บทที่ 4
โครงสร้างการควบคุมโปรแกรมแบบเรียงลำดับ
Sequential Control Structure

บทนำ

การแก้ปัญหาอย่างเป็นระบบเป็นหัวใจสำคัญของวิทยาการคำนวณ เพราะไม่ว่าปัญหาจะอยู่ในรูปของการคำนวณ การตัดสินใจ หรือการพัฒนาชิ้นงานดิจิทัล ผู้เรียนจำเป็นต้องสามารถลำดับความคิดและกำหนดขั้นตอนการดำเนินงานได้อย่างชัดเจน หนึ่งในพื้นฐานสำคัญของการเขียนโปรแกรมและการออกแบบวิธีการแก้ปัญหา คือ ความเข้าใจเกี่ยวกับโครงสร้างการควบคุมโปรแกรม ซึ่งเป็นรูปแบบการกำหนดลำดับการทำงานของคำสั่งในโปรแกรม โดยโครงสร้างพื้นฐานของการควบคุมโปรแกรมโดยทั่วไปประกอบด้วย 3 รูปแบบสำคัญ ได้แก่ โครงสร้างแบบเรียงลำดับ โครงสร้างแบบทางเลือก และโครงสร้างแบบทำซ้ำ

ในบรรดาโครงสร้างพื้นฐานทั้งสามรูปแบบ โครงสร้างการควบคุมแบบเรียงลำดับถือเป็นจุดเริ่มต้นที่สำคัญที่สุด เนื่องจากเป็นรูปแบบการทำงานที่ง่ายและตรงไปตรงมา กล่าวคือ คำสั่งจะถูกดำเนินการจากบนลงล่างหรือจากต้นไปปลายตามลำดับที่กำหนดไว้โดยไม่มีการข้าม การเลือกเงื่อนไขหรือการวนซ้ำ ความเข้าใจในโครงสร้างนี้จึงเป็นรากฐานสำคัญสำหรับการเรียนรู้การเขียนอัลกอริทึม การสร้างผังงาน และการเขียนโปรแกรมในระดับต่อไป

สำหรับผู้เรียนในสาขาครุศาสตร์ โดยเฉพาะนักศึกษาครูที่ต้องเรียนรายวิชาวิทยาการคำนวณ ความเข้าใจเรื่องโครงสร้างแบบเรียงลำดับมิได้มีคุณค่าเฉพาะในด้านการเขียนโปรแกรมเท่านั้น แต่ยังมีความสำคัญในฐานะเครื่องมือในการฝึกคิดอย่างมีขั้นตอน และเป็นพื้นฐานสำหรับการออกแบบกิจกรรมการเรียนรู้ให้แก่ผู้เรียนระดับประถมศึกษาและมัธยมศึกษา ครูที่เข้าใจโครงสร้างแบบเรียงลำดับอย่างแท้จริงจะสามารถอธิบายเรื่องอัลกอริทึมและผังงานให้ผู้เรียนเข้าใจได้ง่าย โดยเชื่อมโยงกับสถานการณ์ในชีวิตจริง เช่น การแปรงฟัน การทำอาหาร การเดินทางไปโรงเรียน หรือการจัดกิจวัตรประจำวัน

ในบทนี้จะกล่าวถึงความหมายของอัลกอริทึมและการแสดงลำดับขั้นตอนการทำงาน หลักการของโครงสร้างการควบคุมแบบเรียงลำดับ สัญลักษณ์ผังงานเบื้องต้นและการออกแบบผังงานแบบเรียงลำดับ การใช้โปรแกรม Flowgorithm เพื่อเขียนและทดสอบผังงานแบบเรียงลำดับ ตลอดจน

แนวทางการออกแบบกิจกรรมการเรียนรู้เพื่อสอนโครงสร้างแบบเรียงลำดับในระดับประถมศึกษาและมัธยมศึกษา เพื่อให้นักศึกษาครูสามารถนำความรู้ไปประยุกต์ใช้ได้ทั้งในเชิงวิชาการและเชิงวิชาชีพครู

4.1 ความหมายของอัลกอริทึมและการแสดงลำดับขั้นตอนการทำงาน

อัลกอริทึม (Algorithm) หมายถึง ลำดับขั้นตอนหรือวิธีการแก้ปัญหาที่ถูกกำหนดไว้อย่างชัดเจน เป็นระบบ และสามารถปฏิบัติตามได้เพื่อให้ได้ผลลัพธ์ตามต้องการ อัลกอริทึมจึงเป็นเสมือนแผนปฏิบัติการที่บอกว่า หากต้องการแก้ปัญหาหนึ่ง ๆ ควรทำอะไร อย่างไร และตามลำดับขั้นตอนใด

ในบริบทของวิทยาการคำนวณ อัลกอริทึมมีบทบาทสำคัญอย่างยิ่ง เพราะเป็นพื้นฐานของการเขียนผังงาน การพัฒนาโปรแกรม และการออกแบบระบบต่าง ๆ ก่อนที่ผู้เรียนจะเขียนคำสั่งด้วยภาษาโปรแกรมได้ จำเป็นต้องสามารถคิดและเรียบเรียงขั้นตอนการทำงานให้เป็นระบบเสียก่อน หากขาดความเข้าใจในอัลกอริทึม การเขียนโปรแกรมอาจกลายเป็นเพียงการจำคำสั่งโดยไม่เข้าใจโครงสร้างการทำงานอย่างแท้จริง

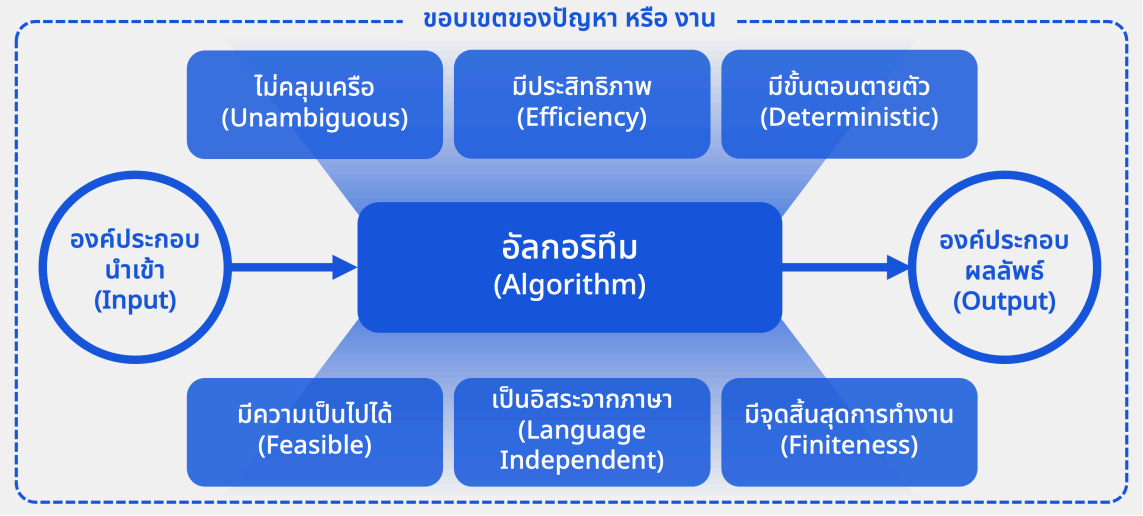
ลักษณะสำคัญของอัลกอริทึมที่ดีควรประกอบด้วยประเด็นต่อไปนี้ ประการแรก คือ มีความชัดเจน กล่าวคือ แต่ละขั้นตอนต้องระบุอย่างชัดเจนว่าให้ทำอะไร ประการที่สอง คือ มีลำดับขั้นตอนที่ถูกต้อง ไม่สับสนหรือวกวน ประการที่สาม คือ สามารถปฏิบัติได้จริง หมายความว่าผู้ปฏิบัติหรือเครื่องคอมพิวเตอร์สามารถดำเนินการตามขั้นตอนนั้นได้ และประการที่สี่ คือ มีจุดเริ่มต้นและจุดสิ้นสุดที่แน่นอน

อัลกอริทึมสามารถแสดงได้หลายรูปแบบ เช่น การบรรยายเป็นภาษาธรรมชาติ การเขียนเป็นรหัสลำลอง (Pseudocode) และการเขียนเป็นผังงาน (Flowchart) การบรรยายเป็นภาษาธรรมชาติเป็นวิธีที่เข้าใจง่าย เหมาะกับผู้เริ่มต้น เช่น การเขียนขั้นตอนการชงกาแฟว่า 1) ต้มน้ำ 2) ใส่กาแฟ 3) เติมน้ำร้อน 4) คนให้เข้ากัน 5) เสิร์ฟ เป็นต้น อย่างไรก็ตาม หากขั้นตอนมีความซับซ้อนมากขึ้น การใช้ผังงานหรือรหัสลำลองจะช่วยให้เห็นลำดับความคิดได้ชัดเจนกว่า

การแสดงลำดับขั้นตอนการทำงานเป็นทักษะสำคัญสำหรับผู้เรียน เพราะช่วยฝึกให้คิดอย่างมีระบบและสามารถถ่ายทอดแนวคิดให้ผู้อื่นเข้าใจได้ ตัวอย่างในชีวิตประจำวัน เช่น ขั้นตอนการล้างมือ การแต่งกาย การส่งงานออนไลน์ หรือการเปิดใช้งานแอปพลิเคชัน ล้วนสามารถอธิบายในลักษณะของอัลกอริทึมได้ทั้งสิ้น

ลักษณะของอัลกอริทึมที่ดีที่ประกอบด้วยคุณสมบัติ ดังนี้

1. **ไม่คลุมเครือ (Unambiguous)** อัลกอริทึมต้องไม่มีความกำหนดที่ไม่ชัดเจน และต้องมีชุดคำสั่งที่เข้าใจง่ายและไม่ทำให้เกิดความสับสนในการประมวลผล
2. **มีประสิทธิภาพ (Efficiency)** อัลกอริทึมควรมีประสิทธิภาพในการทำงาน คือต้องให้ผลลัพธ์ที่ถูกต้องและใช้เวลาน้อยที่สุดในการแก้ปัญหา
3. **มีขั้นตอนตายตัว (Deterministic)** อัลกอริทึมต้องเป็น deterministic ทำหมายความว่าจะต้องให้ผลลัพธ์ที่เหมือนกันสำหรับข้อมูลนำเข้าที่เหมือนกันทุกครั้งที่รัน
4. **มีความเป็นไปได้ (Feasible)** อัลกอริทึมต้องเป็นไปได้ในการนำไปใช้งานจริง สามารถทำงานบนเครื่องคอมพิวเตอร์หรือสภาพแวดล้อมที่เหมาะสม
5. **เป็นอิสระจากภาษา (Language Independent)** อัลกอริทึมควรไม่เกี่ยวข้องกับภาษาเฉพาะเสียเพียงอย่างใด ๆ และสามารถนำไปใช้งานในหลายภาษาโปรแกรมได้
6. **มีจุดสิ้นสุดการทำงาน (Finiteness)** อัลกอริทึมต้องสามารถสิ้นสุดการทำงานหลังจากทำขั้นตอนทุกขั้นตอนเสร็จสมบูรณ์ และไม่ต้องเข้าสู่วงจรไม่สิ้นสุด (Infinity Loop)



สำหรับนักศึกษาครู การเข้าใจอัลกอริทึมไม่ควรหยุดอยู่ที่การเขียนลำดับขั้นตอนเพียงอย่างเดียว แต่ควรสามารถเชื่อมโยงไปสู่การออกแบบกิจกรรมการเรียนรู้ เช่น ให้ผู้เรียนลองเขียนขั้นตอนจากกิจกรรมใกล้ตัว ทดลองปฏิบัติตามขั้นตอนของเพื่อน และตรวจสอบว่ามีจุดใดที่สื่อสารไม่ชัดเจน กิจกรรม

ลักษณะนี้จะช่วยให้ผู้เรียนตระหนักว่าอัลกอริทึมที่ดีต้องชัดเจน ถูกต้อง และสื่อสารได้อย่างมีประสิทธิภาพ

กล่าวโดยสรุป อัลกอริทึมเป็นรากฐานของการคิดเชิงคำนวณและการเขียนโปรแกรม เพราะช่วยให้ผู้เรียนสามารถวางแผนและลำดับขั้นตอนการแก้ปัญหาอย่างเป็นระบบ ซึ่งเป็นสมรรถนะพื้นฐานที่จำเป็นต่อการเรียนรู้วิทยาการคำนวณในระดับที่สูงขึ้น

4.2 หลักการของโครงสร้างการควบคุมแบบเรียงลำดับ (Sequential Structure)

โครงสร้างการควบคุมแบบเรียงลำดับ (Sequential Structure) คือ รูปแบบการทำงานของโปรแกรมที่คำสั่งแต่ละคำสั่งจะถูกดำเนินการตามลำดับจากต้นไปปลาย หรือจากบนลงล่างตามที่กำหนดไว้ โดยไม่มีการข้ามขั้นตอน ไม่มีการตรวจสอบเงื่อนไขเพื่อเลือกทางใดทางหนึ่ง และไม่มีการทำซ้ำหลายรอบ โครงสร้างนี้จึงเป็นรูปแบบที่ง่ายที่สุดและเป็นพื้นฐานของการเขียนโปรแกรมทุกประเภท

หลักการสำคัญของโครงสร้างแบบเรียงลำดับ คือ การให้ความสำคัญกับ “ลำดับ” ของคำสั่ง เพราะหากจัดลำดับผิด ผลลัพธ์ที่ได้อาจไม่ถูกต้อง ตัวอย่างเช่น หากต้องการคำนวณพื้นที่สี่เหลี่ยมผืนผ้าจำเป็นต้องรับค่าความกว้างและความยาวก่อน จากนั้นจึงนำมาคูณกัน และสุดท้ายจึงแสดงผลลัพธ์ หากสลับลำดับ เช่น ให้แสดงผลก่อนรับข้อมูล ผลลัพธ์ย่อมไม่สมบูรณ์หรือไม่ถูกต้อง

ในทางปฏิบัติ โครงสร้างแบบเรียงลำดับมักใช้กับงานที่มีขั้นตอนตายตัว เช่น การรับข้อมูล การคำนวณ และการแสดงผล เช่น โปรแกรมคำนวณอายุ โปรแกรมหาค่าเฉลี่ยคะแนน หรือโปรแกรมแปลงหน่วยอุณหภูมิ ขั้นตอนการทำงานมักเริ่มจากกำหนดตัวแปร รับค่าจากผู้ใช้ ประมวลผลตามสูตรที่กำหนด แล้วจึงแสดงผลออกมา

ตัวอย่างง่าย ๆ ของโครงสร้างแบบเรียงลำดับ เช่น การคำนวณผลรวมของตัวเลขสองจำนวน อาจมีลำดับขั้นตอนดังนี้

1. เริ่มต้นการทำงาน
2. รับค่าตัวเลขจำนวนที่ 1
3. รับค่าตัวเลขจำนวนที่ 2
4. นำค่าทั้งสองมาบวกกัน
5. แสดงผลลัพธ์
6. สิ้นสุดการทำงาน

จากตัวอย่างจะเห็นได้ว่า ทุกขั้นตอนต้องเกิดขึ้นตามลำดับ หากละขั้นตอนใดไปหรือสลับลำดับผลลัพธ์ของโปรแกรมก็อาจไม่เป็นไปตามที่ต้องการ ดังนั้น ผู้เรียนจึงควรฝึกมองเห็นความสัมพันธ์ระหว่างข้อมูลนำเข้า (Input) กระบวนการประมวลผล (Process) และผลลัพธ์ (Output) ซึ่งเป็นแกนสำคัญของโครงสร้างแบบเรียงลำดับ

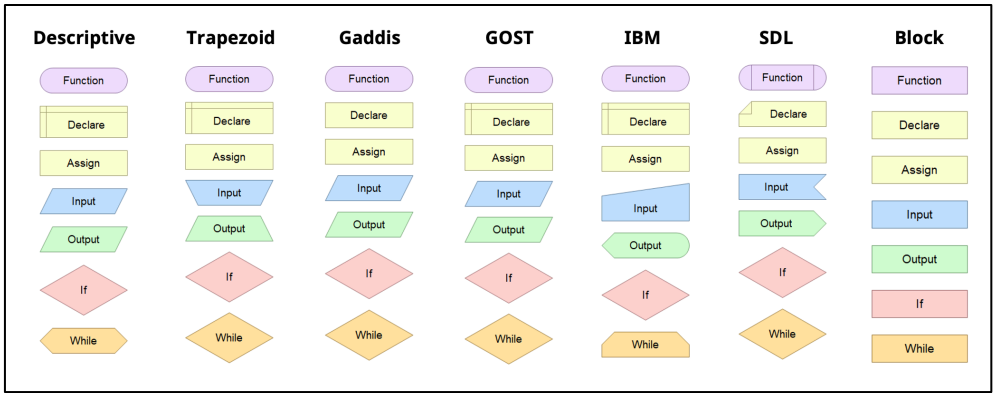
การเรียนรู้โครงสร้างแบบเรียงลำดับยังช่วยปูพื้นฐานให้ผู้เรียนเข้าใจแนวคิดการออกแบบโปรแกรมในระดับสูงขึ้น เนื่องจากโครงสร้างแบบทางเลือกและแบบทำซ้ำต่างก็อาศัยการวางลำดับขั้นตอนเป็นฐาน หากผู้เรียนยังไม่เข้าใจการจัดลำดับคำสั่งอย่างถูกต้อง การเรียนโครงสร้างที่ซับซ้อนขึ้นย่อมเกิดความสับสนได้ง่าย

สำหรับนักศึกษาครู ควรตระหนักว่าการสอนโครงสร้างแบบเรียงลำดับไม่จำเป็นต้องเริ่มจากภาษาโปรแกรมเสมอไป แต่สามารถเริ่มจากกิจกรรมที่ผู้เรียนคุ้นเคย เช่น การเรียงลำดับภาพ การจัดลำดับขั้นตอนกิจกรรม การอธิบายวิธีทำงานบ้าน หรือการให้เพื่อนทำตามคำสั่งทีละขั้น สิ่งเหล่านี้ช่วยให้ผู้เรียนเห็นภาพของการทำงานแบบเรียงลำดับอย่างเป็นรูปธรรม ก่อนที่จะเชื่อมโยงไปสู่การเขียนผังงานหรือโปรแกรม

ดังนั้น โครงสร้างการควบคุมแบบเรียงลำดับจึงเป็นรากฐานสำคัญของการเขียนโปรแกรมและการคิดเชิงระบบ เพราะช่วยให้ผู้เรียนเข้าใจว่าการทำงานของโปรแกรมต้องอาศัยคำสั่งที่ดำเนินไปอย่างเป็นลำดับและมีความสัมพันธ์กันอย่างชัดเจน

4.3 สัญลักษณ์ผังงานเบื้องต้นและการออกแบบผังงานแบบเรียงลำดับ

ผังงาน (Flowchart) คือ เครื่องมือที่ใช้แสดงลำดับขั้นตอนการทำงานของอัลกอริทึมหรือกระบวนการต่าง ๆ ด้วยสัญลักษณ์มาตรฐานและเส้นลูกศร เพื่อช่วยให้เห็นภาพรวมของการทำงานได้อย่างชัดเจน ผังงานมีประโยชน์อย่างมากต่อการเรียนวิทยาการคำนวณ เพราะช่วยให้ผู้เรียนสามารถถ่ายทอดความคิดเชิงขั้นตอนออกมาในรูปแบบที่เป็นระบบ เข้าใจง่าย และตรวจสอบได้



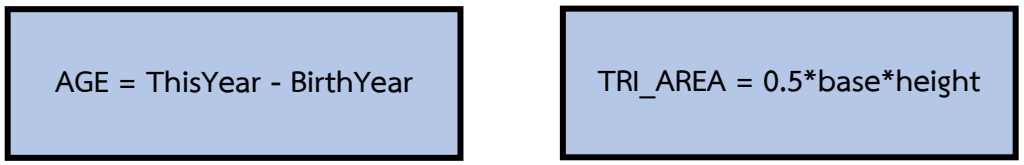
การออกแบบผังงานแบบเรียงลำดับจำเป็นต้องรู้จักสัญลักษณ์พื้นฐานที่ใช้บ่อยในงานเขียนผังงาน ได้แก่

4.3.1 สัญลักษณ์เริ่มต้นและสิ้นสุด (Terminator)

ใช้แทนจุดเริ่มต้นและจุดสิ้นสุดของการทำงาน มักมีลักษณะเป็นรูปวงรีหรือรูปมุมมน ในผังงานแบบเรียงลำดับทุกผังงานควรมีจุดเริ่มต้นและจุดสิ้นสุดอย่างชัดเจน

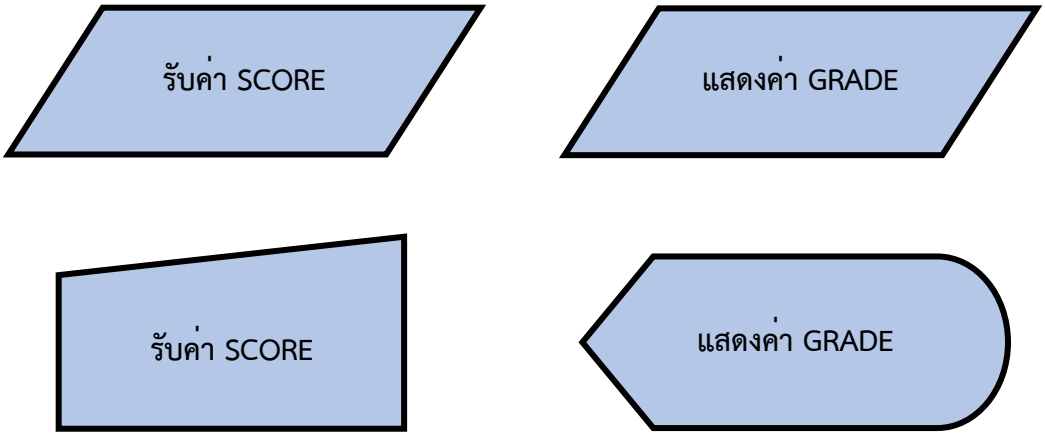
4.3.2 สัญลักษณ์กระบวนการ (Process)

ใช้แทนขั้นตอนการทำงานหรือการประมวลผล เช่น การคำนวณ การกำหนดค่าให้ตัวแปร หรือการดำเนินการใด ๆ ภายในโปรแกรม สัญลักษณ์นี้มักเป็นรูปสี่เหลี่ยมผืนผ้า



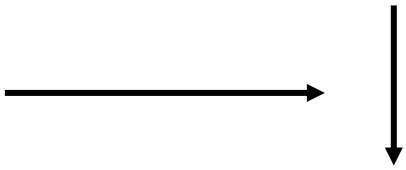
4.3.3 สัญลักษณ์ข้อมูลเข้าและข้อมูลออก (Input/Output)

ใช้แทนการรับข้อมูลจากผู้ใช้ หรือการแสดงผลลัพธ์จากโปรแกรม เช่น การรับค่าตัวเลข การแสดงผลข้อความ หรือการพิมพ์ผลลัพธ์ออกทางหน้าจอ สัญลักษณ์นี้มักเป็นรูปสี่เหลี่ยมด้านขนาน



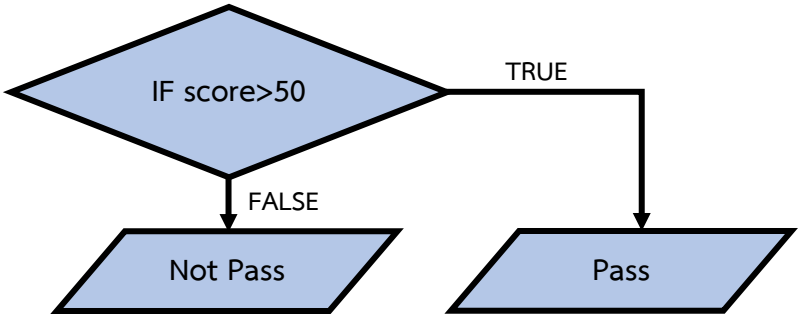
4.3.4 เส้นแสดงทิศทางการไหล (Flowline)

ใช้ลูกศรเชื่อมสัญลักษณ์ต่าง ๆ เพื่อแสดงลำดับการดำเนินงานของโปรแกรมจากขั้นตอนหนึ่งไปยังอีกขั้นตอนหนึ่ง เส้นลูกศรช่วยให้ผู้เรียนเห็นว่าขั้นตอนใดต้องทำก่อนหรือหลัง



4.3.5 สัญลักษณ์การตัดสินใจ (Decision)

ใช้แทนการตัดสินใจตามเงื่อนไข เพื่อเลือกเส้นทางที่จะทำงานต่อไป เช่น ถ้าคะแนนตั้งแต่ 80 ขึ้นไปได้เกรด A หรือ ถ้าคะแนนมากกว่า 50 แสดงว่า ผ่าน และถ้าต่ำกว่า 50 แสดงผลว่า ตก



สำหรับผังงานแบบเรียงลำดับ การออกแบบมีหลักการไม่ซับซ้อน กล่าวคือ ควรวางขั้นตอนจากบนลงล่างอย่างชัดเจน แต่ละคำสั่งควรเชื่อมต่อกันตามลำดับ ไม่มีทางแยกและไม่มีการวนกลับ ข้อความในแต่ละสัญลักษณ์ควรกระชับ ชัดเจน และสื่อความหมายตรงกับการทำงานจริง

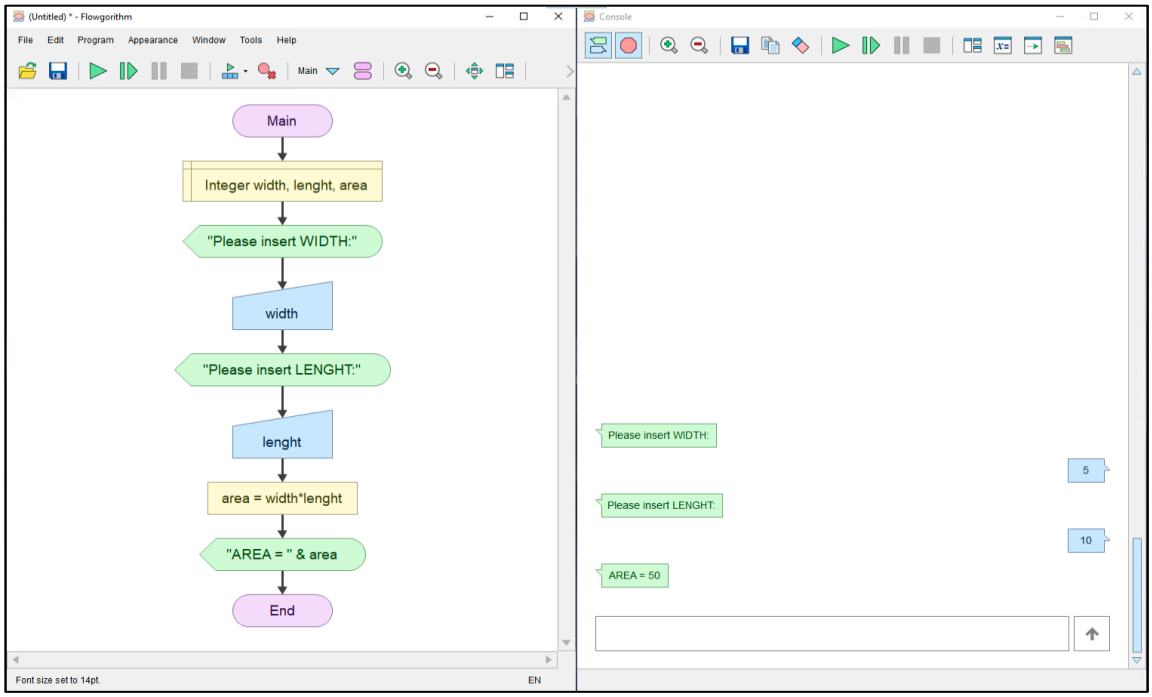
ตัวอย่างการออกแบบผังงานแบบเรียงลำดับสำหรับการคำนวณพื้นที่สี่เหลี่ยมผืนผ้า อาจประกอบด้วยขั้นตอนดังนี้

1. เริ่มต้น
2. รับค่าความกว้าง (width)
3. รับค่าความยาว (length)
4. คำนวณพื้นที่ = ความกว้าง × ความยาว

   area = width * length

5. แสดงผลพื้นที่ (area)
6. สิ้นสุด

เมื่อนำขั้นตอนดังกล่าวไปเขียนเป็นผังงาน ผู้เรียนจะเห็นภาพการทำงานอย่างชัดเจนว่าโปรแกรมรับข้อมูลก่อน แล้วจึงประมวลผล และสุดท้ายจึงแสดงผลลัพธ์ ซึ่งสอดคล้องกับหลักการของโครงสร้างแบบเรียงลำดับ



การฝึกออกแบบผังงานมีประโยชน์ต่อผู้เรียนหลายด้าน ได้แก่ ช่วยฝึกการคิดอย่างมีลำดับ ช่วยลดความผิดพลาดก่อนลงมือเขียนโปรแกรมจริง และช่วยให้ครูสามารถตรวจสอบความเข้าใจของผู้เรียนได้ง่ายขึ้น หากผังงานมีลำดับผิดพลาดหรือขาดขั้นตอนบางส่วน ครูจะสามารถให้ข้อเสนอแนะได้ตั้งแต่ต้น สำหรับนักศึกษาครู การใช้ผังงานเป็นเครื่องมือในการจัดการเรียนรู้มีความเหมาะสมอย่างยิ่ง เพราะช่วยให้ผู้เรียนในระดับประถมศึกษาและมัธยมศึกษาเข้าใจหลักการทำงานของโปรแกรมโดยไม่ต้องเริ่มจากโค้ดซึ่งอาจมีความซับซ้อนเกินไป ครูสามารถใช้ผังงานร่วมกับภาพ กิจกรรมการ์ดลำดับขั้น หรือสถานการณ์ใกล้ตัว เพื่อสร้างความเข้าใจเรื่องอัลกอริทึมอย่างเป็นรูปธรรม

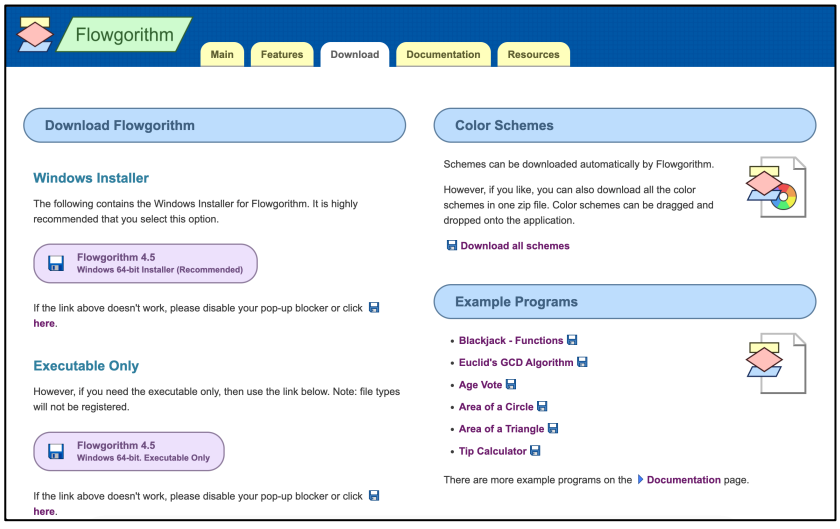
กล่าวโดยสรุป ผังงานเป็นเครื่องมือสำคัญในการแสดงลำดับขั้นตอนการทำงานของอัลกอริทึม โดยเฉพาะโครงสร้างแบบเรียงลำดับ ซึ่งช่วยให้ผู้เรียนสามารถเห็นภาพการทำงานของโปรแกรมอย่างชัดเจน เป็นระบบ และพร้อมต่อยอดไปสู่การเขียนโปรแกรมจริงในลำดับถัดไป

4.4 การใช้โปรแกรม Flowgorithm ในการเขียนและทดสอบผังงานแบบเรียงลำดับ

Flowgorithm เป็นโปรแกรมที่ออกแบบมาเพื่อช่วยให้ผู้เรียนสามารถสร้างผังงานและทดสอบการทำงานของอัลกอริทึมได้อย่างสะดวก โดยเฉพาะสำหรับผู้เริ่มต้นเรียนเขียนโปรแกรม โปรแกรมนี้มีจุดเด่นคือ ใช้การออกแบบผังงานแทนการพิมพ์โค้ดโดยตรง ทำให้ผู้เรียนมองเห็นลำดับขั้นตอนการทำงานได้อย่างชัดเจน และสามารถเข้าใจความสัมพันธ์ระหว่างการรับข้อมูล การประมวลผล และการแสดงผลได้ดียิ่งขึ้น

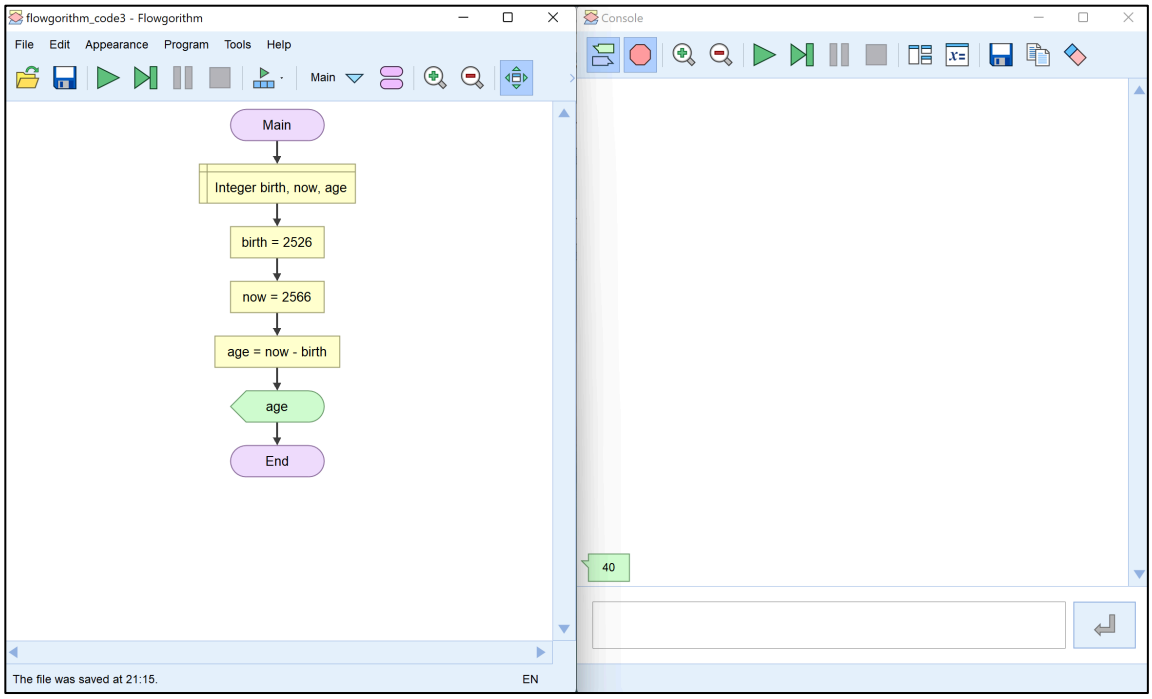
การใช้ Flowgorithm มีความเหมาะสมอย่างมากในรายวิชาวิทยาการคำนวณสำหรับนักศึกษาครู เพราะเป็นเครื่องมือที่ช่วยเชื่อมโยงระหว่างแนวคิดอัลกอริทึมกับการปฏิบัติจริง ผู้เรียนสามารถทดลองสร้างผังงาน แก้ไขขั้นตอน ตรวจสอบผลลัพธ์ และเรียนรู้จากข้อผิดพลาดได้ทันที จึงเป็นสื่อการเรียนรู้ที่ส่งเสริมการคิดเชิงระบบและการเรียนรู้แบบลงมือปฏิบัติ

การใช้งานโปรแกรม Flowgorithm สามารถดาวน์โหลดโปรแกรมเพื่อติดตั้งได้ที่เว็บไซต์ http://www.flowgorithm.org/download/index.html โดยควรเลือกรุ่นของโปรแกรมที่เป็นรุ่นล่าสุด



4.4.1 ส่วนประกอบเบื้องต้นของโปรแกรม Flowgorithm

เมื่อเปิดโปรแกรม Flowgorithm ผู้เรียนจะพบพื้นที่สำหรับวางผังงาน แถบเครื่องมือสำหรับเพิ่มสัญลักษณ์ เช่น การรับข้อมูล การแสดงผล และการประมวลผล รวมถึงคำสั่งสำหรับรันหรือทดสอบผังงาน โปรแกรมจะแสดงจุดเริ่มต้นและสิ้นสุดไว้ให้เบื้องต้น เพื่อให้ผู้เรียนสามารถเพิ่มเติมขั้นตอนการทำงานระหว่างกลางได้ตามต้องการ

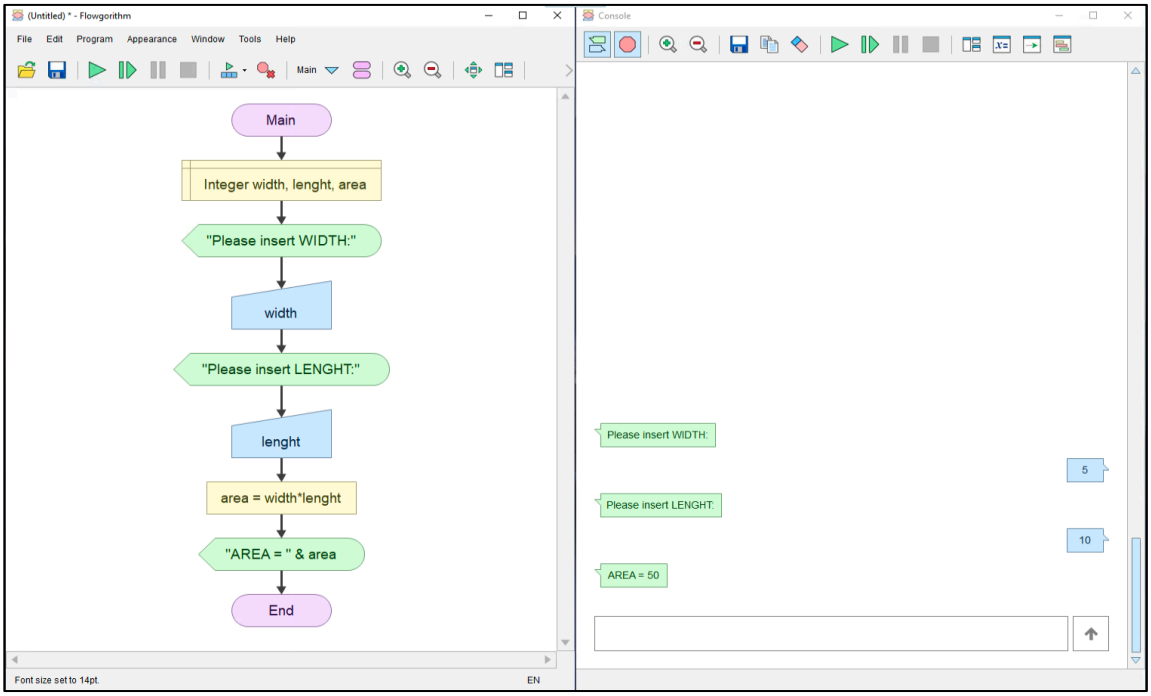


4.4.2 ขั้นตอนการสร้างผังงานแบบเรียงลำดับ

การสร้างผังงานแบบเรียงลำดับใน Flowgorithm โดยทั่วไปประกอบด้วยขั้นตอนดังนี้

1. กำหนดปัญหาหรือโจทย์ที่ต้องการแก้
2. วิเคราะห์ข้อมูลนำเข้า กระบวนการ และผลลัพธ์
3. เพิ่มสัญลักษณ์รับข้อมูล หากโจทย์ต้องรับค่าจากผู้ใช้
4. เพิ่มสัญลักษณ์กระบวนการ เพื่อคำนวณหรือกำหนดค่าตัวแปร
5. เพิ่มสัญลักษณ์แสดงผล เพื่อสื่อสารผลลัพธ์ออกทางหน้าจอ
6. ตรวจสอบความถูกต้องของลำดับขั้นตอนก่อนทดสอบการทำงาน

ตัวอย่างเช่น หากต้องการเขียนผังงานเพื่อคำนวณพื้นที่สี่เหลี่ยมผืนผ้า ผู้เรียนอาจกำหนดตัวแปร width, length และ area จากนั้นสร้างผังงานดังนี้ รับค่าความกว้าง รับค่าความยาว คำนวณ area = width × length และแสดงค่า area



4.4.3 การทดสอบหรือ Run ผังงาน

จุดเด่นสำคัญของ Flowgorithm คือ ผู้เรียนสามารถทดสอบการทำงานของผังงานได้ทันที โดยโปรแกรมจะทำงานตามลำดับขั้นตอนที่ออกแบบไว้ หากมีการรับข้อมูล โปรแกรมจะเปิดช่องให้ผู้ใช้ป้อนค่า และเมื่อถึงขั้นตอนการแสดงผล ก็จะแสดงคำตอบตามกระบวนการที่กำหนดไว้ การทดสอบเช่นนี้ช่วยให้ผู้เรียนเห็นว่าผังงานที่ตนเขียนสามารถนำไปสู่ผลลัพธ์จริงได้หรือไม่

4.4.4 การตรวจสอบข้อผิดพลาดและการปรับปรุงผังงาน

เมื่อผู้เรียน Run ผังงาน อาจพบข้อผิดพลาดหลายลักษณะ เช่น ลำดับขั้นตอนไม่ถูกต้อง ลืมกำหนดตัวแปร ใช้สูตรคำนวณผิด หรือแสดงผลผิดจากที่คาดไว้ การใช้ Flowgorithm ช่วยให้ผู้เรียนตรวจสอบข้อผิดพลาดได้ง่ายขึ้น เพราะสามารถมองย้อนกลับไปยังผังงานทีละขั้นตอน และปรับปรุงแก้ไขได้อย่างเป็นระบบ

4.4.5 ประโยชน์ของ Flowgorithm ต่อการจัดการเรียนรู้

Flowgorithm มีประโยชน์หลายประการต่อการเรียนการสอน ได้แก่ ช่วยลดความซับซ้อนของการเริ่มต้นเรียนเขียนโปรแกรม ช่วยให้ผู้เรียนเห็นภาพกระบวนการทำงานอย่างชัดเจน ช่วยส่งเสริมการเรียนรู้จากการทดลองและการตรวจสอบผล และช่วยเตรียมความพร้อมก่อนเข้าสู่การเขียนโค้ดด้วยภาษาโปรแกรมจริง นอกจากนี้ ยังเป็นเครื่องมือที่เหมาะสมสำหรับครูในการอธิบายอัลกอริทึมให้ผู้เรียนเข้าใจได้ง่ายขึ้น

สำหรับนักศึกษาครู การเรียนรู้การใช้ Flowgorithm ควรครอบคลุมทั้งการใช้งานเครื่องมือและการพิจารณาว่าจะนำไปออกแบบกิจกรรมการเรียนรู้สำหรับผู้เรียนอย่างไร เช่น การให้ผู้เรียนช่วยกันวิเคราะห์โจทย์ก่อนสร้างผังงาน การให้คาดการณ์ผลลัพธ์ก่อน Run หรือการให้อภิปรายข้อผิดพลาดที่เกิดขึ้นจากผังงานของตนและเพื่อน กิจกรรมเช่นนี้จะช่วยพัฒนาทั้งความรู้ด้านวิทยาการคำนวณและทักษะการอธิบายของผู้เรียน

กล่าวโดยสรุป Flowgorithm เป็นเครื่องมือที่มีประสิทธิภาพสำหรับการเรียนรู้และการสอนโครงสร้างแบบเรียงลำดับ เพราะช่วยให้ผู้เรียนสามารถแปลงอัลกอริทึมเป็นผังงานที่มองเห็นได้ ทดสอบผลลัพธ์ได้ และเรียนรู้จากข้อผิดพลาดได้อย่างเป็นรูปธรรม

4.5 การออกแบบกิจกรรมการเรียนรู้เพื่อสอนโครงสร้างแบบเรียงลำดับในระดับประถมศึกษาและมัธยมศึกษา

การจัดการเรียนรู้เรื่องโครงสร้างการควบคุมแบบเรียงลำดับสำหรับผู้เรียนระดับประถมศึกษาและมัธยมศึกษา ควรคำนึงถึงพัฒนาการและประสบการณ์เดิมของผู้เรียนเป็นสำคัญ แม้เนื้อหาเรื่องอัลกอริทึมและผังงานจะเป็นพื้นฐานทางวิทยาการคำนวณ แต่หากครูนำเสนอในรูปแบบที่เป็นนามธรรมมากเกินไป ผู้เรียนอาจรู้สึกว่าเนื้อหายากและห่างไกลจากชีวิตจริง ดังนั้น การออกแบบกิจกรรมการเรียนรู้จึงควรเริ่มจากสิ่งที่เป็นรูปธรรม ใกล้ตัว และเปิดโอกาสให้ผู้เรียนได้ลงมือปฏิบัติจริง

4.5.1 หลักการออกแบบกิจกรรม

ครูควรเริ่มต้นจากสถานการณ์หรือกิจกรรมในชีวิตประจำวัน เช่น การล้างมือ การทำแซนด์วิช การแต่งกายไปโรงเรียน หรือการปลูกต้นไม้ เพื่อให้ผู้เรียนเห็นว่ากิจกรรมต่าง ๆ ล้วนมีลำดับขั้นตอน หากสลับขั้นตอนอาจทำให้ผลลัพธ์ผิดพลาดได้ แนวทางนี้ช่วยให้ผู้เรียนเข้าใจแนวคิดของโครงสร้างแบบเรียงลำดับก่อนที่จะเชื่อมโยงไปสู่ผังงานและโปรแกรม

4.5.2 แนวทางสำหรับระดับประถมศึกษา

ในระดับประถมศึกษา ควรใช้กิจกรรมที่เน้นภาพ การลงมือทำ และความสนุกสนาน เช่น

1. กิจกรรมเรียงภาพลำดับขั้นตอน
2. การเขียนขั้นตอนจากกิจวัตรประจำวัน
3. เกมสั่งเพื่อนทำตามคำสั่งทีละขั้น
4. การใช้บัตรคำสั่งแทนลำดับกิจกรรม
5. การวาดผังงานอย่างง่ายจากสถานการณ์ใกล้ตัว

กิจกรรมเหล่านี้ช่วยให้ผู้เรียนเข้าใจว่า การทำงานอย่างเป็นลำดับเป็นพื้นฐานของทั้งชีวิตประจำวันและการเขียนโปรแกรม

4.5.3 แนวทางสำหรับระดับมัธยมศึกษา

ในระดับมัธยมศึกษา ผู้เรียนสามารถเรียนรู้ในระดับที่เป็นนามธรรมมากขึ้น ครูอาจออกแบบกิจกรรม เช่น

1. วิเคราะห์โจทย์แล้วระบุ Input, Process, Output
2. เขียนอัลกอริทึมเป็นภาษาธรรมชาติหรือรหัสลำลอง
3. เขียนผังงานแบบเรียงลำดับจากโจทย์คำนวณอย่างง่าย
4. ใช้โปรแกรม Flowgorithm เพื่อสร้างและ Run ผังงาน
5. ให้ผู้เรียนปรับปรุงผังงานจากกรณีตัวอย่างที่ผิดพลาด

กิจกรรมระดับนี้ควรมุ่งให้ผู้เรียนเข้าใจความสัมพันธ์ระหว่างการคิด การออกแบบผังงาน และผลลัพธ์ของโปรแกรมอย่างชัดเจน

4.5.4 การบูรณาการกับการจัดทำแผนการเรียนรู้

สำหรับนักศึกษาครู การออกแบบกิจกรรมไม่ควรหยุดเพียงการเลือกกิจกรรม แต่ควรเชื่อมโยงกับการจัดทำแผนการจัดการเรียนรู้ เช่น การกำหนดจุดประสงค์การเรียนรู้ การเลือกสื่อและอุปกรณ์ การกำหนดขั้นตอนกิจกรรม การวัดและประเมินผล และการสะท้อนผลการเรียนรู้ของผู้เรียน ครูควรออกแบบให้ผู้เรียนได้ทั้งความรู้ ทักษะ และเจตคติที่ดีต่อการเรียนวิทยาการคำนวณ

4.5.5 การวัดและประเมินผล

การประเมินผลการเรียนรู้เรื่องโครงสร้างแบบเรียงลำดับควรครอบคลุมทั้งความเข้าใจและการปฏิบัติ เช่น การตรวจลำดับขั้นตอนที่ผู้เรียนเขียน การประเมินผังงานที่ออกแบบ การสังเกตการทำกิจกรรม การตรวจผลลัพธ์จากการ Run ผังงาน และการให้ผู้เรียนอธิบายเหตุผลของลำดับขั้นตอนที่ตน

กำหนด การประเมินลักษณะนี้จะช่วยให้ครูเห็นพัฒนาการของผู้เรียนได้ชัดเจนกว่าการดูคำตอบสุดท้ายเพียงอย่างเดียว

4.5.6 บทบาทของครู

ครูควรทำหน้าที่เป็นผู้อำนวยความสะดวกในการเรียนรู้ หรือที่เรียกว่า Facilitator โดยใช้คำถามกระตุ้นการคิด เช่น “ถ้าสลับขั้นตอนนี้จะเกิดอะไรขึ้น” หรือ “เหตุใดต้องรับข้อมูลก่อนคำนวณ” คำถามลักษณะนี้ช่วยให้ผู้เรียนไม่ได้เพียงจำลำดับขั้นตอน แต่เข้าใจเหตุผลเบื้องหลังการจัดลำดับของโปรแกรมด้วย

โดยสรุป การสอนโครงสร้างแบบเรียงลำดับควรมุ่งให้ผู้เรียนเข้าใจผ่านกิจกรรมที่เป็นรูปธรรมและค่อย ๆ เชื่อมไปสู่การใช้ผังงานและโปรแกรม Flowgorithm ทั้งนี้ นักศึกษาครูควรได้รับการฝึกฝนให้สามารถออกแบบกิจกรรมการเรียนรู้ที่เหมาะสมกับระดับชั้นของผู้เรียน เพื่อให้การเรียนวิทยาการคำนวณเป็นเรื่องเข้าใจได้ สนุก และมีความหมายต่อชีวิต

บทสรุปประจำบทที่ 4

โครงสร้างการควบคุมโปรแกรมแบบเรียงลำดับเป็นพื้นฐานสำคัญของการเรียนรู้วิทยาการคำนวณและการเขียนโปรแกรม เพราะเป็นรูปแบบการทำงานที่คำสั่งต่าง ๆ ถูกดำเนินไปตามลำดับอย่างชัดเจนจากต้นจนจบ การทำความเข้าใจอัลกอริทึมและการแสดงลำดับขั้นตอนการทำงานจึงเป็นจุดเริ่มต้นที่สำคัญสำหรับผู้เรียนในการพัฒนาทักษะการคิดเชิงระบบและการแก้ปัญหาอย่างมีขั้นตอน

ผังงานเป็นเครื่องมือที่ช่วยให้ผู้เรียนมองเห็นโครงสร้างการทำงานของอัลกอริทึมได้อย่างเป็นรูปธรรม โดยเฉพาะเมื่อใช้ร่วมกับโปรแกรม Flowgorithm ซึ่งช่วยให้สามารถสร้างและทดสอบผังงานได้ทันที ผู้เรียนจึงสามารถเชื่อมโยงระหว่างแนวคิด อัลกอริทึม ผังงาน และผลลัพธ์ของโปรแกรมได้อย่างชัดเจน นอกจากนี้ สำหรับนักศึกษาครู การเรียนรู้เรื่องโครงสร้างแบบเรียงลำดับยังควรเชื่อมโยงไปสู่การออกแบบกิจกรรมการเรียนรู้ที่เหมาะสมกับผู้เรียนในระดับประถมศึกษาและมัธยมศึกษา โดยเน้นการใช้สถานการณ์ใกล้ตัว การลงมือปฏิบัติ และการวัดผลที่ครอบคลุมทั้งความรู้และกระบวนการคิด

คำถามท้ายบท

1. จงอธิบายความหมายของอัลกอริทึม และเหตุใดอัลกอริทึมจึงมีความสำคัญต่อการเขียนโปรแกรม
2. อัลกอริทึมที่ดีควรมีลักษณะสำคัญอย่างไรบ้าง
3. จงอธิบายหลักการของโครงสร้างการควบคุมแบบเรียงลำดับ พร้อมยกตัวอย่าง
4. เพราะเหตุใดลำดับของคำสั่งจึงมีความสำคัญต่อการทำงานของโปรแกรมแบบเรียงลำดับ
5. สัญลักษณ์พื้นฐานของผังงานมีอะไรบ้าง และแต่ละสัญลักษณ์มีหน้าที่อย่างไร
6. จงอธิบายประโยชน์ของการใช้ผังงานก่อนการเขียนโปรแกรม
7. โปรแกรม Flowgorithm มีประโยชน์อย่างไรต่อการเรียนรู้เรื่องอัลกอริทึมและผังงาน
8. หากให้ท่านออกแบบผังงานเพื่อคำนวณผลรวมของตัวเลข 2 จำนวน ท่านจะกำหนดลำดับขั้นตอนอย่างไร
9. หากท่านเป็นครูระดับประถมศึกษา ท่านจะจัดกิจกรรมอย่างไรเพื่อให้ผู้เรียนเข้าใจโครงสร้างแบบเรียงลำดับ
10. หากท่านเป็นครูระดับมัธยมศึกษา ท่านจะใช้ Flowgorithm เพื่อส่งเสริมการเรียนรู้ของผู้เรียนอย่างไร

เอกสารอ้างอิง

กระทรวงศึกษาธิการ. (2560). *ตัวชี้วัดและสาระการเรียนรู้แกนกลาง กลุ่มสาระการเรียนรู้วิทยาศาสตร์ (ฉบับปรับปรุง พ.ศ. 2560) ตามหลักสูตรแกนกลางการศึกษาขั้นพื้นฐาน พุทธศักราช 2551*. กรุงเทพมหานคร: ชุมนุมสหกรณ์การเกษตรแห่งประเทศไทย.

Gaddis, T. (2018). *Starting out with programming logic and design* (5th ed.). Boston, MA: Pearson.

Joyce, D. (2020). *Programming logic and design: Introductory* (9th ed.). Boston, MA: Cengage Learning.

Rapaport, W. J. (2005). Algorithmic thinking, computational thinking, and the humanities. *Journal of Computing Sciences in Colleges, 20*(5), 1–9.

Flowgorithm. (2024). *Flowgorithm documentation*. Retrieved from the official Flowgorithm documentation resources.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35.