นการให้เลือกทำงานจาก Statement ต่างๆ เพียง Statement เดียวที่อยู่ในเงื่อนไขที่เป็นจริง หากเงื่อนไขทั้งหมดไม่เป็นจริง ก็จะออกไปทำที่ Statement4 ดังรูป



3. แบบทำซ้ำ

โครงสร้างควบคุมการทำงาน แบบทำซ้ำ (Repetition) จะต้องประกาศตัวแปรขึ้นมาเพื่อใช้ทดสอบเงื่อนไขในการทำซ้ำ โดยลักษณะการทำงานจะมีการวนทำงานซ้ำบริเวณใดบริเวณหนึ่งของโปรแกรมไปเรื่อยๆ ตามเงื่อนไขที่กำหนด เมื่อวนครบตามเงื่อนไขแล้วก็จะหลุดออกจากวงจร(ลูป) โครงสร้างแบบนี้จำแนกออกเป็น 2 ลักษณะ คือ DOWHILE ลูป และ REPEAT...UNTIL ลูป



อัลกอริทึมการซื้อส้มเขียวหวาน กิโลกรัม

1. หยิบมังคุดมาเลือก โดยกดที่เปลือกที่นิ่มๆ
2. นำไปตรวจสอบเงื่อนไข (น้อยกว่า 1 กิโลกรัม)
3. ถ้าจริง เลือกมังคุดต่อ
4. ถ้าเท็จ หยุดเลือก
5. ชำระเงิน