



โรงเรียนสตรีศึกษา

แผนการจัดการเรียนรู้ รายวิชาวิทยาการคำนวณ (ว3011) ระดับชั้นมัธยมศึกษาปีที่ 4



ผู้จัดทำ :
โดย นายสิทธิชัย ทิพย์สิงห์
ตำแหน่งครู วิทยฐานะครูชำนาญการพิเศษ

โรงเรียนสตรีศึกษา

สำนักงานเขตพื้นที่การศึกษามัธยมศึกษาร้อยเอ็ด
สำนักงานคณะกรรมการการศึกษาขั้นพื้นฐาน
กระทรวงศึกษาธิการ

แผนการจัดการเรียนรู้ที่ 1

กลุ่มสาระการเรียนรู้วิทยาศาสตร์และเทคโนโลยี (วิทยาการคำนวณ)

หน่วยการเรียนรู้ที่ 1: พื้นฐานแนวคิดเชิงคำนวณ (Coding as a Language)

เรื่อง: แนวคิดเชิงคำนวณและอัลกอริทึมในชีวิตประจำวัน

ระดับชั้น: มัธยมศึกษาปีที่ 4 (สายศิลป์-ภาษา/สังคม) เวลา: 2 คาบ (100 นาที)

1. มาตรฐานการเรียนรู้/ตัวชี้วัด

ว 4.2 ม.4/1: ประยุกต์ใช้แนวคิดเชิงคำนวณในการพัฒนาโครงงานที่มีการบูรณาการกับวิชาอื่นอย่างสร้างสรรค์ และเชื่อมโยงกับชีวิตจริง

2. สาระสำคัญ (Concept)

"การเขียนโปรแกรม ก็เหมือนการเรียงประโยค" แนวคิดเชิงคำนวณ (Computational Thinking) ไม่ใช่การคิดเหมือนหุ่นยนต์ แต่คือกระบวนการคิดวิเคราะห์ แก้ปัญหา และถ่ายทอดออกมาเป็นลำดับขั้นตอนที่ชัดเจน (Algorithm) เหมือนกับ "ไวยากรณ์ (Grammar)" ของภาษา ที่ต้องวางโครงสร้าง ประธาน-กริยา-กรรม ให้ถูกต้อง เพื่อให้สื่อสารเรื่อง การเข้าใจหลักการนี้จะช่วยให้เราจัดการปัญหาที่ซับซ้อนในชีวิตจริงได้ง่ายขึ้น

3. จุดประสงค์การเรียนรู้ (K-P-A)

ด้านความรู้ (K): อธิบายองค์ประกอบของแนวคิดเชิงคำนวณ (Decomposition, Pattern Recognition, Abstraction, Algorithm) โดยเปรียบเทียบกับหลักภาษาหรือสถานการณ์จริงได้

ด้านทักษะกระบวนการ (P): สามารถเขียนอัลกอริทึม (ลำดับขั้นตอน) ในการแก้ปัญหาชีวิตประจำวัน หรือกิจกรรมในโรงเรียนได้

ด้านคุณลักษณะ (A): เห็นประโยชน์ของการคิดอย่างเป็นระบบในการทำงานและการเรียน

4. สาระการเรียนรู้

1. แนวคิดเชิงคำนวณ (Computational Thinking):

- Decomposition (การแยกส่วนประกอบ) -> เปรียบเทียบกับการวิเคราะห์โครงสร้างประโยค
- Pattern Recognition (การหารูปแบบ) -> เปรียบเทียบกับ Tense หรือ ฉันทลักษณ์กลอน
- Abstraction (แนวคิดเชิงนามธรรม) -> เปรียบเทียบกับการย่อความ/จับใจความสำคัญ
- Algorithm Design (การออกแบบขั้นตอนวิธี) -> เปรียบเทียบกับการเขียนเรียงความ/ขั้นตอน

การแปล

2. การเขียนอัลกอริทึมด้วยภาษาธรรมชาติ (Natural Language): การเขียนลำดับงานเป็นข้อๆ

5. กิจกรรมการเรียนรู้ (รูปแบบ Active Learning)

ขั้นนำ (Introduction) - 15 นาที

1. กิจกรรม "Lost in Translation":

- ครูยกประโยคภาษาไทยที่ซับซ้อน หรือสำนวนไทย (เช่น "น้ำพึ่งเรือ เสือพึ่งป่า") แล้วถามนักเรียนว่า "ถ้าต้องอธิบายให้เพื่อนชาวต่างชาติเข้าใจ เราจะแปลตรงตัวเลยได้ไหม? หรือต้องมีขั้นตอนคิดอย่างไร?"

- แนวคำตอบ: ต้องแยกแยะความหมาย (Decomposition) -> หาคำศัพท์ที่ตรงกัน (Pattern) -> ตัดคำฟุ่มเฟือยออกเอาแค่ใจความ (Abstraction) -> เรียบเรียงเป็นประโยคใหม่ (Algorithm)

2. ครูเชื่อมโยงว่า "กระบวนการคิดในหัวของพวกเราตอนแปลภาษา นั้นแหละคือ วิทยาการคำนวณ"

ขั้นสอน (Instruction) - 35 นาที

Decomposition (แยกส่วนประกอบ): ครูยกตัวอย่างงานกีฬา

ถาม: "ถ้าครูสั่งให้จัดกีฬา 1 งาน มันดูใหญ่มาก เราจะทำยังไง?"

อภิปราย: เราต้องแตกงานออกเป็น ฝ่ายสถานที่, ฝ่ายการแสดง, ฝ่ายสวัสดิการ ฯลฯ (เหมือนการแยกประโยคเป็น ประธาน, ขยายประธาน, กริยา, กรรม)

Pattern Recognition (หารูปแบบ):

ถาม: "เพลงเชียร์กีฬาปีนี้ กับปีที่แล้ว มีอะไรเหมือนกัน?" (จังหวะ 3 ซ้ำ, การตบมือ)

สรุป: การจำรูปแบบเดิมได้ ช่วยให้เราทำงานใหม่ได้เร็วขึ้น ไม่ต้องเริ่มจากศูนย์

Abstraction (คิดเชิงนามธรรม):

ถาม: "เวลาเพื่อนมาเล่าเรื่องดราม่าให้ฟังยาวเหยียด เราจับประเด็นยังไง?" (ใคร ทำอะไร ที่ไหน ผลเป็นยังไง ส่วนรายละเอียดสี่เสี้ยวข้างมัน)

สรุป: การตัดรายละเอียดที่ไม่จำเป็นออก คือหัวใจของการแก้ปัญหา

Algorithm (ขั้นตอนวิธี):

ครูอธิบายว่า อัลกอริทึม คือ "สูตรสำเร็จ" หรือ "คู่มือ How-to" ที่ใครทำตามก็ได้ผลลัพธ์เหมือนกัน

ขั้นปฏิบัติ (Practice) - 40 นาที

กิจกรรมกลุ่ม: "The Master Planner" (นักวางแผนอัจฉริยะ)

1. แบ่งกลุ่มนักเรียน (คละความสามารถ)

2. ให้แต่ละกลุ่มเลือกโจทย์สถานการณ์ 1 ข้อ (เน้นเรื่องใกล้ตัวเด็กสายศิลป์):

โจทย์ A: วางแผนขั้นตอนการ "จีบ" รุ่นพี่ที่ชอบให้ติด (Algorithm of Love)

โจทย์ B: วางแผนขั้นตอนการ "เตรียมตัวสอบ HSK/JLPT/TOEIC" ให้ผ่านใน 1 เดือน

โจทย์ C: วางแผนขั้นตอนการ "ทำคอนเทนต์ TikTok" ให้แมส

3. ให้เขียนกระบวนการคิดออกมาในรูปแบบ Story Board หรือ Checklist โดยระบุว่าใช้ 4 องค์ประกอบ CT ตรงไหนบ้าง

ตัวอย่าง: จีบรุ่นพี่ -> แยกส่วนประกอบ (สืบความชอบ, สร้างสถานการณ์) -> หารูปแบบ (หักแซทเวลาเดิมทุกวัน) -> ตัดรายละเอียด (เลิกคิดมากเรื่องเล็กน้อย) -> ขั้นตอนปฏิบัติ (1. ทัก 2. ชวนคุย 3. ชวนกินข้าว)

ขั้นสรุป (Conclusion) - 10 นาที

1. สุ่มตัวแทนกลุ่มออกมานำเสนอสั้นๆ
2. ครูสรุปจบ: "วันนี้เราไม่ได้แตะคอมพิวเตอร์เลย แต่เราได้ใช้ Computational Thinking แล้วต่อไปเมื่อเราเริ่มเขียนโค้ด ให้จำความรู้สึกของการวางแผนวันนี้ไว้ เพราะการเขียนโค้ด คือการเขียนแผนงานให้คอมพิวเตอร์ทำตามเรา"

6. สื่อ/แหล่งการเรียนรู้

1. สไลด์ประกอบการสอน (เทียบเคียงหลักภาษา กับ CT)
2. ใบงานกิจกรรม "The Master Planner"
3. ภาพประกอบการทำงานของสมองเวลามนุษย์แก้ปัญหา

7. การวัดและประเมินผล

จุดประสงค์การเรียนรู้	วิธีการวัดผล	เครื่องมือวัดผล	เกณฑ์การประเมิน
1. ด้านความรู้ (K) อธิบายองค์ประกอบของแนวคิดเชิงคำนวณ (4 เสาหลัก) โดยเปรียบเทียบกับสถานการณ์จริงได้	ตรวจการวิเคราะห์ในใบงานที่ 1 (ส่วนที่ 1-3: แยกส่วนประกอบ, หารูปแบบ, คิดเชิงนามธรรม)	ใบงานที่ 1 "The Master Planner"	วิเคราะห์ได้สมเหตุผล และครบถ้วนทั้ง 3 ส่วน (Decomposition, Pattern, Abstraction) ถือว่า ผ่าน
2. ด้านทักษะกระบวนการ (P) เขียนอัลกอริทึม (ลำดับขั้นตอน) ในการแก้ปัญหาชีวิตประจำวันได้	ตรวจลำดับขั้นตอนการแก้ปัญหาในใบงานที่ 1 (ส่วนที่ 4: Algorithm Design)	ใบงานที่ 1 "The Master Planner"	ระดับคุณภาพพอใช้ขึ้นไป (เกณฑ์: ขั้นตอนเรียงลำดับถูกต้อง, อ่านเข้าใจง่าย, สามารถนำไปปฏิบัติตามได้จริง)
3. ด้านคุณลักษณะ (A) เห็นประโยชน์ของการคิดอย่างเป็นระบบ และทำงานร่วมกับผู้อื่นได้	สังเกตพฤติกรรมการทำงานกลุ่มและการนำเสนอผลงาน	แบบสังเกตพฤติกรรม	สมาชิกในกลุ่มมีส่วนร่วมในการระดมสมองและยอมรับฟังความคิดเห็นซึ่งกันและกัน ในระดับ ผ่าน

เกณฑ์การให้คะแนนชิ้นงาน (Rubric) - ใบงาน The Master Planner:

ดีมาก (3 คะแนน):

- ระบุองค์ประกอบ CT ได้ครบถ้วนชัดเจน
- เขียนอัลกอริทึมเป็นลำดับขั้นตอนที่ละเอียดผู้อื่นอ่านแล้วทำตามได้ทันทีโดยไม่สับสน
- เนื้อหามีความคิดสร้างสรรค์และน่าสนใจ

พอใช้ (2 คะแนน):

- ระบุองค์ประกอบ CT ได้ถูกต้องเป็นส่วนใหญ่
- เขียนอัลกอริทึมได้เป็นลำดับแต่บางขั้นตอนอาจยังคลุมเครือหรือไม่ละเอียด

ปรับปรุง (1 คะแนน):

- ไม่สามารถแยกแยะองค์ประกอบ CT ได้ (เช่น สับสนระหว่างการแยกส่วนประกอบกับ

การหารูปแบบ)

- เขียนอัลกอริทึมสลับขั้นตอน หรือข้ามขั้นตอนสำคัญจนไม่สามารถปฏิบัติตามได้

ใบงานที่ 1: The Master Planner (นักวางแผนอัจฉริยะ)

รายวิชา: วิทยาการคำนวณ (ว31101) | ระดับชั้น: มัธยมศึกษาปีที่ 4 /

ชื่อสมาชิกกลุ่ม:

คำชี้แจง: จงเลือก "ภารกิจ" ที่กลุ่มของนักเรียนสนใจที่สุด 1 หัวข้อ จากตัวเลือกด้านล่าง แล้วใช้พลังแห่ง Computational Thinking (แนวคิดเชิงคำนวณ) เพื่อวางแผนพิชิตภารกิจนั้นให้สำเร็จ!

เลือกภารกิจของคุณ (ติ๊ก ✓):

- Mission A: วางแผนพิชิตใจรุ่นพี่/รุ่นน้อง ที่แอบชอบ (Love Algorithm)
- Mission B: วางแผนเตรียมสอบวัดระดับภาษา (HSK / JLPT / TOEIC) ให้ผ่านฉลุย
- Mission C: วางแผนปั้นช่อง TikTok / Instagram ให้ยอดไลก์พุ่ง (Viral Planner)
- Mission D: (คิดเอง)

ส่วนที่ 1: แยกส่วนประกอบ (Decomposition)

ภารกิจนี้มันใหญ่เกินไป! ลอง "หั่น" ภารกิจนี้ออกมาเป็นเรื่องย่อย ๆ 3-4 เรื่องที่เราต้องจัดการสิ

- ส่วนที่ 1:
- ส่วนที่ 2:
- ส่วนที่ 3:
- ส่วนที่ 4:

ส่วนที่ 2: ทหารูปแบบ (Pattern Recognition)

ลองนึกดูสิว่า เราเคยเจอเรื่องแบบนี้มาก่อนไหม? หรือมี "สูตรสำเร็จ" อะไรที่คนอื่นเขาทำแล้วเวิร์กบ้าง?

(เช่น ทักแชทเวลาเดิมทุกวัน, ท่องศัพท์ด้วย Flashcard, ใช้เพลงฮิตใน TikTok)

รูปแบบที่พบ/เทคนิคที่ใช้:

.....
.....

ส่วนที่ 3: คิดเชิงนามธรรม (Abstraction)

โฟกัสจุดจุด ชีวิตเปลี่ยน! อะไรคือ "หัวใจสำคัญ" ที่ต้องทำจริงๆ และอะไรคือ "สิ่งรบกวน" ที่ควรตัดทิ้งหรือมองข้ามไปบ้าง?

สิ่งที่ต้องโฟกัส (Focus):

สิ่งที่ควรตัดทิ้ง/ข้ามมัน (Ignore):

ส่วนที่ 4: ออกแบบขั้นตอน (Algorithm Design)

เขียนลำดับขั้นตอนการปฏิบัติการ (How-to) แบบ Step-by-Step ให้เพื่อนเอาไปทำตามแล้วได้ผลลัพธ์ เหมือนกันเป๊ะๆ

.....

.....

.....

.....

.....

ชวนคิดหลังกิจกรรม (Reflection)

นักเรียนคิดว่าการวางแผนในใบงานนี้ เหมือนหรือต่างกับการ "แต่งประโยค" หรือ "เขียนเรียงความ" ในวิชา ภาษาอย่างไร?

.....

บันทึกท้ายแผนการจัดการเรียนรู้ที่ 1

เทคนิคการสอน: พยายามหลีกเลี่ยงศัพท์เทคนิคทางคอมพิวเตอร์ในช่วงแรก ให้ใช้คำว่า "วางแผน", "จับประเด็น", "รูปแบบ" แทน เพื่อลดกำแพงความกลัว

จุดเน้น: ย้ำให้นักเรียนเห็นว่า พวกเขาเก่งเรื่องการ "สื่อสาร" และ "ลำดับเรื่องราว" อยู่แล้ว ซึ่งนั่นคือทักษะที่สำคัญที่สุดในการเขียนโปรแกรม

เฉลย/แนวคำตอบสำหรับครู (Teacher's Guide)

ใช้สำหรับยกตัวอย่างให้นักเรียนดู เพื่อให้เห็นภาพชัดเจนขึ้น

ตัวอย่าง Mission B: เตรียมสอบ HSK ระดับ 4

ส่วนที่ 1 (Decomposition):

1. ท่องคำศัพท์ 1,200 คำ
2. ฟังไวยากรณ์ (เรียงประโยค)
3. ฟังการฟัง
4. ทำข้อสอบเก่าจับเวลา

ส่วนที่ 2 (Pattern Recognition):

รูปแบบที่พบ: ข้อสอบพาร์ทการอ่าน มักจะถามหา Main Idea เสมอ, คำศัพท์ที่ออกบ่อยมักเกี่ยวกับชีวิตประจำวันและที่ทำงาน

เทคนิค: จำศัพท์วันละ 20 คำ เป็นเวลา 60 วัน (Pattern การจำ)

ส่วนที่ 3 (Abstraction):

Focus: จำศัพท์ตัวที่ยังไม่ได้, ไวยากรณ์ตัวเชื่อมประโยคที่ออกบ่อย

Ignore: การคัดลายมือสวยๆ (เพราะสอบคอมพิวเตอร์), ศัพท์ยากระดับ HSK 5-6 ที่ยังไม่ออกสอบ

ส่วนที่ 4 (Algorithm):

1. ตื่นนอน 06.00 น. ท่องศัพท์ใหม่ 20 คำ
2. ระหว่างนั่งรถ ฟัง Podcast ภาษาจีน
3. พักเที่ยง ทบทวนศัพท์เก่าผ่านแอปฯ มือถือ
4. ก่อนนอน ทำข้อสอบเก่า 1 ชุด แล้วตรวจคำตอบทันที
5. ถ้าคะแนน < 60% ให้กลับไปทวนบทที่ผิด, ถ้า >= 60% ให้พักผ่อนได้

Reflection: เหมือนการแต่งประโยคตรงๆ เราต้องรู้ว่าส่วนประกอบ (ศัพท์) คืออะไร และต้องมีลำดับ (ไวยากรณ์) ที่ถูกต้อง ถึงจะสื่อสาร (สอบผ่าน) ได้สำเร็จ

แผนการจัดการเรียนรู้ที่ 2

กลุ่มสาระการเรียนรู้วิทยาศาสตร์และเทคโนโลยี (วิทยาการคำนวณ)

หน่วยการเรียนรู้ที่ 1: พื้นฐานแนวคิดเชิงคำนวณ (Coding as a Language)

เรื่อง: ผังงานเล่าเรื่อง (Flowchart Storytelling)

ระดับชั้น: มัธยมศึกษาปีที่ 4 (สายศิลป์-ภาษา/สังคม) เวลา: 2 คาบ (100 นาที)

1. มาตรฐานการเรียนรู้/ตัวชี้วัด

ว 4.2 ม.4/1: ประยุกต์ใช้แนวคิดเชิงคำนวณในการพัฒนาโครงงานฯ (เน้นการออกแบบขั้นตอนวิธีที่ชัดเจน)

2. สาระสำคัญ (Concept)

"ผังงาน คือ สตอรี่บอร์ดของคอมพิวเตอร์" ก่อนจะทำภาพยนต์ ผู้กำกับต้องเขียน Storyboard เพื่อลำดับภาพ ก่อนจะเขียนโปรแกรม Programmer ก็ต้องเขียน Flowchart เพื่อลำดับการทำงาน การเขียนผังงานช่วยให้เราเห็น "ทางเลือก (Decision)" ในเรื่องราวชัดเจนขึ้น ว่าถ้าตัวละครเลือกทางนี้ ผลลัพธ์จะเป็นอย่างไร (If-Else) ซึ่งเป็นพื้นฐานสำคัญที่สุดของการเขียนโปรแกรม

3. จุดประสงค์การเรียนรู้ (K-P-A)

ด้านความรู้ (K) : จำและอธิบายความหมายของสัญลักษณ์ Flowchart พื้นฐาน (Start/End, Process, Decision, Input/Output, Flow Line) ได้

ด้านทักษะกระบวนการ (P) : สามารถเปลี่ยนเรื่องเล่าหรือเหตุการณ์ในชีวิตประจำวัน ให้เป็นผังงานที่มีเงื่อนไข (Decision) ได้ถูกต้อง

ด้านคุณลักษณะ (A) : สนุกกับการออกแบบตรรกะและการเล่าเรื่องผ่านสัญลักษณ์

4. สาระการเรียนรู้

1. สัญลักษณ์ผังงานมาตรฐาน:

วงรี (Terminator) = จุดเริ่ม/จบเรื่อง

สี่เหลี่ยมผืนผ้า (Process) = การกระทำ/เหตุการณ์

สี่เหลี่ยมขนมเปียกปูน (Decision) = จุดตัดสินใจ/ทางแยกของเรื่อง

สี่เหลี่ยมด้านขนาน (Input/Output) = การพูด/รับข้อมูล

2. โครงสร้างแบบมีเงื่อนไข (Selection/Branching): การแตกเส้นเรื่องเมื่อมีทางเลือก (จริง/เท็จ)

5. กิจกรรมการเรียนรู้

ขั้นนำ (Introduction) - 20 นาที

1. Game "ทางแยกหัวใจ": ครูเล่าเรื่องสั้นๆ แล้วให้นักเรียนโหวตเลือกทางเดิน เช่น





- ครู: "เช้าวันจันทร์ ตื่นสายมาก! (07.45 น.) จะทำยังไง?"

- *ทางเลือก A:* รับประทานอาหารแต่งตัว (เสี่ยงไปไม่ทัน)
- *ทางเลือก B:* แก่ล้างป่วยลาหยุด (รอดตัวแต่ขาดเรียน)
- *ครู:* วาดเส้นทางบนกระดานให้ดูเป็นแผนภาพง่ายๆ

2. Link to Lesson: ครูเชื่อมโยงว่า "เห็นไหมว่าชีวิตเรามีทางเลือก (Decision) ตลอดเวลา คอมพิวเตอร์ก็เหมือนกัน มันต้องรู้ว่าถ้าเจอ A ต้องทำอะไร หรือถ้าเจอ B ต้องทำอะไร เรามาเรียนรู้วิธีวาดแผนที่ความคิดนี้กัน เรียกว่า Flowchart"

ขั้นสอน (Instruction) - 30 นาที

1. Mapping สัญลักษณ์: ครูสอนสัญลักษณ์โดยเปรียบเทียบกับองค์ประกอบละคร

-  Start/End: เหมือน ไตเติลเปิด/ปิดหนังสือ
-  Process (สีเหลี่ยม): เหมือน ฉากแอคชั่น (พระเอกวิ่ง, นางเอกกินข้าว)
-  Decision (ขนมเปียกปูน): เหมือน จุดหักมุม (Plot Twist) หรือ คำถาม (ใช่/ไม่ใช่?)
-  Input/Output (ด้านขนาน): เหมือน บทพูด (Dialogue) หรือ การเห็นสิ่งของ

2. Demo: ครูเขียน Flowchart ตัวอย่างบนกระดาน: "ปฏิบัติการสั่งชานมไข่มุก"

- Start -> สั่งชานม -> (Decision) ใส่ไข่มุกไหม?
- -> ถ้าใช่ (Yes) -> ตักไข่มุก (+5 บาท)
- -> ถ้าไม่ (No) -> ไม่ตัก
- -> จ่ายเงิน -> รับของ -> End

ขั้นปฏิบัติ (Practice) - 40 นาที

กิจกรรม "The Destiny Designer" (ลิขิตฟ้าชะตาเขียนเอง)

1. แจกใบงานที่ 1.2
2. ให้นักเรียน (เดี่ยวหรือคู่) เลือกสถานการณ์ เพื่อเขียนผังงานแบบมีทางเลือก (Branching Logic)

Theme สายภาษา: การแต่งตัวไปเดท / การเตรียมตัวไปคอนเสิร์ต / การเลือกแผนการเรียน / การเอาตัวรอดในนิยายสยองขวัญ

ขั้นสรุป (Conclusion) - 10 นาที

1. Showcase: ขออาสาสมัคร 1-2 คน มาโชว์ Flowchart ของตนเอง (ครูฉายขึ้นจอหรือวาดตาม)
2. Wrap up: "วันนี้เราเขียนโค้ดโดยไม่รู้ตัวอีกแล้ว การที่พวกเราเขียนลูกศรแยก Yes/No นั่นคือคำสั่ง If-Else ในภาษาคอมพิวเตอร์ คราวหน้าเราจะลองเอาตรรกะนี้ไปใส่ในโปรแกรมจริงๆ กัน"

6. สื่อ/แหล่งการเรียนรู้

- สไลด์ประกอบการสอน: เรื่อง "สัญลักษณ์ผังงานฉบับนักเล่าเรื่อง" (เปรียบเทียบสัญลักษณ์ Flowchart กับองค์ประกอบของภาพยนตร์/นิยาย)
- ใบงานที่ 2: "The Destiny Designer" (ลิขิตฟ้าชะตาเขียนเอง)
- ตัวอย่างผังงาน (Model): ผังงานสถานการณ์ในชีวิตประจำวันขนาดใหญ่สำหรับติดหน้ากระดาน หรือฉายขึ้นจอ (เช่น เรื่อง "เย็นนี้กินอะไรดี?")

7. การวัดและประเมินผล

จุดประสงค์การเรียนรู้	วิธีการวัดผล	เครื่องมือวัดผล	เกณฑ์การประเมิน
1. ด้านความรู้ (K) จำและอธิบายความหมายของสัญลักษณ์ Flowchart ได้	ตรวจใบงานที่ 2 (ตอนที่ 1: จับคู่สัญลักษณ์)	แบบฝึกหัดในใบงาน	ตอบถูกต้องร้อยละ 70 ขึ้นไป (3 ใน 4 ข้อ) ถือว่าผ่าน
2. ด้านทักษะกระบวนการ (P) เขียนผังงานแบบมีเงื่อนไข (Decision) ได้ถูกต้อง	ตรวจผลงานผังงาน "My Story Path" ในใบงานที่ 2	แบบประเมินชิ้นงาน (Rubric)	ระดับคุณภาพพอใช้ขึ้นไป (เกณฑ์: มีจุดเริ่มต้น/จบ, ใช้สัญลักษณ์ Decision ถูกต้อง, ทิศทางถูกต้อง)
3. ด้านคุณลักษณะ (A) สนุกกับการออกแบบและมีความคิดสร้างสรรค์	สังเกตพฤติกรรมขณะทำกิจกรรมและการนำเสนอผลงาน	แบบสังเกตพฤติกรรม	นักเรียนให้ความร่วมมือและมีความกระตือรือร้นในระดับผ่าน

เกณฑ์การให้คะแนนชิ้นงาน (Rubric) แบบย่อ:

ดีมาก (3):

- ผังงานมีความซับซ้อน เลือกใช้สัญลักษณ์ถูกต้องแม่นยำ
- เนื้อเรื่องมีความคิดสร้างสรรค์และมีทางเลือกที่น่าสนใจ

พอใช้ (2):

- ผังงานทำงานได้ตามตรรกะพื้นฐาน
- มีการใช้เงื่อนไข (Decision) อย่างน้อย 1 จุด แต่อาจมีข้อผิดพลาดเล็กน้อยเรื่องทิศทางลูกศร

ปรับปรุง (1):

- ไม่มีการใช้เงื่อนไขทางเลือก (เป็นเส้นตรงอย่างเดียว)
- เลือกใช้สัญลักษณ์ผิดความหมายจนสื่อสารไม่ได้

ใบงานที่ 2: The Destiny Designer

รายวิชา: วิทยาการคำนวณ ม.4 | ชื่อ-สกุล: ชั้น ม. เลขที่

ตอนที่ 1: ถอดรหัสสัญลักษณ์ (Symbol Decoding)

โยงเส้นจับคู่สัญลักษณ์ฝั่งงาน ให้ตรงกับความหมายในการเล่าเรื่อง

(รูปวาด)

(ความหมาย)

วงรี (Start/End)

A. จุดตัดสินใจ (ทางแยกของเรื่อง / คำถาม ใช่หรือไม่)

สี่เหลี่ยมผืนผ้า (Process)

B. บทสนทนา / การรับรู้ข้อมูล / แสดงผล

ขนมเปียกปูน (Decision)

C. จุดเริ่มต้น หรือ จุดจบบริบูรณ์

สี่เหลี่ยมด้านขนาน (I/O)

D. การกระทำ / เหตุการณ์ที่เกิดขึ้น

ตอนที่ 2: สร้างเส้นทางลิขิต (My Story Path)

คำชี้แจง: ให้นักเรียนออกแบบ Flowchart จากสถานการณ์ที่กำหนดให้ โดยต้องมีการตัดสินใจ (Decision)

อย่างน้อย 1 จุด

สถานการณ์ตัวอย่าง: "วันนี้ฉันจะไปดูหนังกับเพื่อน"

1. ไปถึงหน้าโรงหนัง (Start)

2. เช็กรอบหนัง

3. (จุดตัดสินใจ) มีรอบที่ตรงกันไหม?

- ถ้ามี (Yes): ซื้อตั๋ว -> ซื้อป๊อปคอร์น -> เข้าดูหนัง

- ถ้าไม่มี (No): เปลี่ยนไปเดินซื้อของแทน -> กินไอศกรีม

4. กลับบ้าน (End)

แผนการจัดการเรียนรู้รายวิชาวิทยาการคำนวณ ชั้นมัธยมศึกษาปีที่ 4 (สายศิลป์-ภาษา)

โดย นายสิทธิชัย ทิพย์สิงห์ ตำแหน่ง ครู วิทยฐานะ ครูชำนาญการพิเศษ โรงเรียนสตรีศึกษา สำนักงานเขตพื้นที่การศึกษามัธยมศึกษาร้อยเอ็ด

พื้นที่วาด Flowchart ของฉัน: (เลือกแต่งเรื่องเองได้เลย เช่น การสอบสัมภาษณ์, การไปเที่ยว,
หรือสู่มหาชาปอง)

ชื่อเรื่อง:

Challenge Zone (สำหรับคนเก่ง)

ลองเพิ่มเงื่อนไขซ้อนกันดูสิ! (Nested Decision) เช่น ถ้ามีรอบหนึ่ง -> (เช็คต่อ) ว่าที่นั่งเต็มไหม? -> ถ้าไม่เต็ม
ค่อยซื้อตั๋ว

บันทึกท้ายแผนการจัดการเรียนรู้ที่ 2

ลูกศร: เด็กสายศิลป์อาจจะลากลูกศรสวยงามแต่หัวลูกศรมีทิศทาง (Flow ต้องไหลจากบนลงล่าง หรือ ซ้ายไปขวา)

Decision: ในสี่เหลี่ยมขนมเปียกปูน ต้องเป็น "คำถาม" ที่ตอบได้แค่ Yes/No หรือ True/False เท่านั้น (ห้ามเขียนว่า "กินอะไร" แต่ต้องเขียนว่า "กินเผ็ดได้ไหม")

ข้อสังเกตจากตอนที่ 1

การจับคู่ตอนที่ 1: 1 -> C (เริ่ม/จบ) 2 -> D (การกระทำ) 3 -> A (ตัดสินใจ) 4 -> B (รับ/แสดงข้อมูล)

ตัวอย่าง Flowchart (สำหรับวาดให้เด็กดูเป็นไกด์): เรื่อง "เย็นนี้กินอะไรดี?"

Plaintext

(Start)

|

[เปิดดูเงินในกระเป๋า]

|

< มีเงิน > 100 บาท? > ----(No)----> [กินก๋วยเตี๋ยวหน้าปากซอย]

|

(Yes)

|

[กินบุฟเฟต์ชาบู] <-----+>

|

[จ่ายเงิน]

|

(End)

แผนการจัดการเรียนรู้ที่ 3

กลุ่มสาระการเรียนรู้วิทยาศาสตร์และเทคโนโลยี (วิทยาการคำนวณ)

หน่วยการเรียนรู้ที่ 2: ตรรกะและการแก้ปัญหาด้วย Flowgorithm

เรื่อง: เริ่มต้นสร้างโปรแกรมพูดได้ (Variables & Input/Output)

ระดับชั้น: มัธยมศึกษาปีที่ 4 (สายศิลป์-ภาษา/สังคม) เวลา: 2 คาบ (100 นาที)

1. มาตรฐานการเรียนรู้/ตัวชี้วัด

ว 4.2 ม.4/1: ประยุกต์ใช้แนวคิดเชิงคำนวณในการพัฒนาโครงงานที่มีการบูรณาการกับวิชาอื่นอย่างสร้างสรรค์ และเชื่อมโยงกับชีวิตจริง

(คำอธิบาย: แม้ตัวชี้วัดหลักจะเน้นโครงงาน แต่ในแผนนี้เน้น "การประยุกต์ใช้เครื่องมือ (Flowgorithm)" เพื่อเตรียมความพร้อมสู่การทำโครงงาน)

2. สาระสำคัญ (Concept)

"ตัวแปร (Variable) คือ กล่องเก็บความทรงจำ" คอมพิวเตอร์มีความจำสั้น ถ้าเราอยากให้มันจำชื่อเรา หรือ คำตอบของเรา เราต้องสร้าง "กล่อง" ขึ้นมาใส่ข้อมูลนั้นไว้ กล่องนี้เรียกว่า "ตัวแปร (Variable)" ในแผนนี้เราจะฝึกใช้โปรแกรม Flowgorithm สร้างกล่องเพื่อเก็บ "ข้อความ (String)" และนำมาแสดงผลเป็นประโยคโต้ตอบง่ายๆ

3. จุดประสงค์การเรียนรู้ (K-P-A)

ด้านความรู้ (K) : อธิบายหน้าที่ของคำสั่ง Declare (ประกาศตัวแปร), Input (รับค่า), Output (แสดงผล) ได้

ด้านทักษะกระบวนการ (P) : สามารถเขียนผังงานใน Flowgorithm เพื่อรับชื่อและนามสกุล แล้วแสดงผลข้อความที่ทักทายได้

ด้านคุณลักษณะ (A) : ตื่นเต้นและกระตือรือร้นในการเห็นผลลัพธ์ที่ตนเองสร้างขึ้นทำงานได้จริง

4. สาระการเรียนรู้

1. แนะนำโปรแกรม Flowgorithm: ส่วนประกอบหน้าจอ, การเปลี่ยนภาษาเมนู (ถ้าจำเป็น)

2. ชนิดข้อมูล (Data Type): เน้น String (ข้อความ) เป็นหลักในคาบนี้

3. คำสั่งพื้นฐาน:

- Declare: การสร้างตัวแปร (เปรียบเทียบเหมือนการเตรียมกล่อง)
- Input: การรับข้อมูลจากคีย์บอร์ดใส่ลงกล่อง
- Output: การนำข้อมูลในกล่องมาแสดงบนหน้าจอ
- Concatenation (&): การนำข้อความมาต่อกัน (เช่น "สวัสดี" & ชื่อ)

5. กิจกรรมการเรียนรู้

ขั้นนำ (Introduction) - 15 นาที

Demonstrate: ครูเปิดโปรแกรม Flowgorithm ขึ้นจ่อ แล้วรันโปรแกรมง่ายๆ ให้อู (เช่น โปรแกรมถามชื่อ แล้วตอบว่า "ยินดีที่ได้รู้จักครับ น้อง...")

Hook: "จำฝังงานที่เราวาดเมื่อคาบที่แล้วได้ไหม? วันนี้เราจะเสกให้มันมีชีวิต ขยับและคุยกับเราได้จริง!"

ขั้นสอน (Instruction) - 35 นาที

1. รู้จักเครื่องมือ: ครูแนะนำเมนูหลัก และวิธีกดสร้างสัญลักษณ์ (คลิกที่เส้นลูกศร)
2. Concept "กล่องตัวแปร":
 - ครูวาดรูปกล่องบนกระดาน เขียนป้ายชื่อว่า Name
 - อธิบาย: "ก่อนจะจำชื่อใคร เราต้องเตรียมสมองส่วนความจำไว้ก่อน ในโปรแกรมเรียกว่าคำสั่ง

Declare (ประกาศตัวแปร)"

- เน้นย้ำ: ชนิดข้อมูลต้องเลือกเป็น String (เพราะชื่อคนเป็นตัวอักษร ไม่ใช่ตัวเลข)
3. Live Coding (ทำไปพร้อมกัน):

Step 1: เพิ่ม Declare -> ตั้งชื่อ myName -> Type: String

Step 2: เพิ่ม Output -> พิมพ์คำถาม "คุณชื่ออะไรครับ?" (ต้องมีเครื่องหมายคำพูด "...")

เสมอหรือข้อความ)

Step 3: เพิ่ม Input -> เลือกตัวแปร myName (รับค่ามาใส่กล่องนี้)

Step 4: เพิ่ม Output -> แสดงผล "สวัสดีครับคุณ" & myName

TIPS: ครูต้องเน้นเรื่องเครื่องหมาย & (Ampersand) ว่าทำหน้าที่เหมือนการเชื่อมข้อความ

กับตัวแปร

ขั้นปฏิบัติ (Practice) - 40 นาที

กิจกรรม: "The Greeting Bot" (บอททักทายหลายภาษา)

1. ให้นักเรียนเปิดเครื่องและเข้าโปรแกรม Flowgorithm
2. ทำใบงานที่ 1.3: สร้างโปรแกรมแนะนำตัวเองง่ายๆ
3. Challenge: ใครทำเสร็จแล้ว ให้ลองเพิ่มการถาม "นามสกุล" (Surname) และ "ชื่อเล่น"

(Nickname) แล้วแสดงผลรวมกันเป็นประโยคยาวๆ ดู

ขั้นสรุป (Conclusion) - 10 นาที

1. Run & Show: เดินดูผลงานนักเรียน ให้ลองกด Play รันโปรแกรมให้เพื่อนข้างๆ ดู
2. Error Checking: ถามนักเรียนว่าใครเจอ Error สีแดงบ้าง? (ส่วนใหญ่มักลืมใส่ "..." หรือลืม Declare) -> นี่คือนี้ออกาสสอนเรื่อง Syntax Error ว่าคอมพิวเตอร์เปะมาก ผิดนิดเดียวก็ไม่ยอม

6. สื่อ/แหล่งการเรียนรู้

- โปรแกรม Flowgorithm: ติดตั้งลงในเครื่องคอมพิวเตอร์ของนักเรียน
- สไลด์ประกอบการสอน: เรื่อง "ตัวแปรคือกล่องความทรงจำ"
- ใบงานที่ 3: "The Greeting Bot" (บอททักทายหลายภาษา)
- แหล่งเรียนรู้ออนไลน์: เว็บไซต์ documentation ของ Flowgorithm

7. การวัดและประเมินผล

จุดประสงค์การเรียนรู้	วิธีการวัดผล	เครื่องมือวัดผล	เกณฑ์การประเมิน
1. ด้านความรู้ (K) อธิบายหน้าที่ของคำสั่ง Declare, Input, Output และการใช้งานตัวแปร String ได้	ตรวจคำตอบในใบงานที่ 3 (Part 1: รู้จักหน้าต่าง Flowgorithm)	ใบงานที่ 3	ตอบคำถามถูกต้องร้อยละ 70 ขึ้นไป (3 ใน 4 ข้อ) ถือว่า ผ่าน
2. ด้านทักษะ (P) เขียนผังงานใน Flowgorithm เพื่อรับค่าชื่อและแสดงผลข้อความโต้ตอบได้ถูกต้อง	ตรวจสอบการทำงานของโปรแกรม Flowgorithm ที่นักเรียนสร้าง (Part 2)	แบบประเมินผล งานปฏิบัติ (Checklist)	ระดับคุณภาพพอใช้ขึ้นไป (เกณฑ์: โปรแกรมรันได้ไม่ Error, มีการประกาศตัวแปรถูกต้อง, แสดงผลข้อความได้ตามโจทย์)
3. ด้านคุณลักษณะ (A) ตื่นตัวและกระตือรือร้นในการเห็นผลลัพธ์ที่ตนเองสร้างขึ้น	สังเกตพฤติกรรมขณะทำกิจกรรมและการทดลองรันโปรแกรม	แบบสังเกต พฤติกรรม	นักเรียนมีความสนใจ จดจ่อกับงาน และพยายามแก้ไขเมื่อโปรแกรม Error ในระดับ ผ่าน

เกณฑ์การให้คะแนนชิ้นงาน (Rubric) - โปรแกรม Greeting Bot:

ดีมาก (3 คะแนน):

- ประกาศตัวแปร (Declare) ได้ถูกต้อง

- เลือก Type เป็น String ถูกต้อง
- ใช้คำสั่ง Input/Output ได้ถูกต้องตามลำดับ
- เชื่อมข้อความกับตัวแปร (Concatenation) ด้วยเครื่องหมาย & ได้ถูกต้อง จัดวรรคตอนสวยงาม

พอใช้ (2 คะแนน):

- โปรแกรมทำงานได้
- รับค่าและแสดงผลได้แต่อาจมีการจัดรูปแบบข้อความไม่สวยงาม (เช่น ลืมเว้นวรรคระหว่างชื่อกับ

คำว่าสวัสดี)

- ตั้งชื่อตัวแปรไม่สื่อความหมาย

ปรับปรุง (1 คะแนน):

- โปรแกรม Error รันไม่ได้ (เช่น ลืม Declare ตัวแปร, ลืมใส่เครื่องหมายคำพูดในข้อความ)
- เลือกใช้สัญลักษณ์ผิด (เช่น เอาสี่เหลี่ยม Assign มาใช้แทน Input)

ใบงานที่ 3: The Greeting Bot

รายวิชา: วิทยาการคำนวณ ม.4 | ชื่อ-สกุล:

Part 1: รู้จักหน้าตา Flowgorithm

จงเติมคำลงในช่องว่าง ให้ตรงกับหน้าที่ของสัญลักษณ์ในโปรแกรม

1. Declare (ประกาศตัวแปร): ใช้สำหรับ
2. Input (นำเข้า): ใช้สำหรับ
3. Output (แสดงผล): ใช้สำหรับ
4. String (สตริง): คือชนิดข้อมูลที่เป็น

Part 2: ฝึกเขียนโค้ด (Coding Practice)

ภารกิจ: สร้างโปรแกรมที่คุยกับเราได้ โดยทำตามขั้นตอนต่อไปนี้

1. ประกาศตัวแปร (Declare):

สร้างตัวแปรชื่อ firstName (เลือก Type เป็น String)

สร้างตัวแปรชื่อ favoriteFood (เลือก Type เป็น String)

2. เขียนคำสั่ง (Main Flow):

Output: "สวัสดีครับ ขอทราบชื่อหน่อยครับ"

Input: firstName

Output: "ยินดีที่ได้รู้จักครับ คุณ" & firstName

Output: "แล้ววันนี้คุณอยากกินเมนูอะไรครับ?"

Input: favoriteFood

Output: "โอ้โห! " & favoriteFood & " น่าอร่อยมาก เต็มใจพาไปกินนะ"

ทดลองรัน (Run): กดปุ่ม Play สีเขียว แล้วลองคุยกับโปรแกรมดูสิ!

Part 3: บันทึกผลลัพธ์ (Result Log)

เมื่อรันโปรแกรมผ่านแล้ว ให้เขียนสิ่งที่หน้าจอแสดงผลออกมา (สมมติข้อมูลเองได้เลย)

หน้าจอ Console:

สวัสดีครับ ขอทราบชื่อหน่อยครับ (ฉันพิมพ์ว่า): ยินดีที่ได้รู้จักครับ คุณ
..... แล้ววันนี้คุณอยากกินเมนูอะไรครับ? (ฉันพิมพ์ว่า):

คำถามชวนคิด:

ถ้าเราใส่ใส่เครื่องหมาย & ระหว่าง "ข้อความ" กับ ตัวแปร (เช่น พิมพ์ว่า "สวัสดี" firstName) โปรแกรมจะฟ้องว่า Error หรือไม่?

บันทึกท้ายแผนการจัดการเรียนรู้ที่ 3

ปัญหาที่พบบ่อย (Pain Points):

- เครื่องหมายคำพูด (" "): นักเรียนมักลืมใส่ Double Quote ครอบข้อความภาษาไทยในคำสั่ง Output ทำให้โปรแกรม Error
- การเชื่อมคำ (&): ใน Flowgorithm ต้องใช้ & เชื่อม (บางภาษาใช้ + หรือ ,) ครูต้องย้ำจุดนี้
- ชื่อตัวแปร: แนะนำให้นักเรียนตั้งชื่อตัวแปรเป็นภาษาอังกฤษ (เช่น name, food) แม้ Flowgorithm จะรองรับภาษาไทยในบางเวอร์ชัน แต่เพื่อปูพื้นฐาน Python ควรฝึกภาษาอังกฤษครับ

แผนการจัดการเรียนรู้ที่ 4

กลุ่มสาระการเรียนรู้วิทยาศาสตร์และเทคโนโลยี (วิทยาการคำนวณ)

หน่วยการเรียนรู้ที่ 2: ตรรกะและการแก้ปัญหาด้วย Flowgorithm

เรื่อง: นักคำนวณคู่ใจ (Basic Calculations & Data Types)

ระดับชั้น: มัธยมศึกษาปีที่ 4 (สายศิลป์-ภาษา/สังคม) เวลา: 2 คาบ (100 นาที)

1. มาตรฐานการเรียนรู้/ตัวชี้วัด

ว 4.2 ม.4/1: ประยุกต์ใช้แนวคิดเชิงคำนวณในการพัฒนาโครงงานที่มีการบูรณาการกับวิชาอื่นอย่างสร้างสรรค์ และเชื่อมโยงกับชีวิตจริง

2. สาระสำคัญ (Concept)

"คอมพิวเตอร์ คือ เครื่องคิดเลขที่ทำงานตามสั่ง" ในการเขียนโปรแกรม เราไม่ได้เรียนคณิตศาสตร์เพื่อไปสอบแข่งขัน แต่เราเรียนเพื่อสั่งให้คอมพิวเตอร์คิดเลขแทนเรา หัวใจสำคัญคือการเลือกชนิดข้อมูล (Data Types) ให้ถูก ว่าจะเป็นจำนวนเต็ม (Integer) หรือ ทศนิยม (Real) และเลือกใช้เครื่องหมาย (Operators) ให้ถูกต้องตามสิ่งที่เราต้องการคำนวณ

3. จุดประสงค์การเรียนรู้ (K-P-A)

ด้านความรู้ (K) : อธิบายความแตกต่างของชนิดข้อมูลตัวเลข (Integer vs Real) และตัวดำเนินการ (+, -, *, /) ได้

ด้านทักษะ (P) : ออกแบบผังงานคำนวณภาษีมูลค่าเพิ่ม (VAT) หรือ แลกสกุลเงิน (Currency Exchange) ได้

ด้านคุณลักษณะ(A) : ตระหนักถึงความละเอียดรอบคอบในการกำหนดชนิดข้อมูล (ประกาศผิด ผลลัพธ์ผิด)

4. สาระการเรียนรู้

1. ชนิดข้อมูลตัวเลข (Numeric Data Types):

- Integer: จำนวนเต็ม (เช่น จำนวนคน, ปี พ.ศ.)
- Real: จำนวนจริง/ทศนิยม (เช่น เงิน, ส่วนสูง, เกรดเฉลี่ย)

2. การประมวลผล (Process): การเขียนสมการในสัญลักษณ์ Assign (สี่เหลี่ยมผืนผ้า)

- เครื่องหมาย: + (บวก), - (ลบ), * (คูณ), / (หาร), % (หารเอาเศษ), ^ (ยกกำลัง)

5. กิจกรรมการเรียนรู้

ขั้นนำ (Introduction) - 15 นาที

1. Scenario: ครูยกสถานการณ์ "สมมติเราจะไปเที่ยวเกาหลี/ญี่ปุ่น เรามีเงินไทย 5,000 บาท จะแลกเงินวอน/เยน ได้เท่าไร?"
2. Ask: ให้นักเรียนลองจิ้มเครื่องคิดเลขมือถือ
3. Problem: "ถ้าเราต้องแลกเงินให้คนทั้งห้อง 40 คน เราต้องมากดเครื่องคิดเลข 40 รอบเหนื่อยไหม? จะดีกว่าไหมถ้าเราเขียนโปรแกรมขึ้นมา 1 ตัว แล้วแค่รอกตัวเลขบาทลงไป มันคำนวณออกมาให้ทันที?"

ขั้นสอน (Instruction) - 35 นาที

1. Data Types Matter: ครูสอนเรื่องกล่องตัวแปร (ต่อจากคาบที่แล้ว)
 - Integer: เหมือนกล่องใส่ลูกบอล (นับเป็นลูกๆ หั่นครึ่งไม่ได้)
 - Real: เหมือนกล่องใส่น้ำ (ตวงเป็น 1.5 ลิตร, 0.75 ลิตรได้) -> เน้น: เรื่องเงินต้องใช้ Real เสมอ!
2. Symbol "Assign" (กำหนดค่า): แนะนำสัญลักษณ์สี่เหลี่ยมผืนผ้า ใช้สำหรับคำนวณ
 - สูตร: ตัวแปรเก็บผลลัพธ์ = ตัวแปรตั้งต้น * ตัวเลข
3. Demo (โปรแกรมแปลงเงิน):
 - Declare: thaiBaht (Real), rate (Real), yen (Real)
 - Input: thaiBaht
 - Assign: yen = thaiBaht * 0.42 (สมมติเรต)
 - Output: yen

ขั้นปฏิบัติ (Practice) - 40 นาที

กิจกรรม: "Smart Traveler" (นักเดินทางอัจฉริยะ)

1. ให้นักเรียนจับคู่ ทำใบงานที่ 1.4
2. โจทย์คือสร้างโปรแกรมช่วยคำนวณสำหรับการท่องเที่ยว (เลือก 1 โจทย์):
 - โจทย์ A (Currency): รับค่าเงินบาท -> แปลงเป็นเงินต่างประเทศ (ที่อยากไป)
 - โจทย์ B (Shopping): รับราคาสินค้า -> คำนวณภาษี VAT 7% -> แสดงราคารวม
 - โจทย์ C (Sharing): รับค่าอาหารทั้งหมด -> รับจำนวนเพื่อนที่ไปกิน -> คำนวณว่าต้องจ่ายคนละกี่บาท (หารยาว)

ขั้นสรุป (Conclusion) - 10 นาที

1. Debrief: สุ่มถามนักเรียนที่เลือกโจทย์ C (หารค่าข้าว)

- "ถ้าค่าข้าว 500 บาท ไปกัน 3 คน หารแล้วได้เท่าไร?" (166.666...)

- "ถ้าเราประกาศตัวแปรเป็น Integer ผลลัพธ์จะเป็นยังไง?" (คอมพิวเตอร์อาจจะตัดทศนิยมทิ้ง หรือ Error) -> ย้ำความสำคัญของ Real

6. สื่อ/แหล่งการเรียนรู้

- โปรแกรม Flowgorithm: ติดตั้งลงเครื่องคอมพิวเตอร์

- ไฟล์ผังงานแสดงการเขียนโปรแกรมอัตราแลกเปลี่ยน

- ใบงานที่ 4: "Smart Traveler" (นักเดินทางอัจฉริยะ)

- เครื่องคิดเลข/โทรศัพท์มือถือ: สำหรับให้นักเรียนลองกดเปรียบเทียบผลลัพธ์กับโปรแกรม

7. การวัดและประเมินผล

จุดประสงค์การเรียนรู้	วิธีการวัดผล	เครื่องมือวัดผล	เกณฑ์การประเมิน
1. ด้านความรู้ (K) อธิบายความแตกต่างและเลือกใช้ชนิดข้อมูล (Integer/Real) ได้เหมาะสมกับโจทย์	ตรวจคำตอบในใบงานที่ 4 (Part 1: เลือกชนิดข้อมูล)	ใบงานที่ 4 "Smart Traveler"	ตอบถูกต้องร้อยละ 70 ขึ้นไป ถือว่า ผ่าน
2. ด้านทักษะ (P) เขียนผังงานเพื่อคำนวณสูตรทางคณิตศาสตร์พื้นฐาน (+, -, *, /) ได้ถูกต้อง	ตรวจสอบการทำงานของผังงานคำนวณในใบงาน (Part 2 & 3)	แบบประเมินชิ้นงาน	ระดับคุณภาพพอใช้ขึ้นไป* (เกณฑ์: คำนวณถูกต้อง, ใช้ตัวแปรถูกประเภท, โปรแกรมรันได้จริง)*
3. ด้านคุณลักษณะ (A) มีความรอบคอบในการกำหนดชนิดตัวแปร (เช่น เรื่องเงินต้องเป็นทศนิยม)	สังเกตการตั้งค่าตัวแปรในโปรแกรม	แบบสังเกตพฤติกรรม	นักเรียนเลือกใช้ตัวแปร Real กับข้อมูลที่มีทศนิยมได้ถูกต้องในระดับ ผ่าน

เกณฑ์การให้คะแนน (Rubrics) สำหรับหน่วยที่ 2: Flowgorithm

ดีมาก (3 คะแนน):

- เลือกใช้ชนิดข้อมูลถูกต้อง (เช่น เงินเป็น Real, จำนวนคนเป็น Integer)
- เขียนสูตรคำนวณถูกต้องตามหลักคณิตศาสตร์
- โปรแกรมคำนวณผลลัพธ์แม่นยำ
- มีการใส่ข้อความกำกับ Input/Output ที่สื่อความหมายชัดเจน (User-friendly)

พอใช้ (2 คะแนน):

- โปรแกรมทำงานได้
- ค่าผลลัพธ์ได้ถูกต้องเป็นส่วนใหญ่
- อาจเลือกชนิดข้อมูล คลาดเคลื่อนบ้าง (เช่น ใช้ Integer กับเงิน ทำให้ทศนิยมหาย)
- ข้อความกำกับสั้นห้วนหรือไม่มีข้อความกำกับ

ปรับปรุง (1 คะแนน):

- เขียนสูตรคำนวณผิด หรือใช้เครื่องหมายผิด (เช่น ใช้ x แทน *)
- ทำให้ผลลัพธ์ผิดพลาด, โปรแกรม Run ไม่ผ่าน หรือ Error

ใบงานที่ 4: Smart Traveler

รายวิชา: วิทยาการคำนวณ ม.4 | ทีมผู้สร้าง: ชั้น ม.

Part 1: เลือกชนิดข้อมูลให้ป้ (Type Matching)

ถ้าต้องสร้างตัวแปรเพื่อเก็บข้อมูลต่อไปนี้ ควรเลือก Type เป็น Integer (จำนวนเต็ม) หรือ Real (ทศนิยม) ?

1. ราคาตัวเครื่องบิน (เช่น 4500.50 บาท) :
2. จำนวนกระเป๋าเดินทาง (เช่น 2 ใบ) :
3. อุณหภูมิอากาศ (เช่น 25.5 องศา) :
4. จำนวนเพื่อนร่วมทริป (เช่น 5 คน) :

Part 2: ออกแบบโปรแกรม (Coding Time)

ภารกิจ: สร้างเครื่องคำนวณสำหรับการเดินทาง

1. เป้าหมายของโปรแกรมคือ: (เช่น แปลงเงินบาทเป็นเยน, หาค่าข้าว)

2. ตัวแปรที่ต้องใช้ (Variables):

- ตัวแปรรับค่าเข้า (Input): ชื่อ ชนิด (Type)

- ตัวแปรเก็บผลลัพธ์ (Result): ชื่อ ชนิด (Type)

3. สูตรการคำนวณ (Formula):

(ตัวแปรผลลัพธ์) = (ตัวแปรตั้งต้น) [+ - * /]

4. ลำดับการทำงาน (Algorithm): (วาด Flowchart คร่าวๆ หรือเขียนลำดับ)

1. ประกาศตัวแปร (Declare)

2. ถามผู้ใช้งาน (Output question)

3. รับค่า (Input)

4. คำนวณ (Assign):

5. แสดงผลลัพธ์ (Output):

Part 3: ทดสอบจริง (Testing)

รันโปรแกรมแล้วจดบันทึกผลลัพธ์

- ทดสอบครั้งที่ 1: ป้อนค่า ผลลัพธ์ที่ได้ (ถูกต้อง/ไม่ถูก)

- ทดสอบครั้งที่ 2: ป้อนค่า ผลลัพธ์ที่ได้ (ถูกต้อง/ไม่ถูก)

(คำถามพิเศษ): ลองป้อนข้อมูลแปลกๆ ตูสิ เช่น ใส่จำนวนเพื่อนเป็น 0 คน ในโปรแกรมหาค่าข้าว จะเกิดอะไรขึ้น?

.....

บันทึกท้ายแผนการจัดการเรียนรู้ที่ 4

1. **กับดักคณิตศาสตร์ (Math Trap):** เด็กสายศิลป์อาจจะกลัวคำว่า "สมการ" ให้ใช้คำว่า "สูตรคิดเงิน" แทน
2. **Case Study "หารด้วยศูนย์" (Division by Zero):** ใน Part 3 คำถามพิเศษ ถ้าเด็กใส่ตัวหารเป็น 0 โปรแกรมจะ Crash (Error)

- **โอกาสทอง:** ครูสามารถสอนแทรกได้ว่า "นี่คือเหตุผลที่โปรแกรมเมอร์ต้องรอบคอบ เพราะทางคณิตศาสตร์หารด้วย 0 ไม่ได้ คอมพิวเตอร์เลยเออเร่อ" (ซึ่งจะนำไปสู่การป้องกันด้วย If-Else ในบทต่อไป)

3. **การตั้งชื่อตัวแปร:** พยายาม让孩子ตั้งชื่อที่สื่อความหมาย เช่น price, tax, total แทนที่จะเป็น a, b, c เพื่อสร้างนิสัยที่ดีในการเขียนโค้ด (Clean Code)

แผนการจัดการเรียนรู้ที่ 5

กลุ่มสาระการเรียนรู้วิทยาศาสตร์และเทคโนโลยี (วิทยาการคำนวณ)

หน่วยการเรียนรู้ที่ 2: ตรรกะและการแก้ปัญหาด้วย Flowgorithm

เรื่อง: เกมทายใจและทางเลือก (Selection & Boolean Logic)

ระดับชั้น: มัธยมศึกษาปีที่ 4 (สายศิลป์-ภาษา/สังคม) เวลา: 2 คาบ (100 นาที)

1. มาตรฐานการเรียนรู้/ตัวชี้วัด

ว 4.2 ม.4/1: ประยุกต์ใช้แนวคิดเชิงคำนวณในการพัฒนาโครงงานที่มีการบูรณาการกับวิชาอื่นอย่างสร้างสรรค์ และเชื่อมโยงกับชีวิตจริง

2. สาระสำคัญ (Concept)

"ชีวิตคือการเลือก (Life is a choice)" ในชีวิตจริง เราตัดสินใจโดยใช้ "เงื่อนไข" ตลอดเวลา เช่น "ถ้าฝนตก ฉันจะกางร่ม" (ถ้าไม่ตก ก็ไม่กาง) ในคอมพิวเตอร์ก็เช่นกัน เราใช้คำสั่ง If (ถ้า) และ Else (มิฉะนั้น) เพื่อกำหนดทางเลือก โดยอาศัยตรรกะแบบ Boolean ที่มีแค่ 2 ค่า คือ จริง (True) หรือ เท็จ (False) เท่านั้น

3. จุดประสงค์การเรียนรู้ (K-P-A)

ด้านความรู้ (K): อธิบายหลักการการทำงานของโครงสร้างแบบทางเลือก (Selection) และเปรียบเทียบค่าความจริง (True/False) ได้

ด้านทักษะ (P): สร้างโปรแกรม Flowgorithm ที่มีการตรวจสอบเงื่อนไข (If-Then-Else) เพื่อตอบโต้กับผู้ใช้งานได้

ด้านเจตคติ (A): สนุกกับการสร้างเงื่อนไขที่หลากหลายและมีความคิดสร้างสรรค์ในการออกแบบผลลัพธ์

4. สาระการเรียนรู้

1. ตรรกะแบบบูลีน (Boolean Logic): จริง (True) / เท็จ (False)

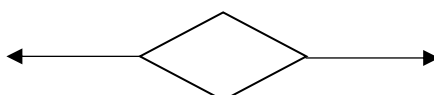
2. ตัวดำเนินการเปรียบเทียบ (Relational Operators):

== (เท่ากับ), != (ไม่เท่ากับ)

> (มากกว่า), < (น้อยกว่า),

<= (น้อยกว่าหรือเท่ากับ), >= (มากกว่าหรือเท่ากับ)

3. สัญลักษณ์ Decision (สี่เหลี่ยมขนมเปียกปูน): ทางแยกซ้าย/ขวา (True/False)



5. กิจกรรมการเรียนรู้

ขั้นนำ (Introduction) - 15 นาที

1. เกม "ยีน-นั่ง" (True/False Game):

ครูพูดประโยคเงื่อนไข ถ้านักเรียนคนไหนตรงเงื่อนไขให้ ยืน ถ้าไม่ตรงให้ นั่ง

โจทย์: "ใครใส่แว่นตา?", "ใครเกิดเดือนมกราคม?", "ใครชอบกินทุเรียน?"

2. สรุปรูป: "คอมพิวเตอร์ก็คิดแบบนี้ มันจะเชื่อกว่า จริง หรือ ไม่จริง ถ้าจริงทำอย่างหนึ่ง ถ้าไม่จริงทำอีกอย่างหนึ่ง"

ขั้นสอน (Instruction) - 35 นาที

1. แนะนำสัญลักษณ์ Decision: สี่เหลี่ยมขนมเปียกปูน คือ "ด้านตรวจ"

2. Syntax การเปรียบเทียบ:

ครูเน้นเรื่องเครื่องหมาย == (เท่ากับสองตัว) ว่าใช้สำหรับ "เปรียบเทียบ" (ต่างจาก = ตัวเดียวที่ใช้ฝากค่า)

3. Demo (โปรแกรมเข้าฝั่บ):

- Declare: age (Integer)

- Input: age

- Decision: age >= 20

True (ฝั่งจริง): Output "เชิญเข้าได้ครับ"

False (ฝั่งเท็จ): Output "กลับบ้านไปกินมมนอนนะน้อง"

4. Demo (โปรแกรมทายรหัสลับ - String):

Declare: password (String)

Input: password

Decision: password == "Love1234" (ต้องมีเครื่องหมายคำพูดเสมอเมื่อเทียบ String)

- True: "ยินดีต้อนรับ!"

- False: "รหัสผิด! ใครนะ?"

ขั้นปฏิบัติ (Practice) - 40 นาที

กิจกรรม: "The Personality Quiz" (แบบทดสอบทายนิสัย)

1. ให้นักเรียนทำใบงานที่ 1.5

2. โจทย์: สร้างโปรแกรมทายนิสัย หรือ ทายใจแบบง่ายๆ (2 ทางเลือก)

ตัวอย่าง: "คุณชอบสุนัขหรือแมว?"

ถ้าตอบ "สุนัข" -> Output: "คุณเป็นคนร่าเริง รักเพื่อนฝูง"

ถ้าตอบ "แมว" -> Output: "คุณเป็นคนโลกส่วนตัวสูง มีเสน่ห์"

ขั้นสรุป (Conclusion) - 10 นาที

1. Testing: สลับกันเล่นโปรแกรมของเพื่อน

2. Discussion: "จะเกิดอะไรขึ้นถ้าเพื่อนพิมพ์คำว่า 'นก' แทนที่จะพิมพ์ 'สุนัข' หรือ 'แมว'?"

(โปรแกรมจะวิ่งไปทาง False เพราะไม่ตรงเงื่อนไขแรก) -> พื้นฐานสู่ Nested If (เงื่อนไขซ้อน)

ในอนาคต

6. สื่อ/แหล่งการเรียนรู้

- ไฟล์การแลกเปลี่ยนสกุลเงินแบบมีตัวเลือกหลายตัว

- ใบงานที่ 1.5: "The Personality Quiz" (แบบทดสอบทายนิสัย)

- โปรแกรม Flowgorithm

7. การวัดและประเมินผล

จุดประสงค์การเรียนรู้	วิธีการวัดผล	เครื่องมือวัดผล	เกณฑ์การประเมิน
1. ด้านความรู้ (K) เปรียบเทียบค่าความจริง (True/False) จากเงื่อนไขที่กำหนดได้	ตรวจคำตอบในใบงานที่ 5 (Part 1: เช็กความเข้าใจ)	ใบงานที่ 5 "The Personality Quiz"	ตอบถูกต้องร้อยละ 70 ขึ้นไป ถือว่าผ่าน
2. ด้านทักษะ (P) สร้างโปรแกรมทายใจที่มีเงื่อนไขทางเลือก (Decision) อย่างน้อย 2 ทางได้	ตรวจผลงานโปรแกรมทายใจบน Flowgorithm	แบบประเมินชิ้นงาน	ระดับคุณภาพพอใช้ขึ้นไป (เกณฑ์: มีเงื่อนไขชัดเจน, โปรแกรมแยกผลลัพธ์ True/False ได้ถูกต้อง)
3. ด้านคุณลักษณะ (A) มีความคิดสร้างสรรค์ในการออกแบบคำถามและผลลัพธ์	สังเกตความน่าสนใจของโจทย์ที่นักเรียนสร้าง	แบบสังเกตพฤติกรรม	นักเรียนสร้างโจทย์ที่สนุกสนานสร้างสรรค์ หรือแปลกใหม่ ในระดับผ่าน

เกณฑ์การให้คะแนนชิ้นงาน (Rubric) - เกมทายใจ (โปรแกรม Personality Quiz)

ดีมาก (3 คะแนน):

- มีการใช้สัญลักษณ์ Decision และกำหนดเงื่อนไขทางเลือก (True/False) ได้ถูกต้อง
- ผลลัพธ์ทั้งสองฝั่งมีความแตกต่างกันชัดเจนและสอดคล้องกับคำตอบ
- โจทย์มีความคิดสร้างสรรค์ สนุกสนาน หรือมีการตกแต่งข้อความน่าสนใจ

พอใช้ (2 คะแนน):

- มีการใช้สัญลักษณ์ Decision และแยกทางเลือกได้
- แต่ผลลัพธ์อาจจะยังไม่สอดคล้องกับเงื่อนไข หรือโจทย์มีความเรียบง่ายเกินไป

ปรับปรุง (1 คะแนน):

- ไม่มีการใช้ Decision (เป็นลำดับเส้นตรง) หรือต่อสายลูกศรผิดทิศทาง
- โปรแกรมไม่สามารถแยกแยะคำตอบของผู้ใช้ได้

ใบงานที่ 5: The Personality Quiz

รายวิชา: วิทยาการคำนวณ ม.4 | ชื่อสกุล: เลขที่.....

Part 1: เช็คความเข้าใจ (Boolean Check)

พิจารณาเงื่อนไขต่อไปนี้ แล้วตอบว่าผลลัพธ์คือ True (จริง) หรือ False (เท็จ)

1. $50 > 100$:
2. "แดง" == "แดง" :
3. อายุ 15 ปี \geq 18 ปี :
4. "A" != "B" (ไม่เท่ากับ) :

Part 2: ออกแบบควิซทายใจ (Quiz Design)

ภารกิจ: สร้างบอทแม่หม่อมทำนายดวง หรือทายนิสัย จากคำถามง่ายๆ 1 ข้อ

1. คำถามของคุณคือ: (เช่น ชอบสีอะไร? / เกรดเฉลี่ยเท่าไร? / ชอบทะเลหรือภูเขา?)
.....
2. ตัวแปรรับคำตอบ: ชื่อ ชนิด (Type) (Integer หรือ String)
3. เงื่อนไขการตัดสินใจ (Condition):
ถ้าตัวแปร [== , > , <]
4. ผลลัพธ์ (Outcome):
 - ทาง True (ใช่): จะบอกว่า
 - ทาง False (ไม่ใช่): จะบอกว่า

Part 3: ลงมือเขียนโค้ด (Flowgorithm)

นำข้อมูลจาก Part 2 มาวาดลงโปรแกรม

Declare: ประกาศตัวแปร

Output: ถามคำถาม

Input: รับคำตอบ

Decision (ขนมเปียกปูน): ใส่เงื่อนไข

Output: ใส่ข้อความตอบกลับทั้ง 2 ฝั่ง (True/False)

(วาดรูปผลลัพธ์หน้าจอเมื่อรันโปรแกรมสำเร็จ)

.....
.....

Challenge Zone (สำหรับคนชอบความท้าทาย)

ลองทำให้โปรแกรมฉลาดขึ้น! โดยใช้เงื่อนไขซ้อนเงื่อนไข (Nested If)

ถาม: ชอบสีอะไร? (แดง / ฟ้ำ / อื่นๆ)

ถ้า "แดง" -> ร้อนแรง

ถ้า "ฟ้ำ" -> อ่อนโยน

(False ของทั้งคู่) -> คุณเป็นคนลึกลับ

บันทึกท้ายแผนการจัดการเรียนรู้ที่ 5

1. ระวัง Case Sensitive: การเปรียบเทียบ String ในคอมพิวเตอร์ "Dog" ไม่เท่ากับ "dog" (ตัวใหญ่ตัวเล็กมีผล)
วิธีแก้เบื้องต้น: บอกให้นักเรียนพิมพ์ให้เหมือนกันเป๊ะๆ หรือสอนว่าคอมพิวเตอร์มันเป๊ะมากนะ
2. เครื่องหมายเปรียบเทียบ:
 - = (ตัวเดียว) คือ เอาค่าใส่กล่อง (Assign)
 - == (สองตัว) คือ ถามว่าเท่ากันไหม? (Compare) จุดนี้นักเรียนผัดบอยที่สุด ต้องย้ำ
3. String vs Integer: ถ้านักเรียนทำโจทย์ทายใจด้วยตัวเลข (เช่น เกรด, อายุ) ให้ใช้ `age > 18` (ไม่ต้องมีฟันทนุ) แต่ถ้าทายด้วยคำ (เช่น สุนัข) ต้องมีฟันทนุ "สุนัข" เสมอ

แผนการจัดการเรียนรู้ที่ 6

กลุ่มสาระการเรียนรู้วิทยาศาสตร์และเทคโนโลยี (วิทยาการคำนวณ)

หน่วยการเรียนรู้ที่ 2: ตรรกะและการแก้ปัญหาด้วย Flowgorithm

เรื่อง: พลังแห่งการทำซ้ำ (Loops & Repetition)

ระดับชั้น: มัธยมศึกษาปีที่ 4 (สายศิลป์-ภาษา/สังคม) เวลา: 2 คาบ (100 นาที)

1. มาตรฐานการเรียนรู้/ตัวชี้วัด

ว 4.2 ม.4/1: ประยุกต์ใช้แนวคิดเชิงคำนวณในการพัฒนาโครงงานที่มีการบูรณาการกับวิชาอื่นอย่างสร้างสรรค์ และเชื่อมโยงกับชีวิตจริง

2. สาระสำคัญ (Concept)

"เหนื่อยครั้งเดียว สบายตลอดไป" ถ้าครูสั่งให้คัดลายมือประโยคเดิม 100 จบ เราคงมือหยุกเหน็บ แต่สำหรับคอมพิวเตอร์ นี่เป็นเรื่องกล้วยๆ! การเขียนโปรแกรมแบบ วนซ้ำ (Loop) ช่วยให้เราสั่งคอมพิวเตอร์ทำงานเดิมซ้ำๆ ก็รอบก็ได้ตามต้องการ โดยไม่ต้องเขียนคำสั่งใหม่ทุกบรรทัด ช่วยลดความซ้ำซ้อนและประหยัดเวลาอย่างมากมหาศาล

3. จุดประสงค์การเรียนรู้ (K-P-A)

ด้านความรู้ (K): อธิบายหลักการทำงานของการทำงานซ้ำ (Loop) และความแตกต่างระหว่างการทำงานซ้ำแบบ รู้จำนวนรอบ (For) กับแบบตรวจสอบเงื่อนไข (While) ได้

ด้านทักษะ (P): เขียนผังงานใน Flowgorithm เพื่อสั่งงานซ้ำๆ (เช่น พิมพ์ข้อความ หรือ ตรวจสอบรหัสผ่านซ้ำๆ) ได้

ด้านคุณลักษณะ (A) : เห็นประโยชน์ของการใช้ Loop เพื่อลดขั้นตอนการทำงานที่ซ้ำซ้อน

4. สาระการเรียนรู้

- โครงสร้างการทำซ้ำ (Repetition Structure):
 - Loop (While/For): สัญลักษณ์ใน Flowgorithm (รูปร่างหกเหลี่ยม)
- ตัวนับ (Counter): ตัวแปรสำคัญที่ใช้กำหนดจำนวนรอบ (เช่น $i = 1, i = i + 1$)
- เงื่อนไขการหยุด (Terminating Condition): ถ้าไม่กำหนด Loop จะวนไปเรื่อยๆ ไม่มีวันจบ (Infinite Loop)

5. กิจกรรมการเรียนรู้

ขั้นนำ (Introduction) - 15 นาที

- Role Play "คัดลายมือ":

- ครูขออาสาสมัคร 1 คน ออกมาหน้ากระดาน
 - ครูสั่ง: "เขียนคำว่า 'หนูรักวิทยาการคำนวณ' 5 รอบ" (นักเรียนเขียน... เริ่มเมื่อ)
 - ครูถาม: "ถ้าครูสั่งให้เขียน 1,000 รอบ จะทำไหม?" (ไม่ทำ!)
2. Concept Link: "แต่นี่คืองานถนัดของคอมพิวเตอร์ วันนี้เราจะมาสร้าง 'หุ่นยนต์คัดลายมือ' ที่เขียนเป็นหมื่นจบได้ในพริบตาเดียว"

ขั้นสอน (Instruction) - 35 นาที

1. Loop ใน Flowgorithm:
 - ครูแนะนำสัญลักษณ์ Loop (While/For) ในโปรแกรม
 - Concept: "Loop เหมือนการวิ่งรอบสนาม เราต้องกำหนดว่า จะวิ่งกี่รอบ (Condition)"
2. Demo 1: The Punishment Writer (For Loop)
 - โจทย์: พิมพ์ "I will not be late." 10 ครั้ง
 - วิธีทำ:
 - Declare count (Integer)
 - Loop: count ตั้งแต่ 1 ถึง 10
 - Output: "I will not be late. รอบที่ " & count
 - End Loop
3. Demo 2: The Gatekeeper (While Loop)
 - โจทย์: ถามรหัสผ่านไปเรื่อยๆ จนกว่าจะตอบถูก
 - วิธีทำ:
 - Declare password (String)
 - Assign password = "" (ค่าว่างเริ่มต้น)
 - Loop: while password != "1234" (ตราบใดที่รหัสยังไม่ถูก ให้ทำวนไป)
 - Output: "ใส่รหัสผ่านครับ:"
 - Input: password
 - Output: "เย้! เข้าระบบสำเร็จ"

ขั้นปฏิบัติ (Practice) - 40 นาที

กิจกรรม: "Loop Master Challenges"

1. ให้นักเรียนจับคู่ทำภารกิจ (เลือกทำ 1 ภารกิจตามความสนใจ):
 - ภารกิจ A (สายภาษา): สร้างโปรแกรม "ฝึกท่องศัพท์" รับคำศัพท์จากผู้ใช้ 1 คำ แล้วให้โปรแกรม

พิมพ์ซ้ำๆ 20 รอบเพื่อช่วยจำ

- ภารกิจ B (สายเกม): สร้างโปรแกรม "ทายเลขลับ" (Guess the Number) กำหนดเลขลับไว้ (เช่น 55) แล้วให้เพื่อนทายไปเรื่อยๆ จนกว่าจะถูก (ใช้ While Loop)
- ภารกิจ C (สายโหด): สร้างโปรแกรม "แม่สูตรคูณ" รับเลขแม่สูตรคูณ (เช่น แม่ 2) แล้วแสดงผลคูณตั้งแต่ 2×1 ถึง 2×12

ขั้นสรุป (Conclusion) - 10 นาที

1. Show "Infinite Loop": ครูแกล้งเขียน Loop ที่ไม่มีเงื่อนไขจบให้ดู (เช่น `while 1 == 1`) แล้วรันโปรแกรม -> โปรแกรมจะรันไม่หยุดจนเครื่องค้าง
2. Lesson Learned: "พลังอันยิ่งใหญ่ มาพร้อมความรับผิดชอบ ถ้าเขียนเงื่อนไขผิด คอมพิวเตอร์จะทำงานจนตัวตาย (เครื่องค้าง) ดังนั้นต้องระวังเงื่อนไขการหยุดเสมอ!"

6. สื่อ/แหล่งการเรียนรู้

- ไฟล์เปรียบเทียบการทำงานของมนุษย์ (เหนื่อยล้า) vs หุ่นยนต์ในโรงงาน (ทำซ้ำได้ตลอด)
- ไฟล์แสดงการทำงานของรูปแบบต่างๆ
- ใบงานที่ 6: "The Loop Master"
- โปรแกรม Flowgorithm

7. การวัดและประเมินผล

จุดประสงค์การเรียนรู้	วิธีการวัดผล	เครื่องมือวัดผล	เกณฑ์การประเมิน
1. ด้านความรู้ (K) อธิบายหลักการทำงานของ การวนซ้ำ (Loop) ได้	ตรวจคำตอบการไล่ลูป (Trace) ในใบงานที่ 6 (Part 1)	ใบงานที่ 6 "The Loop Master"	ตอบถูกต้องร้อยละ 70 ขึ้นไป ถือว่า ผ่าน
2. ด้านทักษะ (P)เขียนผังงาน เพื่อสั่งงานซ้ำๆ ตามจำนวน รอบหรือเงื่อนไขที่กำหนดได้	ตรวจผลงานโปรแกรม Loop (ภารกิจ A/B/C)	แบบประเมิน ชิ้นงาน	ระดับคุณภาพพอใช้ขึ้นไป(เกณฑ์: โปรแกรมหยุดทำงานตามเงื่อนไข ไม่เกิด Infinite Loop)
3. ด้านคุณลักษณะ (A)เห็น ประโยชน์ของการใช้ คอมพิวเตอร์ลดภาระงาน ซ้ำซ้อน	สังเกตจากการตอบ คำถามและการสะท้อน ความคิด (Reflection)	แบบสังเกต พฤติกรรม	นักเรียนสามารถยกตัวอย่าง งานที่ควรใช้ Loop ช่วยทำได้ ในระดับ ผ่าน

เกณฑ์การให้คะแนน (Rubric) : พลังแห่งการทำซ้ำ (ใบงาน Loop Master)

ดีมาก (3 คะแนน):

- เลือกใช้ Loop (While/For) ได้เหมาะสมกับโจทย์
- กำหนดเงื่อนไขการหยุด Loop ได้ถูกต้อง (ไม่เกิด Infinite Loop)
- ผลลัพธ์แสดงออกมาครบถ้วนตามจำนวนรอบที่กำหนด

พอใช้ (2 คะแนน):

- ใช้ Loop ได้ และโปรแกรมทำงานซ้ำได้
- แต่อาจมีข้อผิดพลาดเรื่องจำนวนรอบ (Off-by-one error) เช่น ขาดไป 1 รอบ หรือเกินมา 1 รอบ

ปรับปรุง (1 คะแนน):

- ไม่ได้ใช้ Loop (ใช้วิธี Copy วางคำสั่งซ้ำๆ)
- หรือเขียนเงื่อนไขผิดทำให้โปรแกรมทำงานไม่หยุด (Infinite Loop) จนต้องบังคับปิด

ใบงานที่ 6: The Loop Master

รายวิชา: วิทยาการคำนวณ ม.4 | Codename:

Part 1: วงกลมแห่งชีวิต (Loop Logic)

จงพิจารณาผังงานต่อไปนี้ แล้วตอบว่าข้อความจะถูกพิมพ์ออกมากี่ครั้ง?

Flow: $i = 1 \rightarrow$ [Loop: $i \leq 3$] \rightarrow Output "Hello" $\rightarrow i = i + 1 \rightarrow$ (วนกลับ)

คำตอบ: ครั้ง

Part 2: ภารกิจวนซ้ำ (My Loop Mission)

เลือกภารกิจ: A.ฝึกท่องศัพท์ B.ทนายเลขลับ C.แม่สูตรคูณ

1. ออกแบบอัลกอริทึม (Algorithm Design):

จุดเริ่มต้น (Start)

.....

.....

.....

.....

.....

.....

จุดสิ้นสุด (End)

2. ผลลัพธ์หน้าจอ (Console Screen): วาดภาพสิ่งที่เกิดขึ้นเมื่อรันโปรแกรม

(ตัวอย่าง ภารกิจ B) ทนายเลขลับ: 20 ผิดครับ! ลองใหม่ ทนายเลขลับ: 50 ผิดครับ! ลองใหม่ ทนายเลขลับ: 55

ถูกต้อง! เก่งมาก

Part 3: คำถามวัดกัน (Think Harder)

ถ้าเราอยากเขียนโปรแกรม "นับถอยหลัง" (Countdown) เพื่อจุดระเบิดหรือปล่อยจรวด เริ่มจาก 10 ลงไปถึง 0 แล้วพิมพ์ว่า "BOOM!" เราจะต้องเขียนเงื่อนไขใน Loop อย่างไร?

ค่าเริ่มต้น (Start): $\text{count} = 10$

เงื่อนไขวนซ้ำ (Condition): $\text{count} \geq 0$

การเปลี่ยนแปลงค่า (Step): $\text{count} = \text{count} \dots\dots\dots$ (บวกหรือลบ?)

บันทึกท้ายแผนการจัดการเรียนรู้ที่ 6

1. While vs For:

ใน Flowgorithm เมนู **"For"** จะใช้ง่ายกว่าสำหรับโจทย์ "จำนวนรอบที่แน่นอน" (เช่น คัดลายมือ, สูตรคูณ) ส่วน **"While"** เหมาะกับโจทย์ "ไม่รู้จบเมื่อไหร่" (เช่น ทายรหัสผ่าน, ทายตัวเลข)

คำแนะนำ: สอนทั้งคู่ แต่ให้เลือกใช้ตามความเหมาะสมของโจทย์

2. กับดัก Infinite Loop: นักเรียนมักจะลืมใส่ตัวปรับค่า (Increment/Decrement) เช่น $i = i + 1$ ใน While Loop ทำให้ i เท่าเดิมตลอดกาล -> ครูต้องเดินดูและช่วยเช็ก

3. การเชื่อมโยงสายภาษา: เปรียบเทียบ Loop กับ "บทสวดมนต์" หรือ "ท่อนฮุคเพลง" ที่ต้องร้องซ้ำๆ ตามจำนวนรอบที่กำหนด

แผนการจัดการเรียนรู้ที่ 7

กลุ่มสาระการเรียนรู้วิทยาศาสตร์และเทคโนโลยี (วิทยาการคำนวณ)

หน่วยการเรียนรู้ที่ 3: Python for Creativity (Coding as a Language)

เรื่อง: สวัสดีไพทอน (Hello Python! & String Basics)

ระดับชั้น: มัธยมศึกษาปีที่ 4 (สายศิลป์-ภาษา/สังคม) เวลา: 2 คาบ (100 นาที)

1. มาตรฐานการเรียนรู้/ตัวชี้วัด

ว 4.2 ม.4/1: ประยุกต์ใช้แนวคิดเชิงคำนวณในการพัฒนาโครงงานที่มีการบูรณาการกับวิชาอื่นอย่างสร้างสรรค์ และเชื่อมโยงกับชีวิตจริง

2. สาระสำคัญ (Concept)

"Coding is Writing" การเขียนโปรแกรมภาษา Python ก็เหมือนการเขียนเรียงความหรือบทละคร เราแค่เปลี่ยนจากปากกาเป็นคีย์บอร์ด และเปลี่ยนภาษาพูดเป็นภาษาคอมพิวเตอร์ ในคาบนี้เราจะเรียนรู้ "คำศัพท์" คำแรก คือ print (พูด/แสดงผล) และ "ไวยากรณ์" การเก็บข้อความลงในตัวแปร เพื่อให้เราเริ่มสื่อสารกับคอมพิวเตอร์ได้จริงๆ

3. จุดประสงค์การเรียนรู้ (K-P-A)

ด้านความรู้ (K) : อธิบายโครงสร้างภาษา Python และเปรียบเทียบคำสั่ง Python กับสัญลักษณ์ Flowchart ได้

ด้านทักษะ (P) : เขียนโปรแกรมภาษา Python เพื่อแสดงผลข้อความ (String) และรับข้อมูล (Input) เบื้องต้นได้

ด้านคุณลักษณะ (A) : มีความกล้าที่จะพิมพ์โค้ดและไม่กลัวความผิดพลาด (Syntax Error)

4. สาระการเรียนรู้

1. สภาพแวดล้อมการเขียนโปรแกรม: แนะนำ Google Colab (หรือ Replit) เหตุผล: ใช้งานผ่านเว็บได้เลย ไม่ต้องลงโปรแกรม ไม่ต้องแก้ Path ให้อยุ่งยาก

2. คำสั่งพื้นฐาน:

- print() = แสดงผล (เหมือนสี่เหลี่ยมด้านขนาน Output)
- input() = รับค่า (เหมือนสี่เหลี่ยมด้านขนาน Input)

3. ตัวแปรและการต่อข้อความ:

- การสร้างตัวแปร: name = "Somchai"
- การเชื่อมคำ: ใช้เครื่องหมาย + (คล้าย & ใน Flowgorithm)

5. กิจกรรมการเรียนรู้

ขั้นนำ (Introduction) - 20 นาที

1. The Rosetta Stone: ครูเปิดภาพเปรียบเทียบขึ้นจอ:

ฝั่งซ้าย: รูป Flowchart สี่เหลี่ยมด้านขนานเขียนว่า Output "Hello"

ฝั่งขวา: โค้ด Python เขียนว่า print("Hello")

ครูถาม: "สองฝั่งนี้เหมือนหรือต่างกันยังไง?"

สรุป: "ความหมายเหมือนกันเป๊ะ แค่เปลี่ยนรูปร่างเป็นตัวหนังสือ ต่อไปนี้เราจะเลิกวาดรูป
แล้วมาพิมพ์กัน!"

2. Setup: ให้นักเรียนเข้า Google Colab และล็อกอินด้วย Gmail โรงเรียน หรือ ใช้ Thonny IDE

ขั้นสอน (Instruction) - 40 นาที

1. Hello World:

- ครูให้พิมพ์บรรทัดแรก: print("สวัสดีชาวโลก") แล้วกดปุ่ม Play

- กฎเหล็กไวยากรณ์: "ข้อความต้องอยู่ในวงเล็บ () และเครื่องหมายคำพูด "" เสมอ" (เหมือนเวลาเราเขียนบทพูดตัวละคร)

2. Variables (กล่องเก็บของ):

สอนประกาศตัวแปร: my_name = "Lisa" (สังเกตว่า Python ไม่ต้องใช้คำว่า Declare ไม่ต้องบอก Type มันฉลาดพอจะรู้เอง)

3. Interactive Program (ถาม-ตอบ):

ครูพาเขียนโค้ด 3 บรรทัดเปลี่ยนโลก:

```
print("คุณชื่ออะไร?")
```

```
name = input() # รับค่าจากคีย์บอร์ดมาใส่กล่อง name
```

```
print("สวัสดี " + name) # เอาคำว่าสวัสดี มาต่อกับ ชื่อ
```

4. Common Error (วัดซินป้องกัน Error):

- ครูลองลบวงเล็บปิดออก 1 ตัว แล้วรันให้ดู -> ขึ้น Error สีแดง

- สอนว่า: "ไม่ต้องตกใจ แถบสีแดงคือคอมไพเตอร์กำลังบอกว่า 'เธอเขียนผิดไวยากรณ์นะ ช่วยแก้ไขหน่อย' อ่านดูดีๆ ว่ามันฟ้องบรรทัดไหน"

ขั้นปฏิบัติ (Practice) - 30 นาที

กิจกรรม: "Intro Card" (นามบัตรดิจิทัล)

- ให้นักเรียนเขียนโปรแกรม Python แนะนำตัว โดยต้องมียอดประกอบ:
 - รับชื่อ (Input)
 - รับฉายา หรือ สิ่งที่ชอบ (Input)
 - แสดงผลออกมาเป็นประโยคเล่าเรื่องยาวๆ (Output + Concatenation)
- Challenge: ลองใช้ `\n` เพื่อขึ้นบรรทัดใหม่ใน print เดียว (เช่น `print("บรรทัด1 \n บรรทัด2")`)

ขั้นสรุป (Conclusion) - 10 นาที

- Reflection: "วันนี้ใครเจอ Error บ้าง? แก้ยังไง?"
- Takeaway: "เห็นไหมว่า Python ใจดีกว่าที่คิด ไม่ต้องประกาศ Int/String ให้วุ่นวาย เขียนเหมือนภาษาอังกฤษเลย คาบหน้าเราจะเอา Logic 'ถ้า/แล้ว' มาใส่ให้มันฉลาดขึ้นครับ"

6. สื่อ/แหล่งการเรียนรู้

- Google Colab: หรือโปรแกรม Thonny IDE
- Cheat Sheet: ตารางเปรียบเทียบคำสั่ง "Flowchart vs Python"
- ใบงานที่ 7: "Hello Python!"

7. การวัดและประเมินผล

จุดประสงค์การเรียนรู้	วิธีการวัดผล	เครื่องมือวัดผล	เกณฑ์การประเมิน
1. ด้านความรู้ (K) เปรียบเทียบคำสั่ง Flowchart กับคำสั่ง Python (print/input) ได้	ตรวจคำตอบในใบงานที่ 1.7 (Part 1: นักแปลภาษา)	ใบงานที่ 1.7 "Hello Python!"	ตอบถูกต้องร้อยละ 70 ขึ้นไป ถือว่า ผ่าน
2. ด้านทักษะ (P) เขียนโค้ด Python เพื่อรับค่าและแสดงผลข้อความ (String Concatenation) ได้	ตรวจการเขียนโค้ดใน Google Colab	แบบประเมินผลงานปฏิบัติ	ระดับคุณภาพพอใช้ขึ้นไป (เกณฑ์: Syntax ถูกต้อง, เชื่อมข้อความกับตัวแปรได้, โปรแกรมไม่ Error)
3. ด้านคุณลักษณะ (A) กล้าที่จะทดลอง และแก้ไขข้อผิดพลาด (Debug) ด้วยตนเอง	สังเกตพฤติกรรมขณะเขียนโค้ดและเมื่อเจอ Error	แบบสังเกตพฤติกรรม	นักเรียนพยายามอ่าน Error Message และแก้ไขโค้ดเบื้องต้นได้ ในระดับ ผ่าน

เกณฑ์การให้คะแนน (Rubrics) - สวัสดิ์ไพธอน (ใบงาน Hello Python)

ดีมาก (3 คะแนน): Syntax ถูกต้องแม่นยำ (วงเล็บครบ, เครื่องหมายคำพูดครบ), ใช้คำสั่ง input() รับค่า และนำมาเชื่อมต่อ (Concatenate) กับข้อความอื่นได้สมบูรณ์, มีการจัดรูปแบบเว้นวรรคสวยงามเมื่อแสดงผล

พอใช้ (2 คะแนน): เขียนโค้ดรันได้ รับค่าและแสดงผลได้, อาจมีจุดเล็กน้อยที่ต้องแก้ เช่น ลืมเว้นวรรคระหว่างชื่อ, หรือตั้งชื่อตัวแปรไม่สื่อความหมาย (a, b)

ปรับปรุง (1 คะแนน): มี Syntax Error จนโปรแกรมรันไม่ได้ (เช่น ลืมวงเล็บปิด, พิมพ์คำสั่งผิด), ไม่มีการใช้ตัวแปร (พิมพ์ข้อความ Fixed ตายตัว)

ใบงานที่ 1.7: Hello Python!

รายวิชา: วิทยาการคำนวณ ม.4 | Programmer Name:

Part 1: นักแปลภาษา (Code Translator)

จงแปลงสัญลักษณ์ Flowchart ต่อไปนี้ ให้เป็นคำสั่งภาษา Python

1. [Flowchart: Output "Good Morning"]

Python:

2. [Flowchart: Input name] (สมมติเก็บในตัวแปร user)

Python:

3. [Flowchart: Assign score = 100]

Python:

Part 2: ตามหาจุดผิด (Debug Detective)

โค้ดต่อไปนี้จะมีจุดผิดอยู่ (Error) ช่วยวงกลมจุดที่ผิด และเขียนแก้ไขให้ถูกต้อง

Code ที่ผิด:

Python

```
name = "BamBam"
```

```
print("I love " + name
```

```
print(My favorite color is Red)
```

Code ที่แก้ไขแล้ว:

บรรทัดที่ 2 แก้เป็น:

บรรทัดที่ 3 แก้เป็น:

Part 3: สร้างบทสนทนา (Coding Your Dialogue)

โจทย์: เขียนโปรแกรมถามคำถามสั้นๆ แล้วตอบกลับ ตัวอย่าง:

Bot: วันนี้วันอะไรครับ? Me: วันศุกร์ Bot: ขอให้เป็นวันศุกร์ที่สดใส นะครับ

พื้นที่เขียนโค้ดร่าง (Draft Code):

Python

1. แสดงคำถาม

```
print(".....")
```

2. รับคำตอบเก็บในตัวแปร answer

```
answer = .....
```

3. แสดงผลตอบกลับ โดยทวนคำตอบด้วย

```
print("....." + answer + ".....")
```

บันทึกท้ายแผนการจัดการเรียนรู้ที่ 7

1. เครื่องมือ (Tools): ถ้าอินเทอร์เน็ตโรงเรียนไม่เสถียร การใช้ Thonny IDE (ออฟไลน์) เป็นทางเลือกที่ดีมากครับ โปรแกรมเบาและหน้าต่างสะอาดเหมาะกับมือใหม่
2. f-strings: ในระดับสูงขึ้น เราจะใช้ `print(f"สวัสดี {name}")` แทนการใช้ `+` แต่สำหรับคาบแรก การใช้ `+` จะช่วยให้นักเรียนเข้าใจ Concept การ "นำมาต่อกัน" ได้เห็นภาพกว่าครับ (และตรงกับ Flowgorithm ที่ใช้ &)
3. Error Phobia: เด็กสายศิลป์มักจะตกใจกลัวแถบสีแดง (Error Message) ครูต้องสร้างบรรยากาศว่า "Error is your friend" (Error คือเพื่อนที่ช่วยเตือนเรา ไม่ใช่การตำว่าเราโง่)

แผนการจัดการเรียนรู้ที่ 8

กลุ่มสาระการเรียนรู้วิทยาศาสตร์และเทคโนโลยี (วิทยาการคำนวณ)

หน่วยการเรียนรู้ที่ 3: Python for Creativity

เรื่อง: สร้าง Chatbot เพื่อนคุย (Python Conditionals: If-Elif-Else)

ระดับชั้น: มัธยมศึกษาปีที่ 4 (สายศิลป์-ภาษา/สังคม) เวลา: 4 คาบ (200 นาที)

1. มาตรฐานการเรียนรู้/ตัวชี้วัด

ว 4.2 ม.4/1: ประยุกต์ใช้แนวคิดเชิงคำนวณในการพัฒนาโครงงานที่มีการบูรณาการกับวิชาอื่นอย่างสร้างสรรค์ และเชื่อมโยงกับชีวิตจริง

2. สาระสำคัญ (Concept)

"เว้นวรรคผิด ชีวิตเปลี่ยน (The Power of Indentation)" ในภาษา Python การบอกว่าคำสั่งไหนอยู่ภายใต้เงื่อนไข if ไม่ได้ใช้วงเล็บปีกกา { } เหมือนภาษาอื่น แต่ใช้ "การย่อหน้า" (Indentation) เปรียบเสมือนการเขียนกลอนที่ต้องมีการวางวรรคตอนให้ถูกต้อง คอมพิวเตอร์ถึงจะเข้าใจความหมาย การใช้คำสั่ง if-elif-else จะช่วยให้เราสร้าง Chatbot ที่สามารถโต้ตอบกับผู้ใช้ได้หลากหลายอารมณ์และสถานการณ์

3. จุดประสงค์การเรียนรู้ (K-P-A)

ด้านความรู้ (K) : อธิบายโครงสร้างคำสั่งเลือกทำ (if, if-else, if-elif-else) และความสำคัญของการย่อหน้า (Indentation) ได้

ด้านทักษะ (P) : เขียนโปรแกรม Python เพื่อตรวจสอบเงื่อนไขทางตรรกะและสร้างบทสนทนาโต้ตอบ (Chatbot) ได้

ด้านคุณลักษณะ (A) : มีความรอบคอบในการเขียนโค้ด (ระวังเรื่องเครื่องหมาย : และการย่อหน้า)

4. สาระการเรียนรู้

1. ไวยากรณ์ Python:

- คำสั่ง if (ถ้า...), else (ถ้าไม่...), elif (ถ้าไม่...แต่ถ้า...)
- เครื่องหมาย Colon (:): จบประโยคเงื่อนไข
- Indentation: การกด Tab เพื่อระบุขอบเขตคำสั่ง

2. ตัวดำเนินการเปรียบเทียบ: == (เท่ากับ), != (ไม่เท่ากับ), >, <

5. กิจกรรมการเรียนรู้

คาบที่ 1-2: ทางแยกวัดใจ (Basic If-Else) - 100 นาที

ขั้นนำ (Introduction) - 10 นาที

1. The Tab Key: ครูให้นักเรียนมองหาปุ่ม Tab บนคีย์บอร์ด แล้วถามว่า "ปกติปุ่มนี้ไว้ทำอะไร?" (ย่อหน้า)
2. Concept: "ใน Python ปุ่มนี้คือปุ่มวิเศษ ถ้าเราไม่กด Tab คอมพิวเตอร์จะไม่ว่าคำสั่งนี้เป็นลูกน้องของ if"

ขั้นสอน (Instruction) - 40 นาที

1. Syntax Anatomy: ครูเขียนโครงสร้างบนกระดาน เน้นสีแดงที่เครื่องหมาย : และการย่อหน้า Python

```
password = input("ใส่รหัสผ่าน: ")
if password == "1234":           # อย่าลืม Colon!
    print("ถูกต้องนะคร้าบ")      # ต้องย่อหน้า (กด Tab)
else:                             # else ต้องอยู่ตรงกับ if
    print("ผิดคร้าบ!")          # ต้องย่อหน้า
```

2. Live Coding: พาทำโปรแกรม "Bot คัดกรองคน" (Age Checker)

- รับอายุ -> ถ้า > 18 ให้เข้าผับได้ -> ถ้าไม่ใช่ให้ไปกินนม

ขั้นปฏิบัติ (Activity 1) - 50 นาที : "Password Gate"

ให้นักเรียนเขียนโปรแกรมล็อกอินเข้าระบบลับ

โจทย์: กำหนดรหัสผ่านเอง (เป็นคำภาษาไทยก็ได้) ถ้าพิมพ์ถูกให้แสดงข้อความต้อนรับ

ถ้าผิดให้แจ้งเตือน

จุดเช็ก: เติมนูที่ละเครื่อง ใครลืมย่อหน้า หรือลืม : ให้เกดทันที่

คาบที่ 3-4: ทางเลือกที่หลากหลาย (Elif & Chatbot Project) - 100 นาที

ขั้นนำ (Introduction) - 10 นาที

1. Scenario: "ชีวิตไม่ได้มีแค่ ใช่/ไม่ใช่ บางทีเรามีหลายตัวเลือก เช่น ตัดเกรด (A/B/C/D) หรือเลือกเมนูอาหาร"
2. Introduce elif: ย่อมาจาก Else If แปลว่า "งั้นถ้าเป็นอันนี้ล่ะ..."

ชั้นสอน (Instruction) - 40 นาที

1. Structure:

Python

```
score = int(input("คะแนนเท่าไร: ")) # แปลงเป็นตัวเลขด้วย int()
```

```
if score >= 80:
```

```
    print("เกรด 4")
```

```
elif score >= 70:
```

```
    print("เกรด 3")
```

```
else:
```

```
    print("สู้ต่อไปนะ")
```

2. Project Idea: สาธิต "หมวกคัดสรร (Sorting Hat)" จากแฮร์รี่ พอตเตอร์

- ถาม: "คุณกล้าหาญไหม?" -> ถ้าใช่ -> กริฟฟินดอร์

- ถาม: "คุณฉลาดไหม?" -> ถ้าใช่ -> เรเวนคลอ

ชั้นปฏิบัติ (Activity 2) – 50 นาที : "My Chatbot Buddy" (โปรเจกต์จับหน่วย)

- ให้นักเรียนจับคู่ สร้าง Chatbot ในธีมที่สนใจ (ใช้เวลา 50 นาที)

- Theme A: บอทหมอดู: ถามวันเกิด/เดือนเกิด แล้วทำนายดวง

- Theme B: บอทแนะนำร้านอาหาร: ถามว่าชอบกินเผ็ดไหม? ชอบเส้นหรือข้าว? แล้วแนะนำเมนู

- Theme C: บอทจิตแพทย์: ถามอาการ (เครียด/เหงา/เหนื่อย) แล้วให้คำแนะนำดีๆ

- Showcase: ให้สลับกันเล่นกับเพื่อนข้างๆ

6. สื่อ/แหล่งการเรียนรู้

- Cheat Sheet: ภาพสรุปโครงสร้าง If-Else และการวางตำแหน่งย่อหน้า

- ไฟล์ตัวอย่างโค้ด Chatbot ต้นแบบ

- ใบงานที่ 1.8: "My Chatbot Design"

7. การวัดและประเมินผล

จุดประสงค์การเรียนรู้	วิธีการวัดผล	เครื่องมือวัดผล	เกณฑ์การประเมิน
1. ด้านความรู้ (K) อธิบาย	การตอบคำถามในชั้น	แบบสังเกต	อธิบายได้ถูกต้องว่าต้องย่อ
ความสำคัญของการย่อหน้า	เรียนหรือการตรวจโค้ด	พฤติกรรม	หน้าหลังเครื่องหมาย Colon
(Indentation) ใน Python ได้			ในระดับ ผ่าน

2. ด้านทักษะ (P) พัฒนา Chatbot ที่มีเงื่อนไข if-elif-else ตอบโต้ได้เหมาะสมผล	ตรวจสอบผลงานโปรแกรม Chatbot	แบบประเมิน ชิ้นงาน	ระดับคุณภาพพอใช้ขึ้นไป (เกณฑ์: มีทางเลือกอย่างน้อย 3 ทาง, ย่อหน้าถูกต้อง, บทตอบตรงคำถาม)
3. ด้านคุณลักษณะ (A) มีความละเอียดรอบคอบในการเขียนโค้ดตามโครงสร้างภาษา	ตรวจสอบความถูกต้องของ Syntax (เครื่องหมาย : และการย่อหน้า)	แบบประเมินผล งานปฏิบัติ	โค้ดมีความถูกต้องตามหลักไวยากรณ์ Python ไม่มี Syntax Error ในระดับ ผ่าน

เกณฑ์การให้คะแนนชิ้นงาน (Rubric) - ใบงาน Chatbot เพื่อนคุย (โปรแกรม Chatbot)

ดีมาก (3 คะแนน):

- ใช้โครงสร้าง if-elif-else ได้ถูกต้อง รองรับเงื่อนไขได้ 3 ทางเลือกขึ้นไป
- การย่อหน้า (Indentation) ถูกต้องตามหลักภาษา Python 100%
- บทสนทนามีความสละสลวย เป็นธรรมชาติ และตอบโต้ได้ตรงเงื่อนไข

พอใช้ (2 คะแนน):

- ใช้โครงสร้าง if-else (2 ทางเลือก) ได้ถูกต้อง
- การย่อหน้าถูกต้องเป็นส่วนใหญ่ แต่อาจมีผิดพลาดบ้างเล็กน้อยที่ต้องแก้
- บทสนทนาเรียบง่าย ถามคำ-ตอบคำ

ปรับปรุง (1 คะแนน):

- เขียนเงื่อนไขผิดไวยากรณ์ (เช่น ลืม Colon :) หรือย่อหน้าผิดจนเกิด IndentationError
- Logic ผิดพลาด (เช่น เงื่อนไขซ้อนกันจนทำงานไม่ถึง Else)

ใบงานที่ 8: My Chatbot Design

รายวิชา: วิทยาการคำนวณ ม.4 | Chatbot Name:

Part 1: ออกแบบบทสนทนา (Dialogue Tree)

ก่อนเขียนโค้ด ลองวาดแผนผังบทสนทนาของบอทเราก่อน

เริ่มของบอท: (เช่น หมอดู, แนะนำหนังสือ, ทายนิสัย)

1. คำถามแรก:

- ทางเลือก 1 (เงื่อนไข): ถ้าตอบว่า -> บอทตอบ:

- ทางเลือก 2 (เงื่อนไข): ถ้าตอบว่า -> บอทตอบ:

- ทางเลือก 3 (อื่นๆ): -> บอทตอบ:

Part 2: เขียนโค้ดจริง (Coding)

นำแผนผังข้างบนมาแปลงเป็น Python (อย่าลืมย่อหน้านะ!)

Python

1. รับค่าคำตอบ (Input)

```
user_answer = input(".....")
```

2. ตรวจสอบเงื่อนไข (Logic)

```
if user_answer == ".....":
```

```
    print(".....")
```

```
elif user_answer == ".....":
```

```
    print(".....")
```

```
else:
```

```
    print("ฉันไม่เข้าใจคำตอบของคุณ ลองตอบใหม่นะ")
```

บันทึกท้ายแผนการจัดการเรียนรู้ที่ 8

1. IndentationError: unexpected indent

- อาการ: บรรทัดย่อหน้าไม่เท่ากัน หรือไปย่อหน้าในที่ไม่ควรย่อ
- วิธีแก้: บอกนักเรียนให้ลบช่องว่างหน้าบรรทัดนั้นทิ้งให้หมด แล้วกด Tab ใหม่เพียง 1 ครั้ง

2. SyntaxError: invalid syntax (ที่บรรทัด if/elif/else)

- สาเหตุ: 90% คือลืมใส่เครื่องหมาย Colon (:) ท้ายประโยค

3. String Comparison:

- ถ้านักเรียนรับค่าตัวเลขมาคำนวณ (เช่น คะแนน, อายุ) อย่าลืมสอนใช้ int(input()) เพื่อแปลงข้อความเป็นตัวเลขก่อน ไม่งั้นจะเทียบค่ามากกว่า/น้อยกว่าไม่ได้ (>,<)

4. การเปรียบเทียบคำไทย:

- Python ฉลาดพอที่จะเทียบ "ใช่" == "ใช่" ได้ แต่ต้องระวังเรื่องสระลอยหรือการพิมพ์ผิดเล็กน้อย

อาจจะแนะนำให้เด็กทำ Choice เป็นตัวเลข (กด 1, กด 2) แทนการพิมพ์ยาวๆ หากเจอปัญหาเยอะ

ตัวอย่างโค้ด Chatbot ภาษา python

ตัวอย่างที่ 1 : Chatbot พื้นฐาน (ใช้ if-else)	ตัวอย่างที่ 2 : Chatbot แบบใช้ Dictionary
<pre>print("🤖 สวัสดี ฉันคือ Chatbot") print("พิมพ์ 'ออก' เพื่อจบการสนทนา") while True: user_input = input("คุณ: ") if user_input == "ออก": print("🤖 บ๊ายบาย ไว้คุยกันใหม่") break elif user_input == "สวัสดี": print("🤖 สวัสดีจ้า") elif user_input == "ชื่ออะไร": print("🤖 ฉันชื่อ PythonBot") elif user_input == "ทำอะไรได้": print("🤖 ฉันตอบคำถามง่ายๆ ได้") else: print("🤖 ฉันยังไม่เข้าใจ ลองพิมพ์ใหม่ดูนะ")</pre>	<pre>responses = { "สวัสดี": "สวัสดีจ้า 😊", "ชื่ออะไร": "ฉันชื่อ DictBot", "ทำอะไรได้": "ฉันตอบคำถามจากคลังความรู้ได้", "python คืออะไร": "Python คือภาษาโปรแกรมที่อ่านง่าย" } print("🤖 ยินดีต้อนรับสู่ DictBot") print("พิมพ์ 'ออก' เพื่อจบ") while True: user_input = input("คุณ: ") if user_input == "ออก": print("🤖 แล้วเจอกันใหม่") break if user_input in responses: print("🤖 ", responses[user_input]) else: print("🤖 ฉันยังไม่รู้คำตอบนี้")</pre>

ใบความรู้ Chatbot ถาม-ตอบเกี่ยวกับ “ภาษาไทย” เขียนด้วย Python

```
print("🤖 แชตบอทภาษาไทย")
print("ถามเรื่องภาษาไทยได้เลย")
print("พิมพ์ 'ออก' เพื่อจบการสนทนา\n")
qa = {
    #ความรู้พื้นฐาน
    "ภาษาไทยคืออะไร": "ภาษาไทยเป็นภาษาประจำชาติของประเทศไทย",
    "พยัญชนะไทยมีกี่ตัว": "พยัญชนะไทยมี 44 ตัว",
    "สระไทยมีกี่รูป": "สระไทยมี 32 รูป",
    "วรรณยุกต์ไทยมีกี่รูป": "วรรณยุกต์ไทยมี 4 รูป ได้แก่ ไม้เอก ไม้โท ไม้ตรี และไม้จัตวา",
    "คำควบกล้ำคืออะไร": "คำควบกล้ำคือคำที่มีพยัญชนะ 2 ตัวออกเสียงควบกัน",
    "ประโยคคืออะไร": "ประโยคคือกลุ่มคำที่เรียงกันแล้วมีความหมายสมบูรณ์",
    "คำพ้องเสียงคืออะไร": "คำพ้องเสียงคือคำที่ออกเสียงเหมือนกันแต่เขียนต่างกันและความหมายต่างกัน",
    "คำสุภาพคืออะไร": "คำสุภาพคือคำที่ใช้พูดอย่างสุภาพเหมาะสมกับกาลเทศะ"
}

while True:
    question = input("คุณ: ")
    if question == "ออก":
        print("🤖 ขอขอบคุณที่เรียนภาษาไทยด้วยกัน 🙌")
        break
    if question in qa:
        print("🤖", qa[question])
    else:
        print("🤖 ฉันยังไม่รู้คำตอบ ลองถามใหม่หรือใช้คำถามอื่นนะ")
```

แผนการจัดการเรียนรู้ที่ 9

กลุ่มสาระการเรียนรู้วิทยาศาสตร์และเทคโนโลยี (วิทยาการคำนวณ)

หน่วยการเรียนรู้ที่ 3: Python for Creativity

เรื่อง: สนุกกับลูปและการสุ่ม (Loops for Fun & Randomness)

ระดับชั้น: มัธยมศึกษาปีที่ 4 (สายศิลป์-ภาษา/สังคม) เวลา: 4 คาบ (200 นาที)

1. มาตรฐานการเรียนรู้/ตัวชี้วัด

ว 4.2 ม.4/1: ประยุกต์ใช้แนวคิดเชิงคำนวณในการพัฒนาโครงการที่มีการบูรณาการกับวิชาอื่นอย่างสร้างสรรค์ และเชื่อมโยงกับชีวิตจริง

2. สาระสำคัญ (Concept)

"วนซ้ำจนกว่าจะพอใจ (Repetition) และ ความน่าตื่นเต้นของการสุ่ม (Randomness)" การทำงานซ้ำๆ เป็นงานถนัดของคอมพิวเตอร์ เราใช้คำสั่ง for เพื่อทำซ้ำตามจำนวนรอบที่กำหนด และ while เพื่อทำซ้ำไปเรื่อยๆ จนกว่าเราจะสั่งให้หยุด (เช่น การคุยกับ Chatbot ต่อเนื่อง) นอกจากนี้ การเพิ่มความไม่แน่นอนด้วยการ "สุ่ม" (Random) จะทำให้โปรแกรมของเราดูฉลาดและน่าสนใจขึ้น เหมือนการเสี่ยงเข็มชีหรือการสุ่มกาษาปอง

3. จุดประสงค์การเรียนรู้ (K-P-A)

ด้านความรู้ (K) : อธิบายโครงสร้างการทำซ้ำ (for, while) และการใช้งานไลบรารี random เบื้องต้นได้
ด้านทักษะกระบวนการ (P) : เขียนโปรแกรม Python เพื่อวนลูปแสดงผลข้อความ และสร้างระบบสุ่มตัวเลข/คำตอบได้

ด้านเจตคติ (A) : สนุกกับการสร้างผลลัพธ์ที่หลากหลายและคาดเดาไม่ได้

4. สาระการเรียนรู้

- For Loop: การวนซ้ำแบบรู้จำนวนรอบ
 - คำสั่ง for i in range(n):
- While Loop: การวนซ้ำแบบตรวจสอบเงื่อนไข (Infinite Chat)
 - คำสั่ง while True: และ break (หยุดลูป)
- Module Random: การนำเข้าชุดคำสั่งสุ่ม
 - import random
 - random.randint(a, b) สุ่มตัวเลข
 - random.choice(list) สุ่มเลือกของจากรายการ (เก็ร็นำเรื่อง List)

5. กิจกรรมการเรียนรู้

คาบที่ 1-2: มนต์คาถาแยกเงาพันธุ์ร่าง (For Loop & Patterns) - 100 นาที

ขั้นนำ (Introduction) - 10 นาที

1. Challenge: "ใครพิมพ์คำว่า 'รักนะ' ให้ครู 20 บรรทัดได้เร็วที่สุด?" (เด็กๆ จะก้อปวางรัวๆ)
2. The Magic Spell: ครูโชว์โค้ด 2 บรรทัด:

Python

```
for i in range(20):
```

```
    print("รักนะ")
```

กดรันปุ๊บ ออกมา 20 บรรทัดปั๊บ!

ขั้นสอน (Instruction) - 40 นาที :

1. Anatomy of range():
 - range(5) = 0, 1, 2, 3, 4 (นับนิ้วได้ 5 นิ้ว แต่เริ่มที่ 0)
 - ลองให้ print(i) ดู เพื่อให้เห็นตัวเลขที่วิ่งอยู่
2. Activity "Text Art": สอนสร้างงานศิลปะจากตัวอักษร
 - พิมพ์ดาวเรียงกันเป็นสามเหลี่ยม (ใช้หลักการคูณ String * x i)

Python

```
for i in range(1, 10):
```

```
    print("*" * i) # บรรทัดที่ 1 มี 1 ดวง, บรรทัดที่ 2 มี 2 ดวง...
```

ขั้นปฏิบัติ (Practice) - 40 นาที :

ให้นักเรียนสร้าง "ลายผ้าจีนดิจิทัล" หรือ "กำแพงเมือง" โดยใช้ For Loop พิมพ์ลวดลายซ้ำๆ สลับกัน
อย่างสวยงาม

คาบที่ 3-4: บอทขี้เมาท์ที่ไม่ยอมหยุด (While Loop & Random) - 100 นาที

ขั้นนำ (Introduction) - 10 นาที

1. Problem: "Chatbot ที่เราทำคราวที่แล้ว ถามปุ๊บ ตอบปั๊บ แล้วโปรแกรมก็ดับไป... เหงาไหม?"
2. Solution: "เราจะขังบอทไว้ในลูป ให้มันคุยกับเราไปเรื่อยๆ จนกว่าเราจะพิมพ์ว่า 'ไปละนะ'"

ขั้นสอน (Instruction) - 40 นาที

1. The Infinite Loop: สอนใช้ while True: (ทำตลอดกาล) คู่กับ break (ออกจากลูป)

```
while True:
```

```
    text = input("คุยกับบอท: ")
```

```
if text == "ออก":  
    print("บ๊ายบาย")  
    break # พระเอกขี่ม้าขาวมาช่วยหยุดลูป  
else:  
    print("คุณพูดว่า: " + text)
```

2. The Randomness (พระเอกของงาน):

```
- สอน import random  
- ทำ "บอทเสี่ยงทาย": ถามอะไรมา ตอบ ใช่/ไม่ใช่ แบบสุ่ม  
import random  
num = random.randint(1, 2)  
if num == 1: print("ใช่เลย!")  
else: print("อย่าหาทำ")
```

ขั้นปฏิบัติ (Activity 3) - 40 นาที : โปรแกรม "แม่หมอดิจิทัล (เกมทายใจ)"

โจทย์: ให้นักเรียนเขียนโปรแกรมให้คอมพิวเตอร์สุ่มเลขมา 1 ตัว (เช่น 1-100) แล้วให้ผู้เล่นทายไปเรื่อยๆ โดยโปรแกรมจะคอยบอกว่าเลขที่ทายมานั้น "มากเกินไป" หรือ "น้อยเกินไป" จนกว่าจะทายถูก

Algorithm (Logic ที่ต้องสอน):

1. คอมพิวเตอร์สุ่มเลขเก็บไว้ในใจ (ตัวแปร secret)
2. เริ่มวนลูปถามคำถาม (ใช้ while True)
3. รับค่าตัวเลขจากผู้เล่น (ตัวแปร guess)
4. เช็กรเงื่อนไข:
 - ถ้า $guess > secret$ -> บอกว่า "มากเกินไป"
 - ถ้า $guess < secret$ -> บอกว่า "น้อยเกินไป"
 - ถ้า $guess == secret$ -> บอกว่า "ถูกต้อง!" แล้วสั่งจบเกม (break)

6. สื่อ/แหล่งการเรียนรู้

- ไฟล์ตัวอย่างโค้ด Text Art Pattern, Infinite Chatbot และ The Mind Reader
- ภาพ Infographic: Random Number Generator (ตัวสุ่มเลข)
- ใบงานที่ 9: "Fun with Loops"

7. การวัดและประเมินผล

จุดประสงค์การเรียนรู้	วิธีการวัดผล	เครื่องมือวัดผล	เกณฑ์การประเมิน
1. ด้านความรู้ (K)อธิบายหน้าที่ของคำสั่ง while True, break และ random ได้	ตรวจใบงานที่ 9 (Part 1: ทายผลลัพธ์)	ใบงานที่ 9 "Fun with Loops"	ตอบถูกต้องร้อยละ 70 ขึ้นไป ถือว่า ผ่าน
2. ด้านทักษะ (P)เขียนโปรแกรมเกมทายใจที่มีการสุ่มตัวเลขและวงลูปรับค่าได้	ตรวจผลงานโปรแกรมเกมทายใจ (High-Low Game)	แบบประเมิน ชิ้นงาน	ระดับคุณภาพพอใช้ขึ้นไป (เกณฑ์: สุ่มเลขได้, วงลูปถามซ้ำได้, จบเกมเมื่อทายถูก)
3. ด้านคุณลักษณะ (A)สนุกสนานกับการทดสอบโปรแกรมและแลกเปลี่ยนเรียนรู้	สังเกตบรรยากาศการทำกิจกรรมและการเล่นเกมของเพื่อน	แบบสังเกตพฤติกรรม	นักเรียนมีความกระตือรือร้นในการทดสอบและให้ผลป้อนกลับเพื่อน ในระดับ ผ่าน

เกณฑ์การให้คะแนนชิ้นงาน (Rubric) - ใบงานสนุกกับลูปและการสุ่ม (โปรแกรม High-Low Game)

ดีมาก (3 คะแนน):

- ใช้ import random และสุ่มตัวเลขได้ถูกต้อง
- ใช้ while loop รับค่าซ้ำๆ และมีเงื่อนไข break เพื่อจบเกมเมื่อทายถูก
- มีการแจ้งเตือน "มากไป/น้อยไป" เพื่อใ้ผู้เล่น (Logic สมบูรณ์)

พอใช้ (2 คะแนน):

- มีการสุ่มเลขและวงลูปรับค่า
- แต่ Logic การใ้ หรือเงื่อนไขการจบเกมอาจยังไม่สมบูรณ์ (เช่น ทายถูกแล้วไม่หยุดลูป)

ปรับปรุง (1 คะแนน):

- เขียน Loop ไม่รู้จบ (Infinite Loop) โดยไม่มีทางออก
- หรือโปรแกรมรันไม่ได้เนื่องจาก Syntax Error

ใบงานที่ 9: Fun with Loops

รายวิชา: วิทยาการคำนวณ ม.4 | Code Artist:

Part 1: ทายผลลัพธ์ (Output Prediction)

โค้ดนี้จะแสดงผลออกมาเป็นรูปอะไร? วาดภาพคำตอบลงในกล่อง

```
for i in range(3):  
    print("Meow")  
print("Woof")  
ผลลัพธ์: [ ..... ] [ ..... ] [ ..... ] [ ..... ]
```

Part 2: ศิลปะจากลูป (Loop Art)

จงเขียนโค้ดเพื่อสร้างภาพบันไดดาวดังนี้:

<p>Output:</p> <pre>* ** *** **** *****</pre>	<p>Code:</p> <pre>for i in range(.....,): print("*" *)</pre>
---	--

Part 3: บอทเสียงทาย (Random Bot)

โจทย์: สร้างบอทที่คุยได้เรื่อยๆ (While Loop) เมื่อเราถามอะไรไป บอทจะสุ่มตอบเลข 0-99 กลับมา (เหมือนทอย) และจะหยุดเมื่อเราพิมพ์ว่า "exit"

```
import random # 1. เรียกใช้ไลบรารี  
print("--- ยินดีต้อนรับสู่สำนักเจ้าพ่อ Python ---")  
while True: # 2. เริ่มวนลูปไม่รู้จบ  
    question = input("ถามมาสิโยม (พิมพ์ exit เพื่อออก): ")  
    # 3. ตรวจสอบเงื่อนไขเพื่อหยุด  
    if question == ".....":  
        print("เจริญพร ลาก่อน")
```

```
..... # คำสั่งหยุดลูป
# 4. ถ้ายังไม่หยุด ให้สุ่มเลขตอบ
else:
    lucky_number = random.randint(0, 99)
    print("เลขเดี๋ยวนี้นี้คือ... " + str(lucky_number)) # แปลงเลขเป็นข้อความก่อนบวก
```

Part 4: เกมทายใจ (The Mind Reader)

ภารกิจ: สร้างเกมทายตัวเลขปริศนา โดยทำตามผังงาน (Flowchart) ด้านล่างนี้

Preparation (เตรียมตัว):

เรียกใช้ไลบรารีสุ่ม: import

สุ่มเลข 1-100 เก็บในตัวแปร: secret = random.randint(1, 100)

Coding (เขียนโค้ด):

while True:

รับตัวเลขจากผู้เล่น (อย่าลืมแปลงเป็น int นะ!)

guess = int(input("ป้อนตัวเลขที่คุณทาย: "))

เงื่อนไขที่ 1: ทายมากเกินไป

if guess > secret:

print(".....")

เงื่อนไขที่ 2: ทายน้อยไป

elif guess < :

print(".....")

เงื่อนไขที่ 3: ทายถูก (จบเกม)

else:

print("เย้! ถูกต้องแล้ว")

..... # คำสั่งหยุดลูป

Challenge (สำหรับคนชอบท้าทาย):

ลองเพิ่มตัวแปร count เพื่อ นับจำนวนครั้ง ที่ทาย

เมื่อทายถูก ให้บอกด้วยว่า "คุณทายไปทั้งหมด ครั้ง"

บันทึกท้ายแผนการจัดการเรียนรู้ที่ 8 - 9

1. ระวัง Loop นรก (Infinite Loop):

- ถ้านักเรียนเขียน while True: แล้วลืมเขียน break หรือเงื่อนไขหยุดผิด โปรแกรมจะรันไม่หยุด
- วิธีแก้: ใน Google Colab ให้กดปุ่ม Stop (สี่เหลี่ยม) ที่มุมบนซ้ายของเซลล์โค้ดเพื่อบังคับหยุด

2. Range Start/Stop:

- ย้ำเสมอว่า range(5) คือ 0-4 (ไม่ถึง 5)
- ถ้าอยากได้ 1-5 ต้องใช้ range(1, 6)

3. Type Casting in Print:

- บรรทัด print("เลขคือ " + str(number)) สำคัญมาก เพราะ Python เอา Text + Number โดยตรงไม่ได้ (Error) ต้องแปลงเลขเป็น Text ด้วย str() ก่อนเสมอ (จุดนี้เด็กสายศิลป์จะงงบ่อย)

4. จุดตายเด็กสายศิลป์: คือลืมใส่ int(...) ครอบ input(...) ครับ เพราะโจทย์นี้ต้องเอาตัวเลขมาเทียบค่า (มากกว่า/น้อยกว่า) ถ้าเป็น String จะเทียบผิด

5. เทคนิคการใบ้: ครูอาจจะบอกนักเรียนว่า Algorithm นี้คือหลักการเดียวกับ Binary Search (การค้นหาแบบแบ่งครึ่ง) ซึ่งเป็นวิธีหาของที่เร็วที่สุดวิธีหนึ่งในโลกคอมพิวเตอร์

ตัวอย่าง Code ภาษาไพธอน

ตัวอย่างที่ 1 : Infinite Chatbot แบบสุ่มคำตอบ	ตัวอย่างที่ 2 : The Mind Reader
<pre>import random responses = ["อืม... น่าสนใจนะ", "ช่วยเล่าเพิ่มอีกหน่อยสิ", "ฉันเห็นด้วยนะ", "คิดเหมือนกันเลย", "ขอคิดแป๊บนึง 🤔"] print("🤖 Infinite Chatbot พร้อมคุย") print("พิมพ์ exit เพื่อออก") while True: user_text = input("คุณ: ") if user_text == "exit": print("🤖 บ๊ายบาย 👋 ") break else: print("🤖 ", random.choice(responses))</pre>	<pre>import random predictions = ["ฉันสัมผัสได้ว่าเธอกำลังคิดถึงเลข 7", "ในใจเธอกำลังลี้ลับอยู่ใช่ไหม", "สิ่งที่เธอคิดอยู่ตอนนี้คือเรื่องอนาคต", "เธอกำลังคิดถึงคนสำคัญคนหนึ่ง", "ฉันเห็นภาพบางอย่างลางๆ ในหัวเธอ", "ใจของเธอกำลังสับสนเล็กน้อย"] print("🌟 ยินดีต้อนรับสู่ The Mind Reader ") print("คิดอะไรไว้ในใจแล้วพิมพ์มา") print("พิมพ์ exit เพื่อออก\n") while True: thought = input("คุณ: ") if thought == "exit": print("🌟 พลังจิตจางหาย... ลาก่อน") break else: print("🌟 ", random.choice(predictions))</pre>

แผนการจัดการเรียนรู้ที่ 10

กลุ่มสาระการเรียนรู้วิทยาศาสตร์และเทคโนโลยี (วิทยาการคำนวณ)

หน่วยการเรียนรู้ที่ 3: Python for Creativity

เรื่อง: คลังข้อมูลมหาสงก (Lists & Random Choice)

ระดับชั้น: มัธยมศึกษาปีที่ 4 (สายศิลป์-ภาษา/สังคม) เวลา: 4 คาบ (200 นาที)

1. มาตรฐานการเรียนรู้/ตัวชี้วัด

ว 4.2 ม.4/1: ประยุกต์ใช้แนวคิดเชิงคำนวณในการพัฒนาโครงการที่มีการบูรณาการกับวิชาอื่นอย่างสร้างสรรค์ และเชื่อมโยงกับชีวิตจริง

2. สาระสำคัญ (Concept)

"กล่องใบเดียวไม่พอ ต้องขอเป็นลัง (Managing Groups of Data)" ถ้าเราต้องเก็บชื่อเพื่อน 40 คน การประกาศตัวแปร name1, name2, ... name40 เป็นเรื่องที่เหนื่อยและจัดการยากมาก ในทางคอมพิวเตอร์เรามีโครงสร้างข้อมูลที่เรียกว่า List (รายการ) ที่ช่วยให้เราเก็บข้อมูลเป็นกลุ่มก้อนได้ ภายใต้อันเดียว และสามารถดึงข้อมูลออกมาใช้ได้ผ่าน "หมายเลขลำดับ" (Index)

3. จุดประสงค์การเรียนรู้ (K-P-A)

ด้านความรู้ (K) : อธิบายโครงสร้างของ List, การอ้างอิงตำแหน่ง (Index), และคำสั่งจัดการ List เบื้องต้น (append, len) ได้

ด้านทักษะ (P) : เขียนโปรแกรมเพื่อสร้าง List ของข้อความ และใช้คำสั่ง random.choice() เพื่อสุ่มเลือกข้อมูลใน List มาแสดงผลได้

ด้านคุณลักษณะ (A) : เห็นประโยชน์ของการจัดหมวดหมู่ข้อมูลเพื่อลดความซับซ้อนของโค้ด

4. สาระการเรียนรู้

- การสร้าง List: การใช้วงเล็บก้ามปู [] และค้นข้อมูลด้วยจุลภาค ,
 - fruits = ["Apple", "Banana", "Orange"]
- ดัชนี (Index): ตำแหน่งของข้อมูล (เริ่มนับที่ 0 เสมอ!)
 - fruits[0] คือ "Apple"
- การจัดการข้อมูล:
 - append(): เพิ่มข้อมูลต่อท้าย
 - random.choice(list): สุ่มเลือกข้อมูล 1 ตัวจาก List (พระเอกของคาบนี้)

5. กิจกรรมการเรียนรู้

คาบที่ 1-2: รู้จักกับลิสต์คเกอร์ (List Basics) - 100 นาที

ขั้นนำ (Introduction) :

1. Quiz: "ถ้าครูอยากเก็บรายชื่อเพื่อนทั้งห้องไว้ในโปรแกรม ต้องประกาศตัวแปรกี่ตัว?"
(นักเรียนตอบ: เยอะมาก!)
2. Solution: "จะดีกว่าไหมถ้าเรามีสมุดรายชื่อเล่มเดียว แล้วเปิดหาหน้าที่ต้องการได้เลย?
นี่คือ Concept ของ List"

ขั้นสอน (Instruction) :

1. Syntax: สอนสร้าง List แบบง่ายๆ
ตัวแปรเดียว เก็บได้หลายค่า
`animals = ["แมว", "หมา", "นกฮูก", "หมูแดง"]`
2. The Zero Rule: กฎเหล็กที่เด็กสับสนที่สุด -> "คอมพิวเตอร์นับเลขเริ่มจากศูนย์"
- `print(animals[0])` -> ได้ แมว
- `print(animals[1])` -> ได้ หมา
3. Adding Items: สอนคำสั่ง `.append()`
`animals.append("ช้าง")` -> เอาช้างไปต่อท้ายแถว

ขั้นปฏิบัติ (Activity 1): "My Favorite Things"

- ให้นักเรียนสร้าง List เก็บสิ่งของที่ชอบ 3 หมวด (เช่น อาหาร, เพลง, สถานที่เที่ยว) หมวดละ 5 อย่าง
แล้วลองใช้คำสั่ง `print` ดึงข้อมูลตัวที่ 1, 3, 0 ออกมาโชว์

คาบที่ 3-4: นักแต่งนิทานมั่วซั่ว (Random Story Generator) - 100 นาที

ขั้นนำ (Instruction) :

- Hook: "จำเกมแต่งประโยคได้ไหม? ที่คนนึงเขียนประธาน คนนึงเขียนกริยา คนนึงเขียนกรรม แล้วเอามา
ต่อกันซ้ำๆ วันนี้เราจะให้ Python เล่นเกมนี้กับเรา"

ขั้นสอน (Instruction) :

1. Integrating Random: ทบทวน `import random`
2. Random Choice: แนะนำคำสั่งเทพ `random.choice(ชื่อลิสต์)` -> มันจะหยิบตัวจุ่มของในลิสต์
มาให้ 1 อย่าง

ขั้นปฏิบัติ (Activity 2): "The Chaos Story" (นิทานยาวใหญ่)

- โจทย์: ให้นักเรียนสร้าง List 3 ก่อ่ง

1. who (ใคร): ใส่ชื่อเพื่อน, ดารา, ตัวการ์ตูน
2. doing (ทำอะไร): ใส่กริยาแปลกๆ (เช่น เดินระบำ, ซี่ไม้กวาด, กินส้มตำ)
3. where (ที่ไหน): ใส่สถานที่ (เช่น บนดาวอังคาร, ในห้องน้ำ, บนหลังคาโรงเรียน)

- Coding: ให้โปรแกรมสุ่มเลือกจากทั้ง 3 ก่อ่ง แล้วเอามาต่อเป็นประโยค

```
import random
# ...สร้าง list 3 อัน...
a = random.choice(who)
b = random.choice(doing)
c = random.choice(when)
print(a + " กำลัง " + b + " อยู่ที่ " + c)
```

- Challenge: ใส่ loop while True ให้มันแต่งนิทานไปเรื่อยๆ จนกว่าจะสั่งหยุด

6. สื่อ/แหล่งการเรียนรู้

- แผนภาพแสดงช่องของ List เปรียบเทียบกับตู้ไปรษณีย์ที่มีหมายเลขกำกับ (0, 1, 2...)
- ตัวอย่างโปรแกรม Random Story Generator
- ใบงานที่ 10: "The Chaos Story" (นิทานยาวใหญ่)

7. การวัดและประเมินผล

จุดประสงค์การเรียนรู้	วิธีการวัดผล	เครื่องมือวัดผล	เกณฑ์การประเมิน
1. ด้านความรู้ (K) ระบุตำแหน่ง (Index) ของข้อมูลใน List ได้ถูกต้อง (เริ่มที่ 0)	ตรวจใบงานที่ 10 (Part 1: ถอดรหัสตำแหน่ง)	ใบงานที่ 10 "The Chaos Story"	ตอบถูกต้องร้อยละ 70 ขึ้นไป ถือว่า ผ่าน
2. ด้านทักษะ (P) เขียนโปรแกรมใช้ List ร่วมกับคำสั่งสุ่ม (random.choice) ได้	ตรวจผลงาน โปรแกรมแต่งนิทาน	แบบประเมิน ชิ้นงาน	ระดับคุณภาพพอใช้ขึ้นไป (เกณฑ์: สร้าง List ได้, ดึงข้อมูลมาแสดงผลได้ครบถ้วน)
3. ด้านคุณลักษณะ (A) เห็นประโยชน์ของการจัดเก็บข้อมูลเป็นหมวดหมู่	การตอบคำถาม ท้ายคาบเรียน	แบบสังเกต พฤติกรรม	นักเรียนอธิบายข้อดีของการใช้ List แทนตัวแปรเดี่ยวได้ ในระดับ ผ่าน

เกณฑ์การให้คะแนนชิ้นงาน (Rubric) - ใบงานคลังข้อมูลมหานุก (ใบงาน Chaos Story)

ดีมาก (3 คะแนน):

- สร้าง List ได้ถูกต้อง (ใช้ [] และคั่นด้วย ,)
- ใช้ random.choice() ดึงข้อมูลจาก List มาแต่งประโยคได้ใจความสมบูรณ์
- มีความเข้าใจเรื่อง Index (อ้างอิงตำแหน่งข้อมูลได้ถูกต้อง)

พอใช้ (2 คะแนน):

- สร้าง List และสุมข้อมูลได้
- แต่การต่อประโยคอาจยังไม่สมบูรณ์ หรือมีการอ้างอิง Index ผิดพลาดบ้าง

ปรับปรุง (1 คะแนน):

- ไม่ได้ใช้ List (ใช้ตัวแปรเดี่ยวๆ แทน)
- หรือสับสนวงเล็บ () กับ [] จนโปรแกรม Error:

ใบงานที่ 10: The Chaos Story

รายวิชา: วิทยาการคำนวณ ม.4 | Storyteller:

Part 1: ถอดรหัสตำแหน่ง (Index Decoding)

กำหนดให้ `bnk_members = ["Cherprang", "Pun", "Jennis", "Music", "Mobile"]`

- `bnk_members[0]` คือใคร?
- `bnk_members[3]` คือใคร?
- ถ้าอยากได้ "Mobile" ต้องใช้คำสั่งว่า

Part 2: ออกแบบคลังคำศัพท์ (Word Bank Design)

เตรียมข้อมูลสำหรับแต่งประโยคฮาเฮ

List 1: characters (ตัวละคร 5 ชื่อ) [".....", ".....", ".....", ".....", "....."]

List 2: actions (กริยา 5 อย่าง) [".....", ".....", ".....", ".....", "....."]

List 3: places (สถานที่ 5 แห่ง) [".....", ".....", ".....", ".....", "....."]

Part 3: เขียนโค้ดนักเล่านิทาน (Coding)

```
import random
```

```
# 1. สร้าง List ทั้ง 3 (ลอกข้อมูลจาก Part 2 มาใส่)
```

```
characters = [ .....
```

```
actions = [ .....
```

```
places = [ .....
```

```
# 2. สุ่มเลือกมาอย่างละ 1 ตัว
```

```
actor = random.choice(characters)
```

```
act = .....
```

```
place = .....
```

```
# 3. แสดงผลรวมร่าง
```

```
print("ชาวต่วน! วันนี้ " + actor + " แอบไป " + act + " ที่ " + place)
```

บันทึกท้ายแผนการจัดการเรียนรู้ที่ 10

1. IndexError: list index out of range

- อาการ: นักเรียนพยายามเรียก list[5] ทั้งที่ข้อมูลมีแค่ 3 ตัว
- วิธีแก้: สอนให้นับจำนวนข้อมูลจริง แล้วลบออก 1 จะได้ Index สูงสุด (เช่น มี 3 ตัว Index คือ 0, 1, 2)

2. Bracket Confusion:

- เด็กมักสับสนระหว่าง () (ใช้กับฟังก์ชัน), [] (ใช้กับ List), {} (เดี่ยวเจอใน Dict)
- ย้ำ: ถ้าจะสร้าง List หรือระบุตำแหน่ง ต้องใช้ ก้ามปูเหลี่ยม [] เท่านั้น

3. ภาษาไทยใน List:

- Python รองรับภาษาไทยใน List ได้ปกติ แต่ตอน print ต้องระวังเรื่องการตัดคำหรือสระลอย
ในบาง Console (แต่ใน Colab มักไม่มีปัญหา)

ตัวอย่างโปรแกรม Random Story Generator (Python)

```
import random

characters = [
    "นักเรียนมัธยม",
    "ครูหนุ่มปริศนา",
    "แม่มดใจดี",
    "หุ่นยนต์จากอนาคต",
    "นักสืบอัจฉริยะ"
]

places = [
    "ในโรงเรียนเก่า",
    "กลางป่าลึกลับ",
    "บนดาวอังคาร",
    "ในเมืองที่ไม่มีใครรู้จัก",
    "ภายในห้องทดลองลับ"
]

events = [
    "ค้นพบความลับบางอย่าง",
    "พบกับเหตุการณ์ไม่คาดฝัน",
    "ต้องตัดสินใจครั้งสำคัญ",
    "เผชิญกับความท้าทายใหญ่",
    "เปลี่ยนชะตาชีวิตไปตลอดกาล"
]

print("📖 Random Story Generator")
print("พิมพ์อะไรก็ได้เพื่อสร้างเรื่องใหม่")
print("พิมพ์ exit เพื่อออก\n")

while True:
    command = input("คุณ: ")
    if command == "exit":
        print("📖 ขอขอบคุณที่อ่านเรื่องราวของเรา")
        break
    else:
        story = (
            random.choice(characters) + " "
            + random.choice(events) + " "
            + random.choice(places)
        )
        print("📖 เรื่องของคุณ:")
        print(story)
        print("-" * 40)
```

แผนการจัดการเรียนรู้ที่ 11

กลุ่มสาระการเรียนรู้วิทยาศาสตร์และเทคโนโลยี (วิทยาการคำนวณ)

หน่วยการเรียนรู้ที่ 3: Python for Creativity

เรื่อง: Python บูรณาการ (Python in Action & Mini Project)

ระดับชั้น: มัธยมศึกษาปีที่ 4 (สายศิลป์-ภาษา/สังคม) เวลา: 4 คาบ (200 นาที)

1. มาตรฐานการเรียนรู้/ตัวชี้วัด

ว 4.2 ม.4/1: ประยุกต์ใช้แนวคิดเชิงคำนวณในการพัฒนาโครงงานที่มีการบูรณาการกับวิชาอื่นอย่างสร้างสรรค์ และเชื่อมโยงกับชีวิตจริง

2. สาระสำคัญ (Concept)

"Coding is a Tool for Everyone" การเขียนโปรแกรมไม่ใช่เป้าหมายสูงสุด แต่เป็น "เครื่องมือ" ที่ช่วยแก้ปัญหาในศาสตร์อื่นๆ ได้ เช่น นักภาษาศาสตร์ใช้ Python นับคำและวิเคราะห์วรรณกรรม, นักประวัติศาสตร์ใช้ Python สร้างแบบทดสอบความรู้, ศิลปินใช้ Python สร้างลวดลายกราฟิก ในคาบนี้เราจะลองสวมบทบาทอาชีพต่างๆ แล้วใช้ Python สร้างเครื่องมือช่วยงาน

3. จุดประสงค์การเรียนรู้ (K-P-A)

ด้านความรู้ (K) : วิเคราะห์ปัญหาจากวิชาอื่นและเลือกใช้คำสั่ง Python (List, Loop, Condition) ที่เหมาะสมในการแก้ปัญหาได้

ด้านทักษะกระบวนการ (P) : เขียนโปรแกรมประยุกต์ขนาดเล็ก (Mini Project) ได้ (เช่น โปรแกรมนับคำ, เกมตอบคำถาม)

ด้านเจตคติ (A) : มองเห็นแนวทางการนำวิทยาการคำนวณไปปรับใช้กับวิชาที่ตนเองถนัด

4. สาระการเรียนรู้

1. การจัดการข้อความขั้นสูง (String Methods):

- .split(): การตัดคำในประโยคให้กลายเป็น List (สำคัญมากสำหรับสายภาษา)
- len(): การนับจำนวนสมาชิก

2. การประยุกต์ใช้ List คู่กับ Loop:

- การสร้างเกมถาม-ตอบ โดยเก็บคำถามไว้ใน List หนึ่ง และคำตอบไว้อีก List หนึ่ง

5. กิจกรรมการเรียนรู้

คาบที่ 1-2: นักภาษาศาสตร์ดิจิทัล (The Word Counter) - 100 นาที

ขั้นนำ (Introduction):

1. Scenario: "ครูภาษาอังกฤษสั่งให้เขียน Essay 500 คำ... เราจะรู้ได้ไงว่าเขียนครบแล้ว?"
2. Solution: "ให้ Python ช่วยนับสิ! แป๊บเดียวเสร็จ"

ขั้นสอน (Instruction):

1. String Splitting:
 - ครูสอนคำสั่ง `.split()`
 - sentence = "I love Python"
 - words = sentence.split() -> ผลลัพธ์จะได้ List: ['I', 'love', 'Python']
2. Counting:
 - สอนคำสั่ง `len(words)` -> นับได้ 3 คำ จบข่าว!

ขั้นปฏิบัติ (Activity 1): "Essay Checker"

- ให้นักเรียนเขียนโปรแกรมรับข้อความ (Input) เป็นพารากราฟยาวๆ
- โปรแกรมจะบอกว่า:
 1. มีทั้งหมดกี่คำ? (Word Count)
 2. มีคำว่า "ฉัน" (หรือคำที่สนใจ) ปรากฏกี่ครั้ง? (ใช้ `.count()`)

คาบที่ 3-4: นักประวัติศาสตร์ผู้สร้างเกม (The History Quiz) - 100 นาที

ขั้นนำ (Introduction):

1. Challenge: "ใกล้สอบกลางภาควิชาสังคม/ประวัติศาสตร์แล้ว เรามาสร้างเกมช่วยติวหนังสือให้เพื่อนกันดีกว่า"

ขั้นสอน (Instruction):

1. Parallel Lists Concept: สอนเทคนิคการใช้ List 2 กลุ่มคู่ขนานกัน
 - questions = ["อยุธยาเสียกรุงครั้งที่ 2 ปี พ.ศ. อะไร?", "ใครแต่งเรื่องมัจฉะพารา?"]
 - answers = ["2310", "ร.6"]
 - Tip: ตำแหน่ง (Index) ที่ 0 ของทั้งสองกลุ่มต้องตรงกัน
2. Looping through Quiz:
 - ใช้ `for i in range(len(questions))`: เพื่อวนถามทีละข้อ

- รับคำตอบ -> เช็กกับเฉลย (answers[i]) -> นับคะแนน

ขั้นปฏิบัติ (Activity 2): "Quiz Master Integration"

1. ให้นักเรียนจับคู่ เลือกวิชาที่ชอบ (ไทย/สังคม/วิทย์พื้นฐาน)
2. สร้างเกมถาม-ตอบ 5 ข้อ
3. Extra: ต้องมีการสรุปคะแนนตอนจบว่าได้กี่คะแนน เต็ม 5

6. สื่อ/แหล่งการเรียนรู้

- ตัวอย่างโค้ด: Word Counter และ Quiz Game
- บทความสั้นๆ ภาษาอังกฤษและภาษาไทย (สำหรับทดสอบนับคำ)
- ใบงานที่ 11: "Python in Action"

7. การวัดและประเมินผล

จุดประสงค์การเรียนรู้	วิธีการวัดผล	เครื่องมือวัดผล	เกณฑ์การประเมิน
1. ด้านความรู้ (K) วิเคราะห์โจทย์ จากวิชาอื่นเพื่อนำมาแก้ปัญหา ด้วยคอมพิวเตอร์ได้	ตรวจการวางแผน ในใบงานที่ 11	ใบงานที่ 11 "Python in Action"	วิเคราะห์ Input/Process/Output ได้ สอดคล้องกับโจทย์ ในระดับ ผ่าน
2. ด้านทักษะ (P) เขียนโปรแกรม ประยุกต์ขนาดเล็ก (เช่น นับคำ/ Quiz) ที่ทำงานได้จริง	ตรวจผลงาน Mini Project	แบบประเมิน ชิ้นงาน	ระดับคุณภาพพอใช้ขึ้นไป(เกณฑ์: โปรแกรมทำงานได้ถูกต้องตาม วัตถุประสงค์ของวิชานั้นๆ)
3. ด้านคุณลักษณะ (A) เห็น แนวทางการนำความรู้ไป ประยุกต์ใช้ในสิ่งที่ตนถนัด	สังเกตความสนใจ และการเลือกหัวข้อ โครงการงาน	แบบสังเกต พฤติกรรม	นักเรียนเลือกหัวข้อที่เชื่อมโยงกับ ความสนใจหรือวิชาเอกของตนเอง ในระดับ ผ่าน

เกณฑ์การให้คะแนนชิ้นงาน (Rubric) - Python บูรณาการ (ชิ้นงาน Mini Project)

ดีมาก (3 คะแนน): เลือกใช้เครื่องมือ Python (List/Loop/String) ได้เหมาะสมกับการแก้ปัญหา, โปรแกรมทำงานได้ถูกต้องตามโจทย์ของวิชาที่เลือกบูรณาการ, โค้ดมีความเป็นระเบียบ มีการใส่ Comment อธิบาย

พอใช้ (2 คะแนน): โปรแกรมทำงานได้ แก้ปัญหาได้ระดับพื้นฐาน, โค้ดอาจจะซับซ้อนเกินความจำเป็น หรือยังไม่เป็นระเบียบ แต่ผลลัพธ์ถูกต้อง

ปรับปรุง (1 คะแนน): โปรแกรมทำงานไม่ตรงตามโจทย์ หรือแก้ปัญหาไม่ได้จริง, มี Error ที่ทำให้ โปรแกรมหยุดทำงานกลางคัน

ใบงานที่ 11: Python in Action

รายวิชา: วิทยาการคำนวณ ม.4 | Specialist:

Part 1: นักวิเคราะห์ข้อความ (Text Analyzer)

จงเขียนโปรแกรมเพื่อนับจำนวนคำในประโยคภาษาอังกฤษ

Code:

```
text = input("วางบทความภาษาอังกฤษของคุณที่นี่: ")
# 1. ตัดคำด้วยช่องว่าง (Split)
word_list = text.....
# 2. นับจำนวนคำ (Count)
total_words = .....(word_list)
print("บทความนี้มีทั้งหมด " + str(total_words) + " คำ")
# 3. (แถม) เช็คว่ามีคำว่า "the" กี่คำ
the_count = word_list.count("the")
print("พบคำว่า the ทั้งหมด " + str(the_count) + " ครั้ง")
```

Part 2: เกมแฟนพันธุ์แท้ (Subject Quiz)

เลือกวิชาที่ถนัดมา 1 วิชา แล้วสร้างชุดข้อสอบ 3 ข้อ วิชา:

เตรียมข้อมูล (Data Prep):

Q1: | A1:

Q2: | A2:

Q3: | A3:

เขียนโค้ด (Coding):

```
# สร้าง List เก็บคำถามและเฉลย
```

```
questions = [
```

```
    ".....",
```

```
    ".....",
```

```
    "....."
```

```
]
```

```
answers = [".....", ".....", "....."]
```

```
score = 0 # ตัวแปรเก็บคะแนน
```

```
# วนลูปถามทีละข้อ
```

```
for i in range(3):
```

```
    print("ข้อที่ " + str(i+1) + ": " + questions[i])
```

```
    user_ans = input("ตอบ: ")
```

```
    # ตรวจสอบคำตอบ (เทียบกับเฉลยในตำแหน่ง i)
```

```
    if user_ans == answers[i]:
```

```
        print("ถูกต้อง!")
```

```
        score = score + 1
```

```
    else:
```

```
        print("ผิด! คำตอบคือ " + answers[i])
```

```
print("คุณได้คะแนนรวม: " + str(score) + "/3")
```

บันทึกท้ายแผนการจัดการเรียนรู้ที่ 11

1. Split ภาษาไทย:

- คำสั่ง `.split()` โดยปกติจะตัดที่ "ช่องว่าง (Space)" ซึ่งภาษาอังกฤษใช้ได้ดี แต่ภาษาไทยเราเขียนติดกันเป็นพืด (เช่น "ฉันทินข้าว")

- วิธีแก้: บอกนักเรียนว่า ถ้าจะนับคำไทย ให้พิมพ์แบบเคาะวรรคตอน Input (เช่น "ฉันทิน ข้าว") หรือใช้บทความภาษาอังกฤษในการทดสอบจะง่ายกว่าสำหรับระดับเบื้องต้นครับ

2. Parallel Lists:

- การใช้ List แยก (คำถามกล่องนี้ คำตอบกล่องนี้) เป็นวิธี Basic ที่เข้าใจง่าย แต่อาจจะผิดพลาดถ้านักเรียนใส่จำนวนไม่เท่ากัน

- Note: ในระดับสูงเราจะใช้ Dictionary (`{'Q1': 'A1'}`) แต่สำหรับ ม.4 สายศิลป์ การใช้ List คู่ขนานแบบนี้ถือว่าเพียงพอและเห็นภาพชัดเจนครับ

3. Input Type:

- ระวังเรื่องคำตอบที่เป็นตัวเลข (เช่น พ.ศ. 2310) ถ้าใน List เก็บเป็น String "2310" แต่ตอนรับ input นักเรียนไปแปลงเป็น `int()` มันจะเทียบกันไม่เจอ ("2310" != 2310)

- แนะนำ: ให้เก็บเฉลยเป็น String ทั้งหมด และรับ Input เป็น String ทั้งหมด จะปัญหาน้อยที่สุดครับ

Python Code : Word Counter

ตัวอย่างที่ 1 : นับจำนวนคำในประโยค (พื้นฐาน)	
<pre>ตัวอย่างที่ 1 : นับจำนวนคำในประโยค (พื้นฐาน) text = "ฉันรักการเขียนโค้ดมาก" words = text.split() # แยกคำออกจากประโยค count = 0 for word in words: count += 1 print("จำนวนคำ =", count)</pre>	<p>อธิบายง่ายๆ</p> <p>split() = แยกประโยคเป็นคำ</p> <p>for = วนทีละคำ</p> <p>count = ตัวนับคำ</p>
ตัวอย่างที่ 2 : รับข้อความจากผู้ใช้	
<pre>text = input("พิมพ์ข้อความ: ") words = text.split() count = 0 for word in words: count += 1 print("มีทั้งหมด", count, "คำ")</pre>	<p>อธิบายง่ายๆ</p> <ul style="list-style-type: none"> - split() = แยกประโยคเป็นคำ - for = วนทีละคำ - count = ตัวนับคำ - ได้ฝึกใช้ input()
ตัวอย่างที่ 3 : นับ + แสดงแต่ละคำ	
<pre>text = "Python สนุก และ เข้าใจง่าย" words = text.split() for i in range(len(words)): print("คำที่", i + 1, "คือ", words[i])</pre>	<p>อธิบายง่ายๆ</p> <ul style="list-style-type: none"> - split() = แยกประโยคเป็นคำ - for = วนทีละคำตามจำนวน len ของคำ - ฝึกคิดลำดับ (index)

ตัวอย่างโค้ด Quiz Game เวอร์ชันพื้นฐาน (3 ข้อ)

<pre>score = 0 questions = [("2 + 2 = ?", "4"), ("เมืองหลวงของประเทศไทยคืออะไร?", "กรุงเทพ"), ("ภาษา Python ใช้เขียนโปรแกรมได้หรือไม่? (ใช่/ ไม่ใช่)", "ใช่")] for q, answer in questions: print(q) user = input("ตอบ: ") if user == answer: print("✔ ถูกต้อง") score += 1 else: print("✘ ผิด คำตอบคือ", answer) print("-----") print("คะแนนรวม =", score, "/", len(questions))</pre>	<p>เด็กจะได้เรียนรู้อะไรบ้าง</p> <p>for loop --> วนคำถามทีละข้อ</p> <p>if else --> ตรวจสอบคำตอบ</p> <p>list --> เก็บคำถาม</p> <p>score --> นับคะแนน</p>
---	---

ปัญหาหลักของ Word Counter ภาษาไทย คือ ภาษาไทยไม่มีช่องว่างคั่นคำ
หากต้องการให้รองรับภาษาไทยจริงต้องใช้ PyThaiNLP

```
pip install pythainlp
```

python code :

```
from pythainlp.tokenize import word_tokenize
```

```
text = input("พิมพ์ข้อความภาษาไทย: ")
```

```
words = word_tokenize(text)
```

```
count = 0
```

```
for word in words:
```

```
    if word.strip() != "":
```

```
        count += 1
```

```
print("จำนวนคำ =", count)
```

```
print("คำที่ตรวจพบ:", words)
```

บทความสั้นสำหรับใช้ในโปรแกรมนับคำ

บทความภาษาอังกฤษ (สำหรับใช้ในการเขียนโปรแกรมนับคำ)

Thai identity and national independence are the foundations of Thailand. Thai people should be grateful to the land and to their ancestors who sacrificed their lives to protect the nation's sovereignty. Thailand has remained independent through history because the monarchy has served as the spiritual center of the nation, safeguarding the land and guiding the country through many challenges. Remembering and honoring the contributions of the King and our ancestors is a responsibility of all Thai people. This shared awareness helps preserve independence, cultural heritage, and national identity for future generations.

บทความภาษาไทย (สำหรับใช้ในการเขียนโปรแกรมนับคำ)

ความเป็นไทยและความเป็นเอกราชคือรากฐานสำคัญของชาติไทย คนไทยควรสำนึกในบุญคุณของแผ่นดินที่บรรพชนได้เสียสละเลือดเนื้อและชีวิต เพื่อปกป้องรักษาเอกราชไม่ให้ตกเป็นของผู้อื่น ประเทศไทยดำรงอยู่ได้จนถึงปัจจุบัน เพราะมีพระมหากษัตริย์ทรงเป็นศูนย์รวมจิตใจของชาติ ทรงปกป้องรักษาแผ่นดินและนำพาประเทศผ่านวิกฤตการณ์การ ความสำนึกในคุณค่าของชาติ ศาสนา พระมหากษัตริย์ และบรรพชน คือหน้าที่ของคนไทยทุกคน เพื่อร่วมกันธำรงรักษาเอกราช วัฒนธรรม และอัตลักษณ์ไทยให้คงอยู่สืบไป

แผนการจัดการเรียนรู้ที่ 12

กลุ่มสาระการเรียนรู้วิทยาศาสตร์และเทคโนโลยี (วิทยาการคำนวณ)

หน่วยการเรียนรู้ที่ 3: Python for Creativity

เรื่อง: โครงการสร้างสรรค์ (Creative Capstone Project)

ระดับชั้น: มัธยมศึกษาปีที่ 4 (สายศิลป์-ภาษา/สังคม) เวลา: 6 คาบ (300 นาที)

1. มาตรฐานการเรียนรู้/ตัวชี้วัด

ว 4.2 ม.4/1: ประยุกต์ใช้แนวคิดเชิงคำนวณในการพัฒนาโครงการที่มีการบูรณาการกับวิชาอื่นอย่างสร้างสรรค์ และเชื่อมโยงกับชีวิตจริง

2. สาระสำคัญ (Concept)

"จากผู้เสพ สู่มือสร้าง (From Consumer to Creator)" ตลอดทั้งเทอมเราเรียนรู้เครื่องมือต่างๆ มาแล้วถึงเวลาที่จะนำมาประกอบร่างเพื่อแก้ปัญหาหรือสร้างความบันเทิงให้ผู้อื่น โครงการนี้ไม่ได้วัดว่าโค้ดใครซับซ้อนกว่ากัน แต่วัดที่ "ไอเดีย" และ "การเล่าเรื่อง" ว่าโปรแกรมของเราตอบโจทย์ผู้ใช้งานได้จริงหรือไม่ และมีความน่าสนใจเพียงใด

3. จุดประสงค์การเรียนรู้ (K-P-A)

ด้านความรู้ (K) : อธิบายกระบวนการพัฒนาโครงการคอมพิวเตอร์ (วางแผน -> พัฒนา -> ทดสอบ -> นำเสนอ) ได้

ด้านทักษะ (P) : พัฒนาโปรแกรมประยุกต์ด้วยภาษา Python ที่มีการใช้เงื่อนไข (Condition) และการวนซ้ำ (Loop) หรือ List เพื่อแก้ปัญหาหรือสร้างความบันเทิงได้

ด้านเจตคติ (A) : มีความภูมิใจในผลงานของตนเอง และยอมรับฟังคำติชมจากผู้อื่น

4. สาระการเรียนรู้

กระบวนการออกแบบเชิงวิศวกรรม (Engineering Design Process) ฉบับย่อ:

- Define: ระบุปัญหา/สิ่งที่อยากทำ
- Design: ออกแบบการทำงาน (Flowchart/Storyboard)
- Develop: เขียนโค้ด (Coding)
- Test: ทดสอบและแก้ไขบั๊ก (Debugging)
- Present: นำเสนอผลงาน

5. กิจกรรมการเรียนรู้ (แบ่งเป็น 3 สัปดาห์)

สัปดาห์ที่ 1: ระดมสมองและออกแบบ (Dream It & Map It) - 100 นาที

1. Team Up: จับกลุ่ม 2-3 คน (คละคนเก่งคอมกับคนเก่งครีเอทีฟ)
2. Brainstorming: แจกใบงาน Project Canvas ให้แต่ละกลุ่มเลือกธีม:
 - ธีม A: Interactive Fiction (นิยายทางเลือก/เกมจับหนุม/เกมสยองขวัญ)
 - ธีม B: Chatbot Assistant (บอทแนะนำหนังสือ/บอทแปลคำศัพท์สแลง/บอทเสียงเขียนซี)
 - ธีม C: Quiz Game Show (เกมตอบคำถามแฟนพันธุ์แท้)
3. Storyboard/Flowchart:
 - บังคับให้ "วาดก่อนเขียน" ห้ามแตะคอมพิวเตอร์จนกว่าจะวาดผังงานเสร็จ
 - ครูเดินตรวจ Logic ของแต่ละกลุ่มว่ามีความเป็นไปได้ไหม (Feasibility Check)

สัปดาห์ที่ 2: ลงมือสร้างและแก้บั๊ก (Hackathon Week) - 100 นาที

1. Coding Sprint: ปล่อยให้ นักเรียนลงมือเขียนโค้ดเต็มที่
2. Teacher as Consultant: ครูเปลี่ยนบทบาทเป็น "รุ่นพี่ Senior Dev" คอยเดินดูและช่วยแก้ Error
 - Tip: ถ้ากลุ่มไหนติดปัญหาใดๆ ให้ช่วยไต่หรือตัดพีเจอร์ที่ซับซ้อนเกินไปออก
3. Cross-Testing: ช่วง 20 นาทีสุดท้าย ให้แต่ละกลุ่มแลกคอมกันเล่นกับกลุ่มข้างๆ เพื่อหา "จุดผิด" (Bug Hunting) แล้วจดใส่กระดาษชิ้นเจ้าของ

สัปดาห์ที่ 3: เปิดตัวผลงาน (Demo Day) - 100 นาที

1. Pitching: แต่ละกลุ่มออกมานำเสนอ 5 นาที
 - แนะนำชื่อทีมและคอนเซปต์
 - สาธิตวิธีการเล่น (Live Demo)
 - บอกปัญหาที่เจอและวิธีแก้
2. Popular Vote: ให้เพื่อนในห้องโหวต "ขวัญใจมหาชน" (ห้ามโหวตกลุ่มตัวเอง)
3. Wrap Up: ครูสรุปภาพรวม ให้กำลังใจ และชี้ให้เห็นว่าพวกเขาก้าวข้ามขีดจำกัดของตัวเองมาไกลแค่ไหนจากคาบแรก

6. สื่อ/แหล่งการเรียนรู้

- ใบงานที่ 12: "The Project Canvas" (แบบร่างโครงการ)
- เว็บไซต์: รวมไอเดียโครงการคอมพิวเตอร์เบื้องต้น
- แบบประเมิน: แบบประเมินเพื่อน (Peer Review Form)

7. การวัดและประเมินผล

จุดประสงค์การเรียนรู้	วิธีการวัดผล	เครื่องมือวัดผล	เกณฑ์การประเมิน
1. ด้านความรู้ (K) อธิบายกระบวนการพัฒนาโครงการคอมพิวเตอร์ได้	การนำเสนอขั้นตอนการทำงานหน้าชั้นเรียน	แบบประเมินการนำเสนอ	อธิบายที่มา ขั้นตอน และวิธีแก้ปัญหาได้ชัดเจน ในระดับผ่าน
2. ด้านทักษะ (P) พัฒนาโปรแกรมภาษา Python เพื่อแก้ปัญหาหรือสร้างความบันเทิงได้	ตรวจชิ้นงานโปรแกรมโครงการ (Capstone Project)	แบบประเมินโครงการ (Rubric)	ระดับคุณภาพพอใช้ขึ้นไป (เกณฑ์: โปรแกรมรันได้, ใช้ความรู้ If/Loop/List ประกอบ, ไม่มี Error ร้ายแรง)
3. ด้านคุณลักษณะ (A) มีความรับผิดชอบ ทำงานเป็นทีม และยอมรับฟังความคิดเห็น	ประเมินจากการทำงานกลุ่มและการประเมินเพื่อน (Peer Review)	แบบประเมินเพื่อน / แบบสังเกต	สมาชิกกลุ่มให้ความร่วมมือดี และส่งงานตามกำหนดเวลา ในระดับ ผ่าน

เกณฑ์การให้คะแนนชิ้นงาน (Rubric) - โครงการสร้างสรรค์ (Capstone Project)

ดีมาก (3 คะแนน): ความสมบูรณ์: โปรแกรมรันได้สมบูรณ์ ไม่มี Error ร้ายแรง (Critical Bug), ความซับซ้อน: มีการประยุกต์ใช้ความรู้ (Variable, Condition, Loop, List) ครบถ้วน, ความคิดสร้างสรรค์: พล็อตเรื่องน่าสนใจ หรือมีฟีเจอร์ที่แปลกใหม่

พอใช้ (2 คะแนน): ความสมบูรณ์: โปรแกรมรันได้ แต่อาจมีบั๊กเล็กน้อย (Minor Bug), ความซับซ้อน: ใช้คำสั่งพื้นฐานเป็นหลัก (Print/Input/If), ความคิดสร้างสรรค์: เป็นไปตามมาตรฐานทั่วไป ไม่หวือหวา

ปรับปรุง (1 คะแนน): ความสมบูรณ์: โปรแกรมรันไม่จบ หรือ Error บ่อยครั้ง, ความซับซ้อน: โค้ดน้อยเกินไป หรือก๊อปปี้งานผู้อื่นมาโดยไม่ดัดแปลง

ใบงานที่ 12: The Project Canvas

รายวิชา: วิทยาการคำนวณ ม.4 | Team Name:

1. ไอเดียของเรา (The Big Idea)

ชื่อโปรเจกต์:

ประเภท: เกมนินยา แชนบอท เกมตอบคำถาม อื่นๆ

กลุ่มเป้าหมาย (ใครคือคนเล่น?):

เรื่องย่อ/ประโยชน์: (โปรแกรมนี้อะไรได้บ้าง? เล่าสั้นๆ)

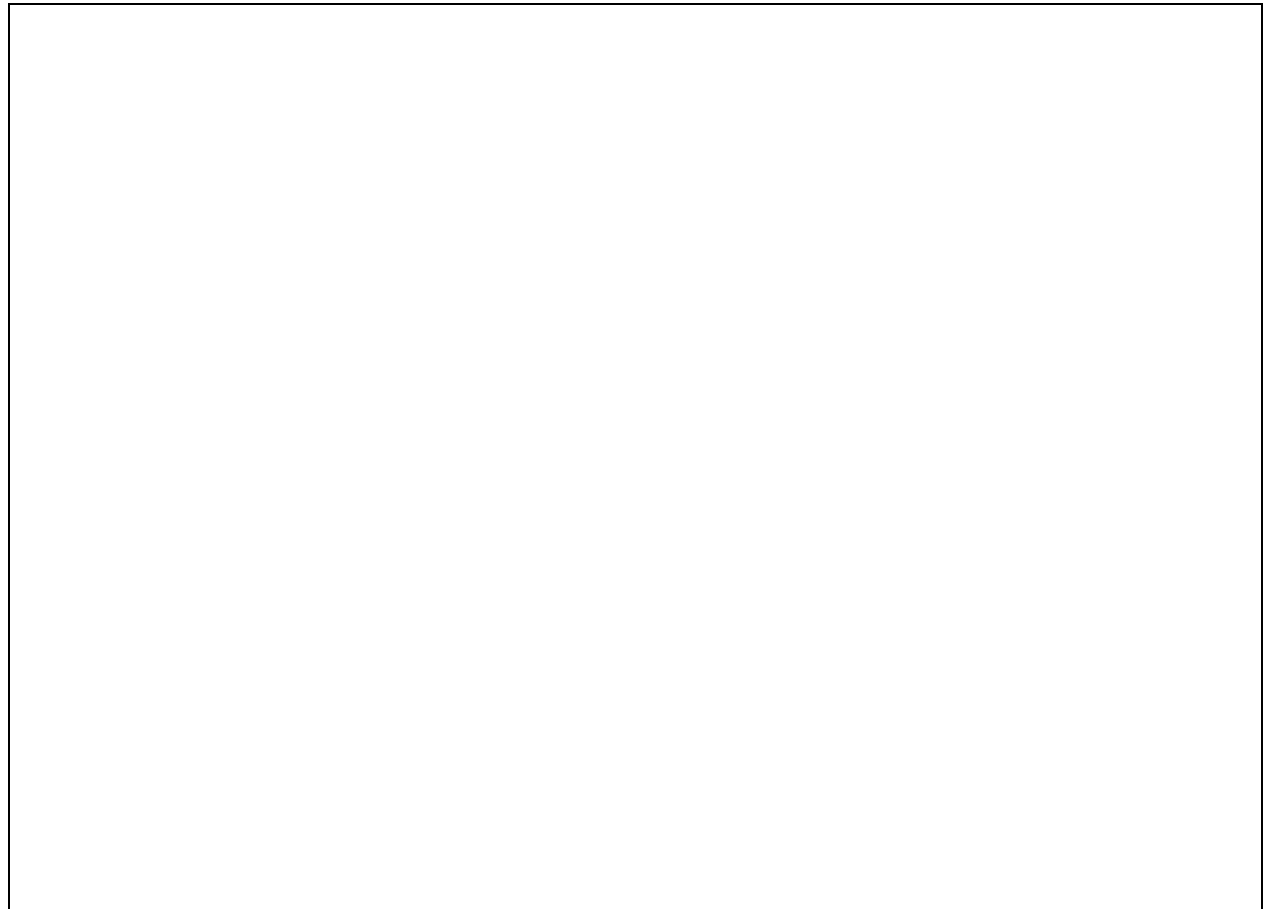
.....

.....

.....

.....

2. ออกแบบโครงสร้าง (Flowchart / Sketch) วาดแผนผังการทำงานคร่าวๆ (ถ้าเป็นเกม ให้วาดเส้นทางเลือก)



3. รายการตัวแปรที่ต้องใช้ (Variables Checklist)

ตัวแปรรับชื่อผู้เล่น:

ตัวแปรเก็บคะแนน/สถานะ:

ลิสต์ (List) สำหรับเก็บข้อมูล:

4. แบ่งงานกันทำ (Roles)

Coder (คนเขียนโค้ด):

Storyteller (คนแต่งเรื่อง/คำถาม):

Tester/Presenter (คนทดสอบ/นำเสนอ):

5. แผนการทำงาน (Action Plan)

สัปดาห์ที่ 1 (Plan & Write)

สัปดาห์ที่ 2 (Code & Dev)

สัปดาห์ที่ 3 (Test & Show)

แบ่งหน้าที่

เริ่มเขียนโค้ดใน Colab

สลับกลุ่มทดสอบ (Beta Test)

เขียนบท/เตรียมคำถาม

เชื่อมต่อบทเข้ากับโค้ด

แก้ไขบั๊ก (Fix Bugs)

ร่าง Flowchart ส่งครู

ตกแต่งหน้าจอให้สวยงาม]

ช้อมนำเสนอ (Pitching)

6. ความเห็นครูที่ปรึกษา (Mentor Feedback): อนุมัติ (Approved) แก้ไข (Revise)

เรื่อง.....

ลงชื่อครู

แบบประเมินเพื่อน (Peer Review Form)

(สำหรับใช้ในสัปดาห์ที่ 3)

ชื่อกลุ่มที่ประเมิน:

ความสนุก/น่าสนใจ: (1-5 ดาว) ★★★★★

ความสับสนของโปรแกรม: (เจอ Error ไหม?)

สิ่งที่ชอบที่สุดในงานเพื่อน:

คำแนะนำเพิ่มเติม:

บันทึกท้ายแผนการจัดการเรียนรู้ที่ 12

1. จัดการความคาดหวัง: ย้ำกับนักเรียนว่า "Done is better than perfect" (งานเสร็จดีกว่างานเพอร์เฟกต์แต่ไม่เสร็จ) โปรแกรมไม่ต้องใหญ่โต ขอแค่ทำงานได้ตามที่คุยไว้ก็พอ

2. การช่วยเหลือ: เตรียม "Code Snippet" หรือ "โค้ดสำเร็จรูป" ไว้แจกสำหรับฟังก์ชันยากๆ เช่น การเคลียร์หน้าจอ, การใส่ Delay เพื่อให้ตัวหนังสือค่อยๆ ขึ้น (เพิ่มลูกเล่นให้สายศิลป์ชอบ)

```
import time
print("กำลังโหลด...", end="", flush=True)
time.sleep(2) # หยุดรอ 2 วินาที
print("เสร็จแล้ว!")
```

3. บรรยากาศ Demo Day: ถ้าเป็นไปได้ ให้จัดบรรยากาศแบบสบายๆ เปิดเพลงคลอ หรือมีรางวัลเล็กๆ น้อยๆ (ขนม) ให้กับทีมที่ชนะโหวต จะช่วยจบบริชาด้วยความประทับใจครับ