

อัลกอริทึม

ตัวชี้วัด

- ออกแบบอัลกอริทึมที่ใช้แนวคิดนามธรรมเพื่อแก้ปัญหาหรืออธิบายการทำงานที่พบในชีวิตจริง
- ออกแบบและเขียนโปรแกรมอย่างง่ายเพื่อแก้ปัญหาทางคณิตศาสตร์ หรือวิทยาศาสตร์

สาระสำคัญ

แนวคิดเชิงนามธรรม เป็นการประเมินความสำคัญของรายละเอียดของปัญหา แยกแยะส่วนที่เป็นสาระสำคัญออกจากส่วนที่ไม่ใช่สาระสำคัญ คอมพิวเตอร์อัลกอริทึม เป็นแก่นของวิทยาการคอมพิวเตอร์ เป็นศาสตร์ที่ทำให้สามารถประมวลผลแบบทีละขั้นตอน ซึ่งทำให้คอมพิวเตอร์สามารถประมวลผลเพื่อแก้ไขปัญหาด้วยเครื่องคอมพิวเตอร์ การออกแบบอัลกอริทึม เพื่อแก้ปัญหาทางคณิตศาสตร์ วิทยาศาสตร์ อย่างง่าย อาจใช้แนวคิดเชิงนามธรรมในการออกแบบเพื่อให้การแก้ปัญหามีประสิทธิภาพ

ประกอบไปด้วย 5 เรื่อง ดังนี้

1. แนวคิดเชิงนามธรรม
2. อัลกอริทึมเบื้องต้น
3. การเขียนอัลกอริทึมด้วยภาษาธรรมชาติ (Natural language)
4. การเขียนอัลกอริทึมด้วยรหัสจำลอง (Pseudocode)
5. การเขียนอัลกอริทึมด้วยผังงาน (Flowchart)

1. แนวคิดเชิงนามธรรม

แนวคิดเชิงนามธรรม

ในชีวิตประจำวันเรามักเจอปัญหาต่างๆ ที่ต้องแก้ไข ซึ่งการแก้ไขปัญหาใดๆ นั้น อาจมีวิธีการแก้ไขปัญหาได้หลากหลายวิธี โดยจะต้องวิเคราะห์ปัญหานั้นอย่างรอบคอบและถี่ถ้วน เพื่อใช้ในการพิจารณาเลือกวิธีการแก้ไขปัญหานั้นได้อย่างดีที่สุดและเกิดประสิทธิภาพสูงสุด

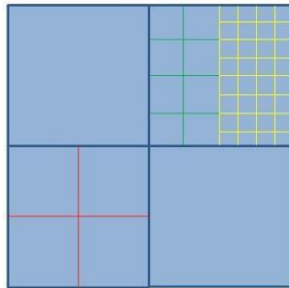
1.1 แนวคิดเชิงนามธรรม

แนวคิดเชิงนามธรรม (abstract thinking หรือ abstraction) เป็นองค์ประกอบหนึ่งของแนวคิดเชิงคำนวณ (computational thinking) ซึ่งใช้กระบวนการคัดแยกคุณลักษณะที่สำคัญออกจากรายละเอียดปลีกย่อยในปัญหา หรืองานที่กำลังพิจารณา เพื่อให้ได้ข้อมูลที่จำเป็นและเพียงพอในการแก้ปัญหา

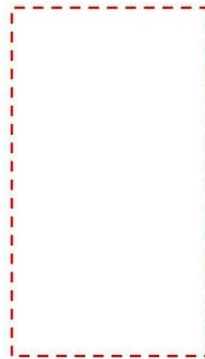
ในการแก้ปัญหานี้ อาจมีวิธีการแก้ไขได้หลายวิธี ขึ้นอยู่กับการมองปัญหา การมองเห็นรายละเอียดเป้าหมายของโจทย์ปัญหา และประสบการณ์ของผู้แก้ปัญหา

แนวคิดเชิงนามธรรม

แนวคิดเชิงนามธรรม (abstraction) : เป็นเครื่องมือในการแก้ปัญหาซึ่งเป็นองค์ประกอบของแนวคิดเชิงคำนวณ โดยมีแนวคิดในการแก้ไขปัญหาย่อยเป็นระบบ เพื่อย่อยสถานการณ์และสามารถกำหนดขั้นตอนการแก้ปัญหาได้อย่างชัดเจน



พิจารณาเฉพาะส่วนที่เป็นสาระสำคัญ



โดยแยกส่วนที่ไม่ใช่สาระสำคัญออกไป

ส่วนที่เป็นสาระสำคัญ

หากทำการแยกสี่เหลี่ยมจัตุรัสรูปใหญ่ออกเป็น 4 ส่วนแล้วจะได้รูปแบบทั้งหมดที่มี หากพิจารณาเฉพาะส่วนที่เป็นสาระสำคัญ โดยแยกส่วนที่ไม่ใช่สาระสำคัญออกไป จะเห็นว่าส่วนที่เป็นสาระสำคัญก็คือ รูปสี่เหลี่ยม

1.2 การใช้แนวคิดเชิงนามธรรมเพื่อแก้ปัญหา

ปัญหาที่กำลังพิจารณาอยู่นั้นอาจประกอบไปด้วยรายละเอียดจำนวนมาก ทั้งที่จำเป็นและไม่จำเป็นต่อการแก้ปัญหา

- ใช้แนวคิดเชิงนามธรรมในการแก้ปัญหาทางคณิตศาสตร์

หากนำแนวคิดเชิงนามธรรมมาประยุกต์ใช้กับรูปทรงเรขาคณิต เช่น หากต้องการหาแนวคิดเชิงนามธรรมของสี่เหลี่ยมมุมฉาก สามารถกระทำได้โดยพิจารณาเฉพาะสาระสำคัญ และจัดรวมของสี่เหลี่ยมของมุมฉาก

จากข้อมูลจะเห็นว่า เมื่อไม่พิจารณาสิ่งที่ไม่จำเป็น เช่น ความยาวของแต่ละด้าน ดังนั้นแนวคิดเชิงนามธรรมของสี่เหลี่ยมมุมฉาก คือ สี่เหลี่ยมใดๆ ที่ประกอบด้วยด้าน 4 ด้าน และทุกมุมของสี่เหลี่ยมเป็นมุมฉาก

ใช้แนวคิดเชิงนามธรรมในการแก้ปัญหาทางคณิตศาสตร์

▶ หากต้องการหาแนวคิดเชิงนามธรรมของสี่เหลี่ยมมุมฉาก สามารถกระทำได้ โดยพิจารณาเฉพาะสาระสำคัญ และจุดรวมของสี่เหลี่ยมมุมฉาก

สี่เหลี่ยมจัตุรัส



- ประกอบด้วยด้าน 4 ด้าน
- ทุกมุมเป็นมุมฉาก

สี่เหลี่ยมผืนผ้า



- ประกอบด้วยด้าน 4 ด้าน
- ทุกมุมเป็นมุมฉาก

จากข้อมูลข้างต้น เมื่อไม่พิจารณาสิ่งที่ไม่จำเป็น เช่น ความยาวของแต่ละด้าน เป็นต้น
ดังนั้น แนวคิดเชิงนามธรรมของสี่เหลี่ยมมุมฉาก คือ สี่เหลี่ยมใด ๆ ที่ประกอบด้วยด้าน 4 ด้าน และทุกมุมของสี่เหลี่ยมเป็นมุมฉาก

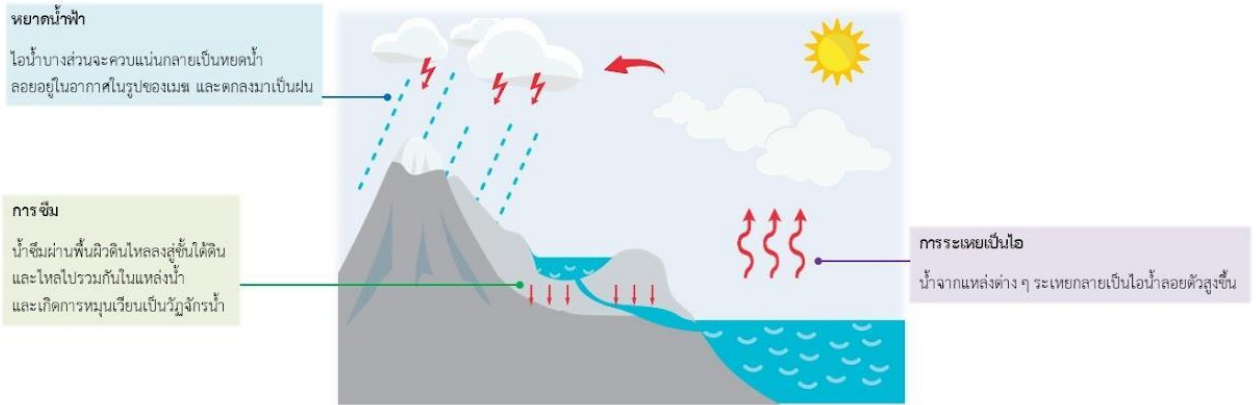
▪ ใช้แนวคิดเชิงนามธรรมในการแก้ปัญหาทางวิทยาศาสตร์

เมื่อแสงอาทิตย์ส่องลงมายังพื้นโลก จะทำให้น้ำจากแหล่งต่างๆ ระเหยกลายเป็นไอลอยตัวสูงขึ้นรวมทั้งไอน้ำที่เกิดจากการคายน้ำของพืช จากการหายใจของสิ่งมีชีวิต หรือของเหลวจากสิ่งขับถ่าย เช่น เหงื่อ ปัสสาวะ เป็นต้น ซึ่งไอน้ำดังกล่าวจะลอยตัวขึ้นสู่บรรยากาศ โดยไอน้ำที่รวมตัวกันนี้จะมีลักษณะร้อนขึ้น เมื่อลอยตัวสูงขึ้นไปกระทบกับอากาศเย็นด้านบน ไอน้ำบางส่วนจะควบแน่นกลายเป็นหยดน้ำขนาดเล็กลอยอยู่ในอากาศในรูปแบบของเมฆ ถ้าหยดน้ำมีขนาดใหญ่จนอากาศรับน้ำหนักไม่ได้ ก็จะตกลงมาเป็นฝนและถูกกักเก็บไว้ตามแหล่งน้ำต่างๆ ซึ่งเกิดการหมุนเวียนในลักษณะดังที่กล่าวมาข้างต้นนี้ต่อเนื่องกันไป

ใช้แนวคิดเชิงนามธรรมในการแก้ปัญหาทางวิทยาศาสตร์

▶ เมื่อแสงอาทิตย์ส่องมายังพื้นโลก จะทำให้น้ำจากแหล่งต่างๆ ระเหยกลายเป็นไอลอยตัวสูงขึ้นสู่บรรยากาศและความชื้นกลายเป็นหยดน้ำขนาดเล็กลอยอยู่ในอากาศในรูปของเมฆ
ถ้าหยดน้ำมีขนาดใหญ่จนอากาศรับน้ำหนักไม่ได้ ก็จะตกลงมาเป็นฝนและถูกกักเก็บไว้ตามแหล่งน้ำต่างๆ

เมื่อต้องการหาแนวคิดเชิงนามธรรมของวัฏจักรน้ำจากข้อมูลข้างต้น โดยพิจารณาเฉพาะสาระสำคัญจะต้องประกอบสำคัญ ดังนี้



2. อัลกอริทึมเบื้องต้น

อัลกอริทึมเบื้องต้น

อัลกอริทึม (Algorithm) หมายถึง ขั้นตอนหรือลำดับการประมวลผลในการแก้ปัญหาใดปัญหาหนึ่งซึ่งจะช่วยให้ผู้พัฒนาโปรแกรมเห็นขั้นตอนการเขียนโปรแกรมอย่างง่ายขึ้น

อัลกอริทึม (Algorithm) หมายถึง แนวคิดอย่างมีเหตุมีผลที่ผู้พัฒนาโปรแกรมโปรแกรมเมอร์ หรือนักวิเคราะห์ระบบ ใช้ในการอธิบายวิธีการทำงานอย่างเป็นขั้นตอน ลำดับในการที่จะพัฒนาโปรแกรมนั้นๆ ให้กับผู้ที่สนใจหรือผู้ที่เป็นเจ้าของงาน หรือผู้ที่รับผิดชอบได้ทราบถึงขั้นตอนต่างๆ ในการเขียนหรือพัฒนาโปรแกรม

ขณะเดียวกันสามารถช่วยให้ผู้ที่เป็นเจ้าของงาน หรือผู้ที่รับผิดชอบได้ตรวจสอบขั้นตอนต่างๆ ในการทำงาน และความถูกต้องในแต่ละขั้นตอนการทำงาน โดยผู้ที่เป็นเจ้าของงานหรือผู้ที่รับผิดชอบนั้นๆ ไม่จำเป็นต้องเขียนโปรแกรมเป็น

รูปแบบการเขียนอัลกอริทึม

การเขียนอัลกอริทึมมีหลายรูปแบบ โดยผู้เขียนสามารถใช้อัลกอริทึมหลายรูปแบบประกอบกันในการออกแบบอัลกอริทึมนั้นเพื่อใช้ในการแก้ปัญหการเขียนโปรแกรมได้
1.แบบลำดับ (Sequential) มีลักษณะการทำงานจะเป็นไปตามขั้นตอน ก่อน-หลัง ต่อเนื่องกันไปเป็นลำดับ โดยการทำงานแต่ละขั้นตอนต้องทำให้เสร็จก่อน แล้วจึงไปทำขั้นตอนต่อไป

ตัวอย่างเช่น อัลกอริทึมแบบลำดับ การทอดไข่เจียว

1. เริ่มต้น
2. หยิบไข่ไก่
3. ตอกไข่ไก่ใส่ภาชนะ

4. ปรงรส ด้วยเครื่องปรง
5. ตีไข่ด้วยช้อนส้อม
6. ตั้งกระทะบนเตา
7. เปิดแก๊ส และติดไฟ
8. ใส่ น้ำมันพืช
9. นำไข่ที่ปรงรสแล้วใส่ลงในกระทะที่ร้อน
10. ทอดจนสุก
11. ตักขึ้นใส่จานที่เตรียมไว้
12. จบการทำงาน

2.แบบทางเลือก (Decision) อัลกอริทึมรูปแบบนี้ มีเงื่อนไขเป็นตัวกำหนดเส้นทางการทำงานของกระบวนการแก้ปัญหา โดยตัวเลือกนั้นอาจจะมีตั้งแต่ 2 ตัวขึ้นไป เช่น สอบข้อเขียน คะแนนเต็ม 50 ได้คะแนน 30 สอบผ่าน ถ้าต่ำกว่า 30 สอบไม่ผ่าน

ตัวอย่างการเขียนอัลกอริทึมแบบทางเลือก อัลกอริทึมตัดเกรดวิชาคอมพิวเตอร์

1. เริ่มต้น
2. คะแนนสอบของนักเรียน
3. ตรวจสอบคะแนน (คะแนนที่สอบผ่าน 50 คะแนน)
4. ถ้ามากกว่า 50 คะแนน สอบผ่าน
5. ถ้าน้อยกว่า 50 คะแนน สอบตก
6. ประกาศผล
7. จบการทำงาน

3.แบบทำซ้ำ (Repetition) อัลกอริทึมแบบนี้คล้ายกับแบบทางเลือก คือ มีการตรวจสอบเงื่อนไข แต่แตกต่างกันตรงที่เมื่อการทำงานตรงตามเงื่อนไขที่กำหนด โปรแกรมจะกลับไปทำงานอีกครั้งวนการทำงานแบบนี้เรื่อย ๆ จนกระทั่งไม่ตรงกับเงื่อนไขที่กำหนดไว้จึงหยุดการทำงานหรือทำงานในขั้นต่อไป

ตัวอย่างการเขียนอัลกอริทึมแบบทำซ้ำ อัลกอริทึมการชั่งมังคุด 1 กิโลกรัม

1. เริ่มต้น
2. หยิบถุงพลาสติก
3. หยิบมังคุดมาเลือก โดยกดที่เปลือกที่นิ่มๆ
4. ตรวจสอบเงื่อนไข (น้อยกว่า 1 กิโลกรัม)
5. ถ้าจริง เลือกมังคุดต่อ
6. ถ้าเท็จ หยุดเลือก
7. จ่ายเงินให้กับผู้ขาย
8. จบการทำงาน

รูปแบบการเขียนอัลกอริทึมสามารถแบ่งได้ 3 รูปแบบ ดังนี้

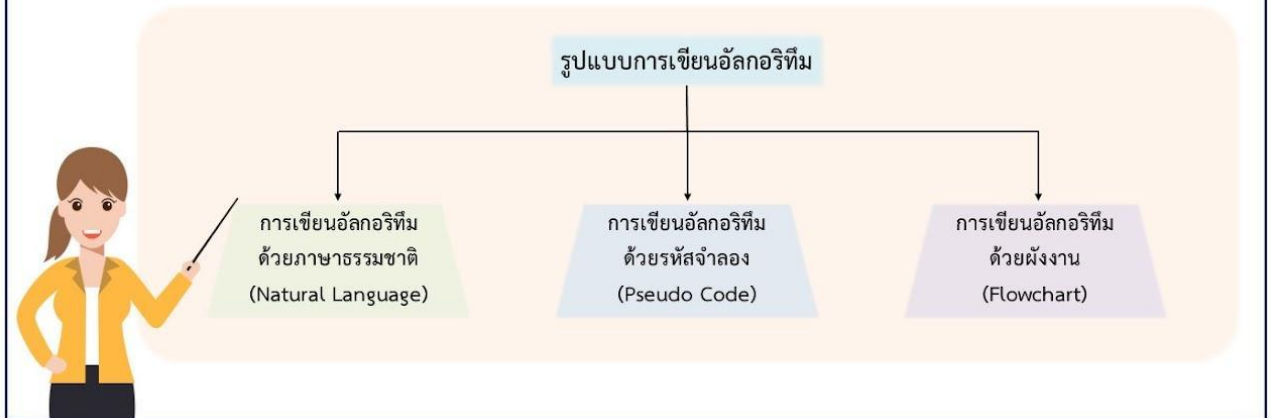
1. การเขียนอัลกอริทึมด้วยภาษาธรรมชาติ
2. การเขียนอัลกอริทึมด้วยรหัสจำลอง

3. การเขียนอัลกอริทึมด้วยผังงาน

อัลกอริทึมเบื้องต้น

อัลกอริทึม (algorithm) : ระเบียบวิธีหรือขั้นตอนวิธีที่ดำเนินการได้ด้วยคอมพิวเตอร์เพื่อใช้ในการแก้ไขปัญหา

โดยเป็นกระบวนการแก้ไขปัญหาที่สามารถอธิบายเป็นขั้นตอนที่ชัดเจน ซึ่งสามารถแบ่งได้ 3 รูปแบบ ดังนี้



ในการแก้ปัญหาต่างๆ อาจมีวิธีการแก้ปัญหามากหลายวิธี ซึ่งเราสามารถเลือกขั้นตอนวิธีการแก้ปัญหาที่แตกต่างกันออกไป โดยขั้นสุดท้ายผลลัพธ์ที่ได้ อาจเหมือนหรือแตกต่างกัน ซึ่งการแก้ปัญหาแบบอัลกอริทึมนั้น จะเป็นการแก้ที่มีลำดับหรือวิธีการแก้ไขปัญหานี้อย่างเป็นขั้นตอนและชัดเจน

ตัวอย่าง การต่อโมเดลเลโก้ (Lego)

อัลกอริทึมที่ 1

- แคะกล่อง
- นำชิ้นส่วนออกจากกล่อง
- ทดลองประกอบชิ้นส่วนเข้าด้วยกัน
- ได้ผลลัพธ์เป็นโมเดลเลโก้ตามรูปแบบหน้ากล่อง

อัลกอริทึมที่ 2

- แคะกล่อง
- นำชิ้นส่วนออกจากกล่อง
- เปิดคู่มือ อ่านวิธีการประกอบโมเดล
- ประกอบโมเดลตามวิธีการที่คู่มือระบุ
- ได้ผลลัพธ์เป็นโมเดลเลโก้ตามรูปแบบหน้ากล่อง

3. การเขียนอัลกอริทึมด้วยภาษาธรรมชาติ (Natural language)

การเขียนอัลกอริทึมด้วยภาษาธรรมชาติ (Natural language)

ภาษาธรรมชาติกับการประยุกต์ใช้ภาษาไทยบน คอมพิวเตอร์ เป็นนวัตวิทยาการด้านวิศวกรรมคอมพิวเตอร์และเทคโนโลยีด้านการประมวลผลภาษา ธรรมชาติมาพัฒนาโปรแกรมประมวลผลภาษาไทยบนคอมพิวเตอร์เพื่อให้ใช้งานได้อย่างมีประสิทธิภาพ

ประกอบด้วย การประมวลผลตัวอักษร (Character) คำ (Word) ข้อความ (Text) ภาพ (Image) และความรู้ด้านภาษาศาสตร์ (Linguistics) ภาษาที่มนุษย์ใช้ติดต่อสื่อสารในชีวิตประจำวัน เช่น ภาษาไทย ภาษาอังกฤษ หรือภาษาจีน ต่างเรียกว่า “ภาษาธรรมชาติ” (Natural Language) เพราะมีการศึกษา ได้ยิน ได้ฟัง

โครงข่ายประสาทเทียม (Artificial Neural Network) การสร้างคอมพิวเตอร์ที่จำลองเอาวิธีการทำงานของสมองมนุษย์ หรือทำให้คอมพิวเตอร์รู้จักคิดและจดจำในแนวเดียวกับโครงข่ายประสาทของมนุษย์ เพื่อช่วยให้คอมพิวเตอร์ฟังภาษามนุษย์ได้ เข้าใจ อ่านออก และรู้จำได้ ซึ่งอาจเรียกได้ว่าเป็น สมองกล

3.1 ภาษาคอมพิวเตอร์

การใช้งานคอมพิวเตอร์ ซึ่งเป็นเครื่องมือทางอิเล็กทรอนิกส์ให้ทำงานตามที่ต้องการ จำเป็นต้องมีการกำหนดภาษา สำหรับใช้ติดต่อสั่งงานกับคอมพิวเตอร์ ภาษาคอมพิวเตอร์จะเป็น “ภาษาประดิษฐ์” (Artificial Language) ที่มนุษย์คิดสร้างมาเอง เป็นภาษาที่มีจุดมุ่งหมายเฉพาะ มีกฎเกณฑ์ที่ตายตัวและจำกัด คืออยู่ในกรอบให้ใช้คำ และไวยากรณ์ที่กำหนดและมีการตีความหมายที่ชัดเจน จึงจัดภาษาคอมพิวเตอร์เป็น ภาษาที่มีรูปแบบเป็นทางการ (Formal Language) ต่างกับภาษาธรรมชาติที่มีขอบเขตกว้างมาก ไม่มีรูปแบบตายตัวที่แน่นอน กฎเกณฑ์ของภาษาจะขึ้นกับหลักไวยากรณ์และการยอมรับของกลุ่มผู้ใช้นั้น ๆ

ภาษาคอมพิวเตอร์สามารถแบ่งออกได้เป็น 5 ยุค ดังนี้

1. ยุคของภาษาเครื่อง (Machine language)
2. ยุคของภาษาแอสเซมบลี (Assembly language)
3. ยุคของภาษาชั้นสูง (High level language)
4. ยุคของภาษาชั้นสูงมาก (Very high level language)
5. ยุคของภาษาธรรมชาติ (National language)

ระดับของภาษา

การใช้งานคอมพิวเตอร์ซึ่งเป็นเครื่องมือทางอิเล็กทรอนิกส์ให้ทำงานตามที่ต้องการ จำเป็นต้องมีการกำหนดภาษาสำหรับใช้ติดต่อสั่งงานกับคอมพิวเตอร์ ภาษาคอมพิวเตอร์ สามารถแบ่งได้ 3 ระดับ ได้แก่

1. ภาษาระดับต่ำ

เป็นภาษาที่มนุษย์ทำความเข้าใจยาก เหมาะแก่การเขียนโปรแกรมควบคุมฮาร์ดแวร์ ซึ่งจะทำงานได้โดยตรงและรวดเร็ว แต่ไม่เหมาะแก่การใช้พัฒนาโปรแกรม ตัวอย่าง เช่น ภาษาเครื่อง ภาษาแอสเซมบลี

2. ภาษาระดับกลาง

เป็นภาษาที่มีลักษณะแบบโครงสร้าง สามารถใช้บนเครื่องที่มีความเร็วแตกต่างกันโดยไม่ต้องดัดแปลง ภาษาระดับกลางเป็นการนำข้อดีของภาษาระดับต่ำกับภาษาระดับสูงมาพัฒนา ดังนั้นภาษาระดับกลางจึงเป็นที่นิยมใช้กันอย่างแพร่หลาย

3. ภาษาระดับสูง

เป็นภาษาที่เข้าใจได้ง่ายมีการใช้คำสั่งเป็นภาษาอังกฤษ ซึ่งมีลักษณะใกล้เคียงกับภาษามนุษย์มาก การสั่งให้คอมพิวเตอร์ทำงานจะต้องมีการแปลความหมายของคำสั่ง

ภาษาคอมพิวเตอร์อาจแบ่งได้เป็น 3 ระดับ คือ ภาษาเครื่อง (Machine Language) ภาษาระดับต่ำ (Low Level Language) และภาษาระดับสูง (High Level Language)

1 ภาษาเครื่อง (Machine Language)

การ เขียนโปรแกรมเพื่อสั่งให้คอมพิวเตอร์ทำงานในยุคแรก ๆ จะต้องเขียนด้วยภาษา ซึ่งเป็นที่ยอมรับของเครื่องคอมพิวเตอร์ที่เรียกว่า “ภาษาเครื่อง” ภาษานี้ประกอบด้วย ตัวเลขล้วน ทำให้เครื่องคอมพิวเตอร์สามารถทำงานได้ทันที ผู้ที่จะเขียนโปรแกรม ภาษาเครื่องได้ ต้องสามารถจำรหัสแทนคำสั่งต่าง ๆ ได้ และในการคำนวณต้องสามารถจำได้ว่าจำนวนต่าง ๆ ที่ใช้ในการคำนวณนั้นถูกเก็บไว้ที่ตำแหน่งใด ดังนั้น โอกาสที่จะเกิดความผิดพลาดในการเขียนโปรแกรมจึงมีมาก นอกจากนี้เครื่องคอมพิวเตอร์แต่ละระบบมีภาษาเครื่องที่แตกต่างกันออก ทำให้เกิดความไม่สะดวกเมื่อมีการเปลี่ยนเครื่องคอมพิวเตอร์เพราะจะต้องเขียน โปรแกรมใหม่ทั้งหมด

2 ภาษาระดับต่ำ (Low Level Language)

เนื่องจากภาษาเครื่องเป็นภาษาที่มีความยุ่งยากในการเขียนดังได้กล่าวมาแล้ว จึงไม่มีผู้นิยมและมีการใช้น้อย ดังนั้นได้มีการพัฒนาภาษาคอมพิวเตอร์ขึ้นอีกระดับหนึ่ง โดยการใช้ตัวอักษรภาษาอังกฤษเป็นรหัสแทนการทำงาน การใช้และการตั้งชื่อตัวแปร แทนตำแหน่งที่ใช้เก็บจำนวนต่าง ๆ ซึ่งเป็นค่าของตัวแปรนั้น ๆ การใช้สัญลักษณ์ช่วยให้การเขียนโปรแกรมนี้นี้เรียกว่า “ภาษาระดับต่ำ” ภาษาระดับต่ำเป็นภาษาที่มีความหมายใกล้เคียงกับภาษาเครื่อง มากบางครั้งจึงเรียกภาษานี้ว่า “ภาษาอิงเครื่อง” (Machine – Oriented Language) ตัวอย่างของภาษาระดับต่ำ ได้แก่ ภาษาแอสเซมบลี เป็นภาษาที่ใช้คำในอักษรภาษาอังกฤษเป็นคำสั่งให้เครื่องทำงาน เช่น ADD หมายถึง บวก SUB หมายถึง ลบ เป็นต้น การใช้คำเหล่านี้ช่วยให้การเขียนโปรแกรมง่ายขึ้นกว่าการใช้ภาษาเครื่องซึ่งเป็นตัวเลขล้วน ดังตารางแสดงตัวอย่างของภาษาระดับต่ำและภาษาเครื่องที่สั่งให้มีการบวกจำนวน ที่เก็บอยู่ในหน่วยความจำ

ตาราง แสดงความสัมพันธ์ของคำสั่งในภาษาระดับต่ำและภาษาเครื่อง

ภาษาระดับต่ำ	ภาษาเครื่อง	รหัสเลขฐานสิบหก
MOV AL,05	10110000 00000101	B0 05
MOV BL,08	10110011 00001000	B3 08
ADD AL,BL	00000000 11011000	00 D8
MOV CL,AL	10001000 11000001	88 C1

จาก ตารางบรรทัดแรก 10110000 00000101 เป็นคำสั่งให้นำจำนวน 5 (หรือเขียนในรูปของเลขฐานสองเป็น 00000101) ไปเก็บในรีจิสเตอร์ชื่อ AL โดยส่วนแรก 10110000 คือรหัสคำสั่ง MOV ซึ่งเป็นการเคลื่อนย้ายข้อมูลจำนวนมาเก็บไว้ในรีจิสเตอร์ AL

บรรทัดที่ สอง 10110011 00001000 เป็นคำสั่งให้นำจำนวน 8 (หรือเขียนในรูปของเลขฐานสองเป็น 00001000) ไปเก็บในรีจิสเตอร์ชื่อ BL โดยส่วนแรก 10110011 คือรหัสคำสั่ง MOV ซึ่งเป็นการเคลื่อนย้ายข้อมูลจำนวนมาเก็บไว้ในรีจิสเตอร์ BL

บรรทัดที่สาม เป็นคำสั่งการบวกระหว่างรีจิสเตอร์ AL กับ BL หรือนำ 5 บวก 8 ผลลัพธ์เก็บในรีจิสเตอร์ AL

บรรทัดที่สี่ เป็นการนำผลลัพธ์จากรีจิสเตอร์ชื่อ AL ไปเก็บไว้ในรีจิสเตอร์ชื่อ CL การ ใช้โปรแกรมที่เขียนด้วยภาษาแอสเซมบลีนั้น เครื่องคอมพิวเตอร์ไม่สามารถทำงานได้ทันที จำเป็นต้องมีการแปลโปรแกรมในการแปลที่มีชื่อว่า “แอสเซมเบลอร์” (Assembler) ซึ่งแตกต่างไปตามเครื่องคอมพิวเตอร์แต่ละชนิด ดังนั้นแอสเซมเบลอร์ของเครื่องชนิดหนึ่งจะไม่สามารถใช้แปลโปรแกรมภาษาแอสเซมบลีของเครื่องชนิดอื่น ๆ ได้ภาษาแอสเซมบลีนี้ยังคงใช้ยาก เพราะผู้เขียนโปรแกรมจะต้องเข้าใจในการทำงานของเครื่องคอมพิวเตอร์อย่าง ละเอียด ต้องรู้ว่าจำนวนที่จะนำมาคำนวณนั้นอยู่ ณ ตำแหน่งใดในหน่วยความจำในทำนองเดียวกับการเขียนโปรแกรมเป็นภาษาเครื่อง ภาษาแอสเซมบลีจึงมีผู้ใช้น้อย และมักจะใช้ในกรณีที่ต้องการควบคุมการทำงานภายในของตัวเครื่องคอมพิวเตอร์

3 ภาษาระดับสูง (High Level Language)

ภาษา ระดับสูงเป็นภาษาที่สร้างขึ้นเพื่อช่วยอำนวยความสะดวกในการเขียนโปรแกรม กล่าวคือลักษณะของคำสั่งจะประกอบด้วยคำต่าง ๆ ในภาษาอังกฤษ ซึ่งผู้อ่านสามารถเข้าใจความหมายได้ทันที ผู้เขียนโปรแกรมจึงเขียนโปรแกรมด้วยภาษาระดับสูงได้ง่ายกว่าเขียนด้วยภาษาแอสเซมบลีหรือภาษาเครื่อง ภาษาระดับสูงมีมากมายหลายภาษา อาทิเช่น ภาษาฟอร์แทรน (FORTRAN) ภาษาโคบอล (COBOL) ภาษาปาสคาล (Pascal) ภาษาเบสิก(BASIC) ภาษาวิซวลเบสิก (Visual Basic) ภาษาซี (C) และภาษาจาวา (Java) เป็นต้น โปรแกรมที่เขียนด้วยภาษาระดับสูงแต่ละภาษาจะต้องมีโปรแกรมที่ทำหน้าที่แปล ภาษาระดับสูงให้เป็นภาษาเครื่อง เช่น โปรแกรมแปลภาษาฟอร์แทรนเป็นภาษาเครื่อง โปรแกรมแปลภาษาปาสคาลเป็นภาษาเครื่อง คำสั่งหนึ่งคำสั่งในภาษาระดับสูงจะถูกแปลเป็นภาษาเครื่องหลายคำสั่ง ภาษาระดับสูงที่จะกล่าวถึงในที่นี้ ได้แก่

1) ภาษาฟอร์แทรน (FORMula TRANstation : FORTRAN)

จัด เป็นภาษาระดับสูงที่เก่าแก่ที่สุด ได้รับการคิดค้นขึ้นเป็นครั้งแรก ราว พ.ศ. 2497 โดยบริษัท ไอบีเอ็ม เป็นภาษาที่เหมาะสมสำหรับงานที่ต้องการการคำนวณ เช่น งานทางด้านวิทยาศาสตร์ วิศวกรรมศาสตร์ และงานวิจัยต่าง ๆ เนื่องจากแนวคิดในการเขียนโปรแกรมในระยะหลังนี้เปลี่ยนมานิยมการเขียน โปรแกรมแบบโครงสร้างมากขึ้น ลักษณะของคำสั่งภาษาฟอร์แทรนแบบเดิมไม่เอื้ออำนวยที่จะให้เขียนได้ จึงมีการปรับปรุงโครงสร้างของภาษาฟอร์แทรนให้สามารถเขียนโปรแกรมแบบโครงสร้างขึ้นมาได้ในปี พ.ศ. 2509 เรียกว่า FORTRAN 66 และในปี พ.ศ. 2520 สถาบันมาตรฐานแห่งชาติของสหรัฐอเมริกา (American National Standard Institute หรือ ANSI) ได้ปรับปรุง FORTRAN 66 และยอมรับให้เป็นภาษาฟอร์แทรนที่เป็นมาตรฐาน เรียกว่า FORTRAN 77 ใช้ได้กับเครื่องคอมพิวเตอร์ที่มีตัวแปลภาษานี้

2) ภาษาโคบอล (Common Business Oriented Language : COBOL)

เป็น ภาษาที่พัฒนาขึ้นในราว พ.ศ. 2502 ต่อมาได้รับการปรับปรุงจากคณะกรรมการซึ่งเป็นตัวแทนของหน่วยงานธุรกิจและ รัฐบาลของสหรัฐอเมริกา เป็นภาษาโคบอลมาตรฐานในปี พ.ศ. 2517 เป็นภาษาที่เหมาะสมสำหรับงานด้านธุรกิจ เครื่องคอมพิวเตอร์ขนาดใหญ่ส่วนมากมีโปรแกรมแปลภาษาโคบอล

3) ภาษาเบสิก (Beginner's All – purpose Symbolic Instruction Code : BASIC)

เป็น ภาษาที่ได้รับการคิดขึ้นเป็นครั้งแรกที่วิทยาลัยดาร์มัทธ (Dartmouth College) และเผยแพร่เป็นทางการในปี พ.ศ. 2508 ภาษาเบสิกเป็นภาษาที่สร้างขึ้นโดยมีจุดประสงค์เพื่อใช้สอนเพื่อใช้สอน เขียนโปรแกรมแทนภาษาคอมพิวเตอร์ภาษาอื่น เช่น ภาษาฟอร์แทรน ซึ่งมีขนาดใหญ่และต้องใช้หน่วยความจำสูงในการทำงาน ซึ่งไม่เหมาะกับเครื่องคอมพิวเตอร์ในสมัยนั้น ภาษาเบสิกเป็นภาษาที่มีขนาดเล็ก เป็นตัวแปลภาษาชนิดที่เรียกว่าอินเทอร์พรีเตอร์

นอกจากนี้ ภาษาเบสิกเป็นภาษาที่ง่ายต่อการเขียน ซึ่งผู้เขียนจะสามารถนำไปประยุกต์กับการแก้ปัญหาต่าง ๆ ได้ทุกสาขาวิชา ผู้ที่เพิ่งฝึกเขียนโปรแกรมใหม่ ๆ หรือผู้ที่ไม่ใช่ นักเขียนโปรแกรมมืออาชีพ แต่เป็นเพียงวิศวกรหรือนักวิจัย จะสามารถหัดเขียนโปรแกรมภาษาเบสิกได้ในเวลาไม่นานนัก ปกติภาษาเบสิกส่วนใหญ่ใช้กับไมโครคอมพิวเตอร์

4) ภาษาปาสคาล (Pascal)

ตั้ง ชื่อตามนักคณิตศาสตร์ชาวฝรั่งเศส ชื่อ เบลส ปาสคาล (Blaise Pascal) ซึ่งเป็นผู้ผลิตเครื่องคิดเลขโดยใช้เฟืองหมุน ภาษาปาสคาลคิดขึ้นในปี พ.ศ. 2514 โดยนิคลอส เวียช (Niklaus Wirth) ศาสตราจารย์วิชาคอมพิวเตอร์ชาวสวิส ภาษาปาสคาลได้รับการออกแบบให้ใช้งานง่ายและมีโครงสร้างที่ดี จึงเหมาะกับการใช้สอนหลักการเขียนโปรแกรม ปัจจุบันภาษาปาสคาลยังคงได้รับความนิยมใช้ในการเรียนเขียนโปรแกรมคอมพิวเตอร์

5) ภาษาซีและซีพลัสพลัส (C และ C++)

ภาษา ซีเป็นภาษาที่พัฒนาจากห้องปฏิบัติการเบลล์ของบริษัทเอทีแอนด์ทีในปี พ.ศ. 2515 หลังจากที่ได้พัฒนาขึ้นได้ไม่นาน ภาษาซีก็กลายเป็นภาษาที่นิยมในหมู่นักเขียนโปรแกรมมาก และมีใช้งานในเครื่องทุกระดับ ทั้งนี้เนื่องจากภาษาซีได้รวมเอาข้อมูลของภาษาระดับสูงและภาษาระดับต่ำเข้าไว้ด้วยกัน กล่าวคือเป็นภาษาที่มีไวยากรณ์ที่เข้าใจง่าย ทำให้เขียนโปรแกรมได้ง่ายเช่นเดียวกับภาษาระดับสูงทั่วไป แต่ประสิทธิภาพและความเร็วในการทำงานดีกว่ามาก เนื่องจากการทำงานเหมือนภาษาระดับต่ำ สามารถทำงานได้ในระดับที่เป็นการควบคุมฮาร์ดแวร์ได้มากกว่าภาษาระดับสูงอื่น ๆ ดังจะเห็นว่าภาษาซีเป็นภาษาที่สามารถพัฒนาระบบปฏิบัติการได้ เช่น ระบบปฏิบัติการยูนิกซ์

นอกจากนี้เมื่อแนวคิดของการเขียนโปรแกรมแบบเชิงวัตถุ (Object Oriented Programming : OOP) ได้เข้ามามีบทบาทในวงการคอมพิวเตอร์มากขึ้น ภาษาซีก็ยังได้รับการพัฒนาโดยประยุกต์ใช้กับการเขียนโปรแกรมหดงกล่าว เกิดเป็นภาษาใหม่ชื่อว่า “ภาษาซีพลัสพลัส” (C++)

6) ภาษาวิซวลเบสิก (Visual Basic)

เป็น ภาษาที่พัฒนาต่อมาจากภาษาเบสิก ใช้ไวยากรณ์บางส่วนของภาษาเบสิกในการเขียนโปรแกรม แต่มีแนวคิดและวิธีการพัฒนาโปรแกรมที่แตกต่างจากภาษาเบสิกโดยสิ้นเชิง รวมทั้งการใช้เนื้อที่ในหน่วยความจำก็แตกต่างกันมาก ทั้งนี้เนื่องจากภาษาริซัลเบสิกใช้แนวคิดที่ต่างออกไป

7) การเขียนโปรแกรมแบบจินตภาพ (Visual Programming)

ภาษา นี้พัฒนาขึ้นโดยบริษัทไมโครซอฟต์ออกแบบเพื่อเขียนโปรแกรมที่สามารถใช้งานได้ บนระบบปฏิบัติการแบบจ็อยโอ เช่น ระบบปฏิบัติการไมโครซอฟต์วินโดวส์ มีการติดต่อกับผู้ใช้โดยใช้รูปภาพ การเขียนโปรแกรมทำได้ง่ายกว่าการเขียนโปรแกรมแบบเก่ามาก

8) ภาษาจาวา (Java)

พัฒนา ขึ้นในปี พ.ศ. 2534 โดยบริษัทซันไมโครซิสเต็มส์ เป็นภาษาที่ได้รับความนิยมสูงมาโดยตลอด เนื่องจากเป็นภาษาที่มีความยืดหยุ่นสูง สามารถเขียนโปรแกรมและใช้งานได้บนเครื่องคอมพิวเตอร์ทุกประเภทและระบบ ปฏิบัติการทุกรูปแบบ ในช่วงแรก ที่เริ่มมีการนำภาษาจาวามาใช้งานจะเป็นการใช้งานบนเครือข่ายอิน เทอร์เน็ต เป็น ภาษาที่เน้นการทำงานบนเว็บ แต่ปัจจุบันสามารถสามารถนำมาประยุกต์สร้างโปรแกรม ใช้งานได้ทั่วไป

นอกจากนี้ เมื่อเทคโนโลยีของการสื่อสารก้าวหน้าขึ้น จนกระทั่งเครื่องคอมพิวเตอร์ ปาล์มท็อป หรือ แม้แต่โทรศัพท์เคลื่อนที่สามารถเชื่อมต่อเข้าสู่ระบบอินเทอร์เน็ตและ ใช้งาน ระบบเว็ลด์ไวด์เว็บได้ ภาษาจาวาก็สามารถสร้างส่วนที่เรียกว่า “แอปเพล็ต” (Applet) ให้อุปกรณ์อิเล็กทรอนิกส์ที่กล่าวข้างต้น เรียกใช้งานจากเครื่องที่เป็นแม่ข่าย (Server) ได้

9) ภาษาเดลฟาย (Delphi)

เป็น ภาษาที่ได้รับความนิยมภาษาหนึ่ง แนวคิดในการเขียนโปรแกรมภาษาเด ลฟายเหมือนกับแนวคิดในการเขียนโปรแกรมภาษาริซัลเบสิก คือเป็นการเขียน โปรแกรมเชิงจินตภาพ แต่ภาษาพื้นฐานที่ใช้ในการเขียนโปรแกรมจะเป็นภาษา ปาสคาล ในการเขียนโปรแกรมเชิงจินตภาพนี้มีคอมโพเนนต์ (Component) ที่ สามารถใช้เป็นส่วนประกอบเพื่อสร้างส่วนติดต่อผู้ใช้ที่เป็นแบบกราฟิก ทำให้ ซอฟต์แวร์ที่พัฒนามีความน่าสนใจและใช้งานง่ายขึ้น การเขียนโปรแกรมด้วยภาษาเด ลฟายจึงเป็นที่นิยมในการนำไปพัฒนาเป็นโปรแกรมใช้ งานมาก รวมทั้งภาษาปาสคาลเป็นภาษาที่เข้าใจง่าย เหมาะแก่การนำมาใช้สอนเขียนโปรแกรม

4. ภาษาระดับสูงมาก

เป็น ภาษาโปรแกรมยุคที่ 4 ซึ่งเป็นภาษาระดับสูงมาก จัดเป็นภาษาไร้กระบวนคำสั่ง หมายความว่าผู้ใช้ เพียงบอกแต่จะให้คอมพิวเตอร์ทำอะไร โดยไม่ต้องบอก คอมพิวเตอร์ว่าสิ่งนั้นทำอย่างไร เรียกว่าเป็นภาษาเชิงผลลัพธ์ คือเน้นว่าทำอะไร ไม่ ใช่ว่าอย่างไร ดังนั้นจึงเป็นภาษาโปรแกรมที่เขียนง่าย

5. ภาษาธรรมชาติ

เป็น ภาษาโปรแกรมยุคที่ 5 ซึ่งคล้ายกับภาษาพูดตามธรรมชาติของคน การเขียน โปรแกรมง่ายที่สุด คือการเขียนคำพูดของเราเองว่าเราต้องการอะไร ไม่ต้องใช้คำสั่ง งานใดๆ เลย

ตัวอย่างภาษาในยุคต่างๆ ดังนี้

Fortran : ภาษาระดับสูงภาษาแรก เป็นภาษาโปรแกรมที่ใช้งานด้านวิทยาศาสตร์ วิศวกรรมศาสตร์ และด้านคณิตศาสตร์ ภาษาฟอร์เทนจะประกอบด้วยข้อความ คำสั่ง ที่ละบรรทัด

Colbol : ภาษาโปรแกรมสำหรับธุรกิจ ที่มีลักษณะคล้ายกับภาษาอังกฤษ และที่สำคัญคือ เป็นภาษาโปรแกรมที่อิสระจากเครื่อง หมายความว่า โปรแกรมที่เขียนขึ้นใช้งานบนคอมพิวเตอร์ชนิดหนึ่งเพียงแค่ปรับปรุงเล็กน้อย ก็สามารถรันได้บนคอมพิวเตอร์อีกชนิดหนึ่ง

Basic : ภาษาโปรแกรมสำหรับผู้เริ่มต้น เป็นภาษาโปรแกรมที่เรียนรู้ง่าย ไม่ซับซ้อน เหมาะสำหรับใช้ในวงการศึกษา

Pascal : เป็นภาษาสำหรับการเรียนการสอนโดยเฉพาะ เป็นภาษาที่เขียนง่าย ใช้ถ้อยคำน้อย

Ada : ภาษามาตรฐาน ซึ่งพัฒนาขึ้นโดย โปรแกรมเมอร์คนแรก คือ เคอด์ Add Lovelace เป็นภาษาที่ประสบความสำเร็จกับงานด้านธุรกิจ

C : ภาษาสมัยใหม่ เป็นภาษาที่ใช้สำหรับเขียนโปรแกรมระบบปฏิบัติการ เหมาะสำหรับโปรแกรมเมอร์ที่มีความสามารถสูง

ALGOL : เป็นภาษาที่ใช้เขียนโปรแกรมด้านวิทยาศาสตร์

LISP : เป็นภาษาที่ใช้เมื่อประมวลผลด้านสัญลักษณ์, อักษร, หรือคำต่างๆ ซึ่งเป็นการโต้ตอบระหว่างคนกับคอมพิวเตอร์ ภาษานี้นิยมใช้เขียนโปรแกรมด้านปัญญาประดิษฐ์

Prolog : เป็นภาษาโปรแกรมสำหรับงานด้านปัญญาประดิษฐ์ ซึ่งแทนการใช้ภาษา LISP

PL/1 : เป็นภาษาที่เรียนรู้ง่าย ใช้งานทั้งด้านวิทยาศาสตร์ และด้านธุรกิจ ดังนั้นภาษานี้จะมีขนาดใหญ่ มี option มาก

ALP : เป็นภาษาที่เหมาะสมกับการทำตาราง มีสัญลักษณ์ต่างๆ มาก

Logo : เป็นภาษาย่อยของ lisp เป็นโปรแกรมสำหรับเด็ก มีการสนทนาโต้ตอบกับคอมพิวเตอร์ โดยใช้ "เต่า" เป็นสัญลักษณ์โต้ตอบกับคำสั่งง่ายเช่น forward, left

Pilot : เป็นภาษาโปรแกรมที่นิยมใช้มากที่สุดใน การเขียนโปรแกรมบทเรียนคอมพิวเตอร์ ช่วยสอน(CAI) เช่น งานเกี่ยวกับคำสั่ง ผูกหัด การทดสอบ เป็นต้น

Smalltalk : เป็นภาษาเชิงโต้ตอบกับเครื่องคอมพิวเตอร์ประกอบด้วย การจำ และการพิมพ์ เป็นภาษาที่สนับสนุนระบบคอมพิวเตอร์ภาพ เป็นภาษาเชิงวัตถุไม่ใช่เชิงกระบวนการ

Forth : เป็นภาษาสำหรับงานควบคุมแบบทันที เช่นการแนะนำกล้องดาราศาสตร์ และเป็นภาษาโปรแกรมที่มีความเร็วสูง

Modula-2 : คล้ายคลึงกับภาษาปาสคาล ออกแบบมาเพื่อให้เขียนซอฟต์แวร์ระบบ

RPG : เป็นภาษาเชิงปัญหา ออกแบบมาเพื่อใช้แก้ปัญหาการทำรายงานเชิงธุรกิจ เช่น การปรับปรุงแฟ้มข้อมูล

3.2 การเขียนอัลกอริทึมด้วยภาษาธรรมชาติ

การบรรยายขั้นตอนการทำงานของอัลกอริทึมใด ๆ โดยใช้ภาษามนุษย์เพื่ออธิบายถึงลำดับขั้นตอนการทำงาน ตามลำดับการทำงานก่อนหลัง

ตัวอย่าง การอธิบายการการคำนวณหาพื้นที่สี่เหลี่ยม และแสดงผลลัพธ์การคำนวณด้วยการใช้ภาษาธรรมชาติ

สูตรการคำนวณหาพื้นที่สี่เหลี่ยม พื้นที่สี่เหลี่ยม = ความกว้าง x ความยาว

1. เริ่มต้นการทำงาน

2. นำเข้าข้อมูลความกว้างของสี่เหลี่ยม

3. นำเข้าข้อมูลความยาวของสี่เหลี่ยม

4. คำนวณพื้นที่สี่เหลี่ยม กว้าง*ยาว

5. แสดงผลพื้นที่สี่เหลี่ยม

6. จบการทำงาน

การเขียนอัลกอริทึมด้วยภาษาธรรมชาติ

▶ การบรรยายขั้นตอนการทำงานของอัลกอริทึมใด ๆ โดยใช้ภาษามนุษย์เพื่ออธิบายถึงลำดับขั้นตอนการทำงาน ตามลำดับการทำงานก่อนหลัง

ตัวอย่าง การอธิบายการการคำนวณหาพื้นที่สี่เหลี่ยม และแสดงผลลัพธ์การคำนวณ ด้วยการ ใช้ภาษาธรรมชาติ

สูตรการคำนวณหาพื้นที่สี่เหลี่ยม พื้นที่สี่เหลี่ยม = ความกว้าง x ความยาว



AKSADON



4. การเขียนอัลกอริทึมด้วยรหัสจำลอง (Pseudocode)

การเขียนอัลกอริทึมด้วยรหัสจำลอง (Pseudocode)

คือ การเขียนคำอธิบายขั้นตอนการทำงานของโปรแกรม โดยใช้ถ้อยคำผสมระหว่างภาษาอังกฤษและภาษาการเขียนโปรแกรมแบบโครงสร้าง ซึ่งจะช่วยให้ผู้เขียนโปรแกรมสามารถพัฒนาขั้นตอนต่าง ๆ ให้เป็นโปรแกรมได้ง่ายขึ้น ส่วนใหญ่มักใช้คำเฉพาะ (Reserve Word) ที่มีในภาษาการเขียนโปรแกรมและมักเขียนด้วยตัวอักษรตัวใหญ่ ชุดโค๊ดที่ดี จะต้องมีความชัดเจน สั้น และใส่ใจความ ข้อมูลต่าง ๆ ที่ใช้จะถูกเขียนอยู่ในรูปของตัวแปร

หลักการเขียนรหัสเทียม (Pseudocode)

หลักการเขียนรหัสเทียมไม่มีกฎหรือมาตรฐานที่ตายตัว ผู้เขียนจึงอยากให้ผู้อ่านได้นำไปใช้เพื่อเป็นแนวทางในการเขียนรหัสเทียม ที่มีประสิทธิภาพยิ่งขึ้น โดยการรวบรวมหลักการต่าง ๆ ที่สำคัญเอาไว้

1. กำหนดจุดเริ่มต้นด้วยคำว่า“Begin”และจุดสิ้นสุดด้วยคำว่า“End”
2. ถ้อยคำต่าง ๆ ให้เขียนเป็นภาษาอังกฤษอย่างง่าย
3. การเขียนรหัสเทียมแต่ละคำสั่งควรเขียนเป็นบรรทัด ๆ
4. ควรมีการย่อหน้า เพื่อสะดวกต่อการอ่านและการตรวจสอบ
5. การเขียนรหัสเทียม จะเขียนจากบนลงล่าง และมีทางเข้าหนึ่งทาง ทางออกหนึ่งทาง
6. การเขียนรหัสเทียมจะไม่เขียนหมายเลขกำกับในแต่ละขั้นตอน
7. ควรใช้การย่อหน้าให้เป็นประโยชน์ การแยกคำเฉพาะ (Keywords) ให้มีความชัดเจน นอกจากนี้ ควรจัดรูปแบบโครงสร้างควบคุมให้เป็นสัดส่วน เพื่อให้อ่านง่าย
8. กลุ่มประโยคคำสั่งต่าง ๆ อาจถูกนำมาจัดรวมกลุ่มเข้าด้วยกันในรูปแบบของโมดูล และทำการกำหนดชื่อโมดูลขึ้นมา เพื่อให้ส่วนของโปรแกรมหลัก หรือโมดูลย่อย อื่นๆ เรียกใช้งานได้

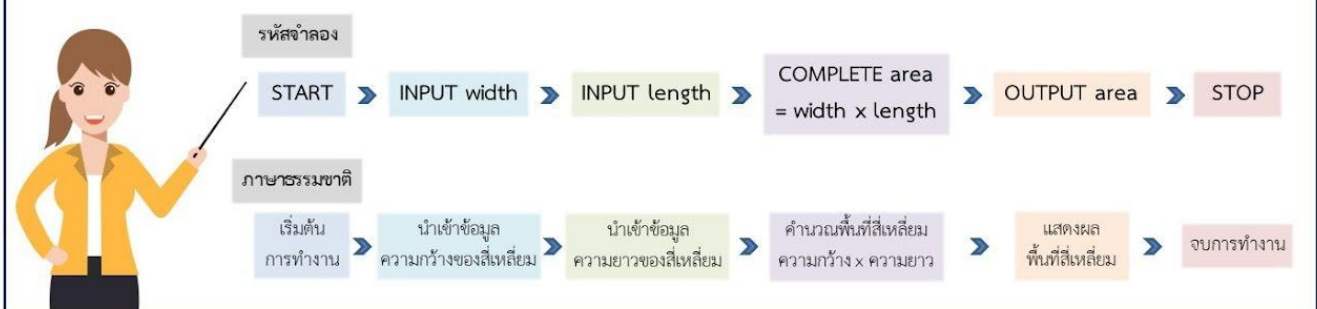
การเขียนอัลกอริทึมด้วยรหัสจำลอง

► เป็นคำสั่งที่จำลองความคิดเป็นลำดับขั้นตอนโดยใช้สัญลักษณ์เป็นประโยคภาษาอังกฤษ ตัวอย่างประโยคภาษาอังกฤษเขียนจำลองคำสั่ง เช่น

Input a number : ใส่ข้อมูลนำเข้าเป็นค่าตัวเลข Find the sum of the number : คำนวณรวมค่าตัวเลขที่นำเข้า

ตัวอย่าง การอธิบายการการคำนวณหาพื้นที่สี่เหลี่ยม และแสดงผลลัพธ์การคำนวณ ด้วยการใช้รหัสจำลอง

สูตรการคำนวณหาพื้นที่สี่เหลี่ยม พื้นที่สี่เหลี่ยม = ความกว้าง x ความยาว



4.1 การเขียนอัลกอริทึมด้วยรหัสจำลอง

การเขียนรหัสจำลอง (Pseudo Code) คือ การเขียนอัลกอริทึมโดยใช้ประโยคภาษาอังกฤษที่สื่อความหมายง่าย ๆ สามารถอ่านแล้วเข้าใจได้โดยทันที แต่ก็สามารถใช้รูปแบบที่เป็นภาษาพูดด้วยภาษาไทยและภาษาอังกฤษก็ได้

โครงสร้างของรหัสจำลองเริ่มต้นด้วยข้อความ Begin แล้วอธิบายขั้นตอนการทำงานโดยใช้คำสั่งต่าง ๆ ที่ใกล้เคียงกับภาษาคอมพิวเตอร์ในการเขียนโปรแกรม เช่น คำสั่ง read หมายถึง การอ่านค่าหรือรับค่าข้อมูลตัวแปรตามที่กำหนดไว้ คำสั่ง print หมายถึง การแสดงผลลัพธ์ที่ได้จากการคำนวณ และพิมพ์ข้อความ End เมื่อจบการทำงาน

Input a number : ใส่ข้อมูลนำเข้าเป็นค่าตัวเลข

Find the sum of the number : คำนวณรวมค่าตัวเลขที่นำเข้า

Print the sum : แสดงผลรวมของข้อมูลทั้งหมด

การเขียนรหัสจำลองจะต้องมีการวางแผนสำหรับการอ้างอิงถึงข้อมูลที่จะต้องนำไปใช้ภายในโปรแกรมด้วย การสร้างตัวแปร โดยใช้เครื่องหมายเท่ากับ (=) แทนการกำหนดค่าตัวแปร

4.2 หลักเกณฑ์การเขียนรหัสจำลอง

ขั้นที่ 1 ควรใช้คำสั่งเป็นรูปแบบภาษาที่สั้น กระชับ เข้าใจงาน เช่น

Start หมายถึง การเริ่มต้นการทำงาน

Stop หมายถึง การจบการทำงาน

ขั้นที่ 2 ควรมีหมายเลขลำดับขั้นตอนที่ชัดเจน

ขั้นที่ 3 ไม่จำเป็นต้องมีเครื่องหมายวรรคตอน

ขั้นที่ 4 รหัสจำลองต้องไม่ขึ้นกับภาษาคอมพิวเตอร์ภาษาใดภาษาหนึ่ง

ขั้นที่ 5 ในหนึ่งบรรทัดให้มีคำสั่งเพียงคำสั่งเดียว

ขั้นที่ 6 ในการเขียนคำสั่งให้เรียงจากบนลงล่างและมีจุดสิ้นสุดเพียงจุดเดียว

การเขียนอัลกอริทึม คำนวณหาพื้นที่สามเหลี่ยม

อัลกอริทึม (Algorithm) การหาพื้นที่สามเหลี่ยม

ภาษาธรรมชาติ

เริ่มต้น

นำเข้าข้อมูล ความยาว รับค่าความยาวของฐานมาเก็บในตัวแปร BASE

นำเข้าข้อมูล ความสูง รับค่าความยาวของสูงมาเก็บในตัวแปร HEIGHT

คำนวณหาพื้นที่สามเหลี่ยม = $0.5 * \text{ยาว} * \text{สูง}$

แสดงผลพื้นที่

จบ

รหัสจำลอง

Start

INPUT BASE

INPUT HEIGHT

COMPUTE AREA = $0.5 * \text{BASE} * \text{HEIGHT}$

OUTPUT AREA

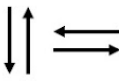
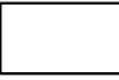
Stop

5. การเขียนอัลกอริทึมด้วยผังงาน (Flowchart)

การเขียนอัลกอริทึมด้วยผังงาน (Flowchart)

ผังงาน (Flowchart) คือ แผนภาพแสดงการทำงานของโปรแกรม โดยใช้สัญลักษณ์แสดงขั้นตอนและลักษณะการทำงานแบบต่างๆ สัญลักษณ์เหล่านี้จะถูกเชื่อมโยงด้วยลูกศรเพื่อแสดงลำดับการทำงาน ช่วยให้เห็นภาพการทำงานโดยรวมของโปรแกรม สะดวกต่อการตรวจสอบความถูกต้องของลำดับการทำงานและการไหลของข้อมูลในโปรแกรม การเขียนผังงานจะใช้สัญลักษณ์สื่อสารความหมายให้เข้าใจตรงกัน

ของสถาบันมาตรฐานแห่งชาติอเมริกัน (The American National Standard Institute, ANSI) ได้กำหนดสัญลักษณ์ไว้เป็นมาตรฐาน

	เส้นเดี่ยว single line		คู่วางแนว double line
	เส้นจุด center line		บล็อก block
	เส้นที่มีรอยขาด line with break		เส้นตัด section line
	เส้นที่มีรอยขาด line with break		เส้นที่มีรอยขาด line with break
	เส้นที่มีรอยขาด line with break		เส้นที่มีรอยขาด line with break
แบบสัญลักษณ์	แบบสัญลักษณ์	แบบสัญลักษณ์	แบบสัญลักษณ์

รูปแสดงแบบสัญลักษณ์ของคู่วางแนว เส้นจุด เส้นที่มีรอยขาด เส้นตัด และเส้นที่มีรอยขาด

แสดงสัญลักษณ์ที่ใช้ในการเขียนผังงาน

5.1 รูปแบบการเขียนผังงาน

การเขียนผังงานจะเขียนในลักษณะบนลงล่าง (Top-Down) หรือซ้ายไปขวา โดยมีรูปแบบการเขียนผังงาน 3 ลักษณะ ดังนี้

1. โครงสร้างแบบลำดับ (Sequential Structure/Sequence) หมายถึง โครงสร้างที่แสดงขั้นตอนการทำงานเป็นไปตามลำดับก่อนหลัง

C 1. โครงสร้างแบบลำดับ

ตัวอย่าง ลำดับขั้นตอนการวางแผนไปโรงเรียน

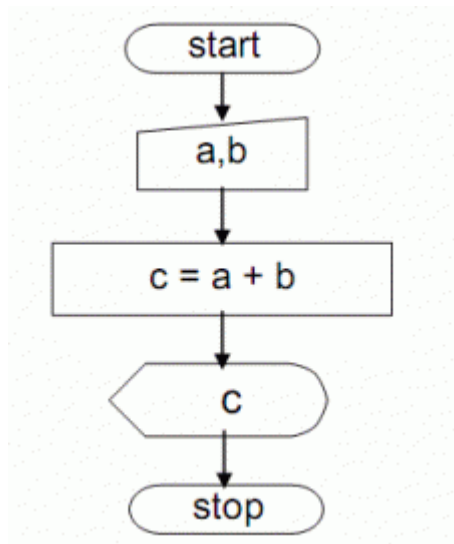
เริ่มต้น

ตื่นนอน

อาบน้ำแต่งตัว

ไปโรงเรียน

จบ



2. โครงสร้างแบบมีทางเลือก (Selection Structure/condition) หมายถึง โครงสร้างที่มีเงื่อนไข ขั้นตอนการทำงานบางขั้นตอน ต้องมีการตัดสินใจ

C 2. โครงสร้างแบบทางเลือก

ตัวอย่าง ลำดับขั้นตอนการประเมินผลสอบ

เริ่มต้น

ทดสอบ

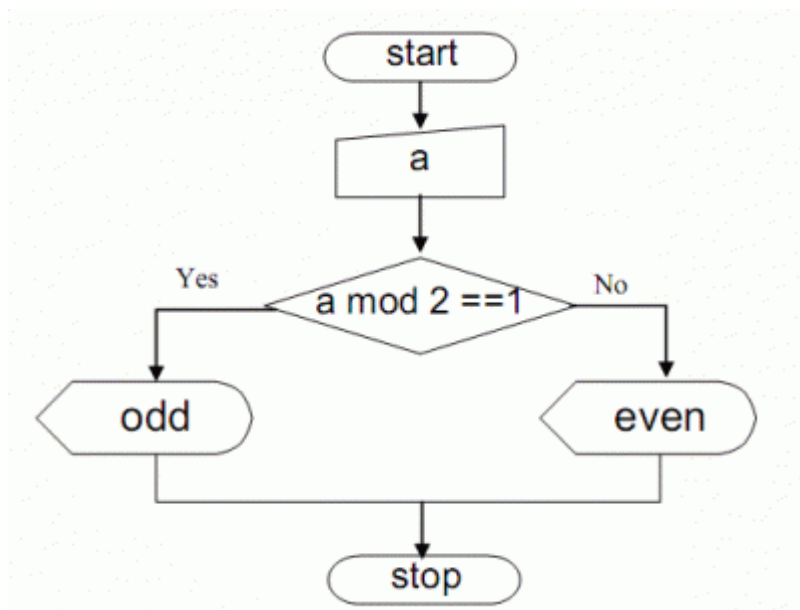
ตรวจผลการสอบและคิดคะแนนที่ได้

ตรวจสอบคะแนนที่ได้ว่าน้อยกว่าร้อยละ 50 หรือไม่

ถ้าน้อยกว่า ให้สอบแก้ตัว

ถ้าไม่น้อยกว่า ให้สอบผ่าน

จบ



3. โครงสร้างแบบทำซ้ำ (Repetition Structure/Loop) หมายถึง โครงสร้างที่ขั้นตอนการทำงานบางขั้นตอนได้รับการประมวลผลมากกว่า 1 ครั้ง

C 3. โครงสร้างแบบทำซ้ำ

ตัวอย่าง ลำดับการเติมน้ำจากตุ่มครั้งละ 1 ชันใสจนถึงน้ำเต็ม
เริ่มต้น

เติมน้ำจากตุ่ม 1 ชัน

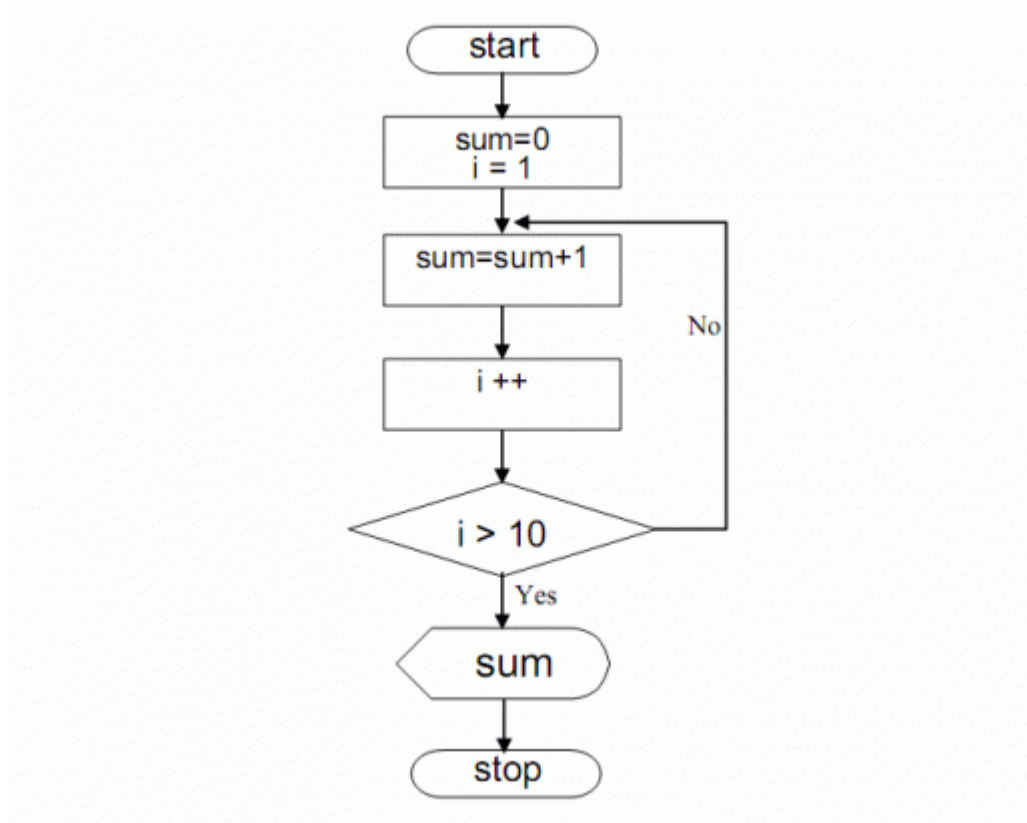
เทน้ำใส่ถัง

ตรวจสอบน้ำเต็มถึง หรือไม่

ถ้าไม่เต็ม ให้เติมน้ำต่อไป

ถ้าเต็ม ให้หยุดเติมน้ำ

จบ



5.2 การเขียนอัลกอริทึมด้วยผังงาน

หลักการเขียนผังงานจะอ้างอิงจากกระบวนการทางคอมพิวเตอร์ 3 กระบวนการ ได้แก่ การรับข้อมูล (Input) การประมวลผล (Process) และการแสดงผลลัพธ์ (Output) โดยการเขียนผังงานที่ดีควรมีหลักการเขียน ดังนี้

- ใช้สัญลักษณ์ที่เป็นสัญลักษณ์มาตรฐาน
- เขียนทิศทางการทำงานจากบนลงล่าง หรือซ้ายไปขวา โดยมีหัวลูกศรกำกับทิศทางเสมอ
- จุดเริ่มต้นและจุดสิ้นสุดควรมีเพียงจุดเดียว
- การเขียนคำอธิบายควรเขียนภายในภาพสัญลักษณ์ โดยใช้ข้อความที่กระชับ ชัดเจน และเข้าใจง่าย

การเขียนอัลกอริทึมด้วยผังงาน

▶ หลักการเขียนอัลกอริทึมด้วยผังงานจะอ้างอิงจากกระบวนการทางคอมพิวเตอร์ 3 กระบวนการ ได้แก่



ประโยชน์ของการเขียนผังงาน

ผังงานนอกจากจะช่วยในการพัฒนาลำดับขั้นตอนของการแก้ปัญหาแล้วยังมีประโยชน์ดังต่อไปนี้

1. ช่วยลำดับขั้นตอนการทำงานได้ง่าย ไม่สับสน เข้าใจได้ง่าย
2. เมื่อเกิดข้อผิดพลาดช่วยในการตรวจสอบ และแก้ไขงานได้ง่าย
3. ช่วยให้การดัดแปลง แก้ไข ทำได้อย่างสะดวกรวดเร็ว
4. ช่วยให้ผู้อื่นสามารถศึกษาการทำงานของโปรแกรมได้อย่างง่าย และรวดเร็วมากขึ้น
5. สามารถเรียนรู้และเข้าใจผังงานได้ง่าย เพราะผังงานเป็นการสื่อความหมายด้วยภาพทำให้ง่ายและสะดวกต่อการพิจารณาลำดับขั้นตอนในการทำงานดีกว่าการบรรยายเป็นตัวอักษร

ข้อจำกัดของผังงาน

ผู้เขียนโปรแกรมบางคนไม่นิยมการเขียนผังงานก่อนที่จะเขียนโปรแกรม เพราะจะทำให้เสียเวลาในการเขียนเป็นรูปภาพหรือสัญลักษณ์ต่างๆ โดยข้อจำกัดของผังงาน มีดังนี้

1. ผังงานเป็นการสื่อความหมายระหว่างบุคคลต่อบุคคลมากกว่าที่จะสื่อความหมายระหว่าง บุคคลกับเครื่องคอมพิวเตอร์ เพราะผังงานระบบไม่ขึ้นอยู่กับภาษาคอมพิวเตอร์ภาษาใดภาษาหนึ่ง ทำให้เครื่องคอมพิวเตอร์ไม่สามารถรับและเข้าใจว่าในผังงานนั้นต้องการให้ทำอะไร
2. บางครั้งเมื่อพิจารณาจากผังงานระบบ จะไม่สามารถทราบได้ว่า ขั้นตอนการทำงานใดสำคัญกว่ากัน เพราะทุกๆ ขั้นตอนจะใช้รูปภาพหรือสัญลักษณ์ในลักษณะเดียวกัน
3. การเขียนผังงานเป็นการสิ้นเปลืองงบประมาณ และเวลา เพราะจะต้องใช้กระดาษและอุปกรณ์ อื่นๆ ประกอบการเขียนภาพ บางครั้งการเขียนผังงานอาจจะต้องใช้กระดาษมากกว่า 1 แผ่น ทั้งที่การอธิบายงานเดียวกันจะใช้เนื้อที่เพียง 3-4 บรรทัดเท่านั้น

4. การเขียนผังงานจะมีขนาดใหญ่ ถ้าโปรแกรมที่พัฒนาเป็นงานใหญ่ ทำให้ผังงานระบบแลดูไม่คล่องตัว และถ้ามีการปรับเปลี่ยนผังงานจะทำได้ยาก บางครั้งอาจจะต้องเขียนผังงานชิ้นใหม่
5. ผังงานจะบอกขั้นตอนการปฏิบัติงานว่าเป็นลำดับอย่างไร ปฏิบัติงานอะไรแต่จะไม่ระบุให้ทราบว่าทำไมจึงต้องเป็นลำดับและต้องปฏิบัติงานอย่างนั้น
6. ภาษาคอมพิวเตอร์ที่ใช้ในปัจจุบันผังงานไม่สามารถแทนลักษณะคำสั่งในภาษาได้ชัดเจนตรงไปตรงมา