

การใช้กิจกรรมการเรียนรู้แบบนักสืบโค้ด เพื่อส่งเสริมการเรียนรู้เท่าทัน AI และทักษะการตรวจสอบความถูกต้องของภาษาไพธอน

เกษรินทร์ ชาวเกวียน^{1*}, จุฑาศินี พรพุทธศรี¹, ชูเกียรติ โชติกเสถียร¹ ขจรศักดิ์ สีมะข¹,
และ เอกนรา จันดา²

¹ครุศาสตร์อุตสาหกรรม คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏนครปฐม

²วิศวกรรมโยธา คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏนครปฐม

*ketsarin@webmail.npru.ac.th

บทคัดย่อ

ปัจจุบัน การนำปัญญาประดิษฐ์แบบสร้างสรรค์ (Generative AI) มาใช้สนับสนุนการเขียนโปรแกรมภาษาคอมพิวเตอร์ เป็นไปอย่างแพร่หลาย อย่างไรก็ตาม ปัญหาสำคัญที่พบคือปรากฏการณ์ภาวะหลอนของ AI (AI Hallucination) ที่มักสร้างข้อผิดพลาดซึ่งดูน่าเชื่อถือแต่แฝงไปด้วยข้อผิดพลาดที่ไม่มีอยู่จริง หากผู้เรียนพึ่งพาเทคโนโลยีดังกล่าวโดยขาดวิจารณญาณจะนำไปสู่ความผิดพลาดเชิงตรรกะที่ร้ายแรงและการถดถอยของทักษะการแก้ปัญหาด้วยตนเอง การวิจัยนี้ จึงมีวัตถุประสงค์เพื่อพัฒนากิจกรรมการเรียนรู้แบบนักสืบโค้ด (Code Detective) ที่ประยุกต์ใช้กระบวนการแกะรอยโค้ดเพื่อรับมือกับภาวะหลอนของปัญญาประดิษฐ์ ส่งเสริมสมรรถนะการเรียนรู้เท่าทัน AI และทักษะการทวนสอบความถูกต้องของภาษา และ ศึกษาความสามารถในการตรวจจับข้อผิดพลาดผ่านโจทย์ลวง (Trap Tasks) กลุ่มตัวอย่าง คือ นักศึกษาชั้นปีที่ 1 จำนวน 35 คน ได้มาโดยการเลือกแบบเจาะจง เครื่องมือที่ใช้ประกอบด้วย สไลด์เนื้อหา, ใบงานกิจกรรมโจทย์ลวง 5 ภารกิจหลัก, แบบทดสอบวัดสมรรถนะ (Pre-test/Post-test) และเกณฑ์การประเมินสมรรถนะ วิเคราะห์ข้อมูลด้วยสถิติค่าเฉลี่ย ($\bar{X}$) ส่วนเบี่ยงเบนมาตรฐาน (S.D.) และการทดสอบค่าทีแบบกลุ่มตัวอย่างไม่เป็นอิสระต่อกัน (Paired Samples t-test) ผลการวิจัยพบว่า การใช้กิจกรรมการเรียนรู้แบบนักสืบโค้ดช่วยให้ผู้เรียนมีทักษะการทวนสอบความถูกต้องของภาษาไพธอนเพิ่มขึ้นอย่างมีนัยสำคัญ ผ่านกระบวนการวิเคราะห์โค้ดทีละบรรทัด นักศึกษาสามารถระบุตำแหน่งและประเภทของข้อผิดพลาดในโจทย์ลวงได้แม่นยำขึ้น โดยค่าเฉลี่ยก่อนเรียน $\bar{X} = 22.31$, หลังเรียน $\bar{X} = 29.77$, $t = -7.114$, $p < .001$, $r = 0.773$ และมีความพึงพอใจต่อรูปแบบการจัดการเรียนรู้ในระดับ มาก

คำสำคัญ: นักสืบโค้ด การแกะรอยโค้ด การทวนสอบความถูกต้อง ภาวะหลอนของ AI การรู้เท่าทัน AI ภาษาไพธอน

Using Code Detective Learning Activities to Promote AI Literacy and Code Verification Skills in Python Language

Ketsarin Chawgien^{1*}, Jutasinee pornputthasri¹, Chukiat Chotikasatian¹,
Khachonsak Simek¹, and Eknara Junda²

¹Industrial Education, Faculty of Science and Technology, Nakhon Pathom Rajabhat University

²Civil Engineering, Faculty of Science and Technology, Nakhon Pathom Rajabhat University

*ketsarin@webmail.npru.ac.th

Abstract

At present, the use of Generative AI to support computer programming has become widespread. However, a significant challenge is AI Hallucination, in which AI generates source code that appears credible but contains non-existent methods or logical errors. If learners rely on this technology without critical judgment, it can lead to severe logical fallacies and a regression in self-directed problem-solving skills. This research aims to develop Code Detective learning activities that apply code-tracing processes to combat AI hallucinations and promote AI literacy and code-verification competencies, and to study the ability to detect errors through Trap Tasks. The sample group consisted of 35 students selected through purposive sampling. The research instruments included instructional slides on AI Hallucination, a worksheet featuring five Trap Task missions, pre- and post-tests, and competency evaluation rubrics. Data were analyzed using Mean ($\bar{X}$), Standard Deviation (S.D.), and Paired Samples t-test. Through a line-by-line analysis process, students were able to identify the location and type of errors in the Trap Tasks more accurately. The mean scores before learning were 22.31, after learning were 29.77, $t = -7.114$, $p < .001$, $r = 0.773$, and satisfaction with the learning management model was at a high level.

Keywords: Code Detective, Code Tracing, Code Verification, AI Hallucination, AI Literacy, Python Language

1. บทนำ

ในยุคปัจจุบันที่เทคโนโลยีดิจิทัลก้าวเข้าสู่ยุคปัญญาประดิษฐ์ (Artificial Intelligence) อย่างเต็มตัว การเรียนรู้ทักษะการเขียนโปรแกรมคอมพิวเตอร์ โดยเฉพาะภาษา Python ได้กลายเป็นทักษะพื้นฐานที่มีความสำคัญยิ่งในทุกสาขาวิชา เนื่องจากเป็นภาษาที่เป็นรากฐานของการพัฒนาเทคโนโลยีสมัยใหม่และการวิเคราะห์ข้อมูลขนาดใหญ่ อย่างไรก็ตาม การอุบัติขึ้นของปัญญาประดิษฐ์แบบสร้างสรรค์ (Generative AI) เช่น ChatGPT และ Gemini ได้ส่งผลให้วิถีการเรียนรู้ของผู้เรียนเปลี่ยนแปลงไปอย่างก้าวกระโดด จากเดิมที่เน้นการเขียนรหัสคำสั่งด้วยตนเองไปสู่การพึ่งพา AI ในการสร้างรหัสคำสั่ง (Code Generation) มากยิ่งขึ้น

ปรากฏการณ์ดังกล่าวได้นำมาซึ่งปัญหาสำคัญที่เรียกว่า “ภาวะหลอนของปัญญาประดิษฐ์” (AI Hallucination) ซึ่ง AI มักจะสร้างรหัสคำสั่งที่ดูมีความน่าเชื่อถือและถูกต้องตามหลักไวยากรณ์ แต่กลับแฝงไปด้วยข้อผิดพลาดเชิงตรรกะ หรือการอ้างอิงฟังก์ชันที่ไม่มีอยู่จริง หากผู้เรียนขาดสมรรถนะในการวิเคราะห์และตรวจสอบ จะส่งผลให้เกิดการเรียนรู้ที่ผิดพลาดและขัดขวางการพัฒนาทักษะการแก้ปัญหา (Problem Solving) ที่แท้จริงในระยะยาว ซึ่งการจัดการเรียนรู้ในรูปแบบบรรยายปกติในปัจจุบันเริ่มไม่เพียงพอต่อการสร้างทักษะการรู้เท่าทัน AI ในสภาวะการณ์ที่เทคโนโลยีเปลี่ยนแปลงไปอย่างรวดเร็ว เช่นนี้

งานวิจัยนี้ เล็งเห็นความสำคัญในการพัฒนากระบวนการเรียนรู้ที่เน้นการปฏิบัติจริง โดยได้นำแนวคิดการจัดการเรียนรู้เชิงรุก (Active Learning) โดย มาบูรณาการร่วมกับการจัดการเรียนรู้แบบสาธิต (Demonstration Method) เพื่อพัฒนาเป็นชุดการเรียนรู้ที่เน้นกระบวนการสังเกตความผิดพลาดเชิงประจักษ์และการแก้ไขสถานการณ์ผ่านนวัตกรรม โจทย์ลวง (Trap Tasks) โดยมีวัตถุประสงค์หลักเพื่อพัฒนากิจกรรมการเรียนรู้ที่มีประสิทธิภาพและหาคุณภาพตามเกณฑ์ 80/80 พร้อมทั้งเปรียบเทียบผลสัมฤทธิ์ทางการเรียนก่อนและหลังเรียน รวมถึงศึกษาความพึงพอใจของผู้เรียนที่มีต่อกระบวนการจัดการเรียนรู้ดังกล่าว การวิจัยในครั้งนี้ มุ่งเน้นไปที่การเขียนโปรแกรม Python เบื้องต้นและการตรวจจับข้อผิดพลาด 5 รูปแบบสำคัญ ได้แก่ ด้าน Library, Logic, Syntax, Method และ Algorithm ภายใต้รูปแบบการจัดการเรียนรู้แบบเร่งรัด (Blitz Model) ภายใน 180 นาที โดยมุ่งหวังว่าประโยชน์ที่ได้รับจากการวิจัยจะช่วยให้ผู้เรียนมีสมรรถนะในการตรวจสอบและแก้ไขรหัสคำสั่งที่ผิดพลาดได้อย่างถูกต้อง และสร้างต้นแบบกิจกรรมการเรียนรู้ที่ครูผู้สอนสามารถนำไปประยุกต์ใช้เพื่อพัฒนาทักษะการคิดขั้นสูงของผู้เรียน สอดคล้องกับการพัฒนาสมรรถนะพลเมืองดิจิทัลในศตวรรษที่ 21 อย่างเป็นรูปธรรม

2 เอกสารและงานวิจัยที่เกี่ยวข้อง

งานวิจัยนี้ประยุกต์ใช้รูปแบบการจัดการเรียนรู้แบบเร่งรัด (Blitz Model) ซึ่งเป็นแนวทางการฝึกอบรมระยะสั้นที่มีความเข้มข้นสูง มุ่งเน้นการพัฒนาสมรรถนะเฉพาะทางร่วมกับกิจกรรม "นักสืบโค้ด" เพื่อกระตุ้นให้ผู้เรียนเกิดการรู้เท่าทัน AI ผ่านกระบวนการแกะรอยโค้ด (Code Tracing) ในการสืบเสาะและทวนสอบความถูกต้อง (Code Verification) จากโจทย์ลวง (Trap Tasks) ที่แฝงปรากฏการณ์ภาวะหลอนของ AI (AI Hallucination) โดยกระบวนการเรียนรู้ดังกล่าวถูกออกแบบให้สอดคล้องกับกรอบแนวคิดอนุกรมวิธานของบลูมฉบับปรับปรุง (Revised Bloom's Taxonomy) เพื่อยกระดับทักษะการคิดขั้นสูงของผู้เรียน จากชั้นความเข้าใจสู่ขั้นการวิเคราะห์ การประเมินค่า และการสร้างสรรค์โค้ดใหม่ที่มีความถูกต้องเชิงตรรกะและโครงสร้างภาษาไพธอนได้อย่างมีประสิทธิภาพ

2.1 แนวคิดเกี่ยวกับการจัดการเรียนรู้เชิงรุก (Active Learning)

ในการจัดการศึกษาที่มุ่งเน้นผลสัมฤทธิ์และสมรรถนะของผู้เรียนเป็นสำคัญ การจัดการเรียนรู้เชิงรุกได้เข้ามามีบทบาทอย่างมากในการพลิกโฉมห้องเรียนแบบดั้งเดิม โดยได้แบ่งประเด็นสำคัญของแนวคิดนี้ออกเป็น 3 ส่วนหลัก ดังนี้

2.1.1 ความหมายและหลักการพื้นฐานของการเรียนรู้เชิงรุก

การจัดการเรียนรู้เชิงรุก (Active Learning) เป็นกระบวนการจัดการเรียนรู้ที่มุ่งเน้นให้ผู้เรียนมีส่วนร่วมและมีปฏิสัมพันธ์กับกิจกรรมการเรียนรู้ผ่านการลงมือปฏิบัติที่หลากหลาย แนวคิดนี้วางรากฐานอยู่บนความเชื่อที่ว่า การเรียนรู้ที่มีประสิทธิภาพสูงสุดเกิดจากการกระทำและการสะท้อนคิด (Doing and Reflecting) โดยหลักการสำคัญคือการเปลี่ยนผ่านบทบาทของผู้เรียนจากการเป็นเพียง ผู้รับข้อมูล (Passive Learner) ที่รอคอยการบรรยายจากผู้สอน มาเป็น ผู้แสวงหาความรู้ และสร้างองค์ความรู้ด้วยตนเอง (Active Learner) กระบวนการนี้เปิดโอกาสให้ผู้เรียนได้คิด วิเคราะห์ แลกเปลี่ยนความคิดเห็น และบูรณาการความรู้ใหม่เข้ากับประสบการณ์เดิม ซึ่งสอดคล้องกับแนวคิดของ Prince [1] ที่ระบุว่าองค์ประกอบสำคัญของการเรียนรู้เชิงรุกคือการดึงผู้เรียนเข้ามามีส่วนร่วมในกระบวนการเรียนรู้ที่มีความหมาย

2.1.2 ความสำคัญของการเรียนรู้เชิงรุกในบริบทคอมพิวเตอร์ศึกษา

ในบริบทของการศึกษาวิชาคอมพิวเตอร์ โดยเฉพาะอย่างยิ่งในการเรียนการสอนวิชาการเขียนโปรแกรม (Programming) การจัดการเรียนรู้เชิงรุกมีความสำคัญอย่างยิ่งยวด เนื่องจากธรรมชาติของวิชาคอมพิวเตอร์เป็นวิชาทักษะปฏิบัติที่ทักษะการเขียนโปรแกรมไม่สามารถเกิดขึ้นได้จากการท่องจำทฤษฎีหรือการจำไวยากรณ์ (Syntax) เพียงอย่างเดียว ผู้เรียนจำเป็นต้องอาศัยการลงมือปฏิบัติจริง การเผชิญหน้ากับข้อผิดพลาด (Error) และการค้นหาวិธีแก้ไข การให้ผู้เรียนได้โต้ตอบกับระบบคอมพิวเตอร์และระบบปัญญาประดิษฐ์ด้วยตนเองผ่านกิจกรรมเชิงรุก จะช่วยบ่มเพาะทักษะการคิดวิเคราะห์เชิงตรรกะและทักษะการแก้ปัญหา (Problem Solving) ประเด็นนี้ได้รับการยืนยันจากงานวิจัยของ Freeman et al. [2] ที่พบว่าการจัดการเรียนรู้เชิงรุกในกลุ่มวิชา STEM (รวมถึงวิทยาการคอมพิวเตอร์) สามารถเพิ่มผลสัมฤทธิ์ทางการเรียนและลดอัตราการสอบตกของผู้เรียนได้อย่างมีนัยสำคัญเมื่อเทียบกับการสอนแบบบรรยาย

2.1.3 การพัฒนากระบวนการคิดขั้นสูงผ่านสถานการณ์ปัญหา

จุดเด่นที่สำคัญที่สุดประการหนึ่งของการเรียนรู้เชิงรุก คือการออกแบบกิจกรรมที่เปิดโอกาสให้ผู้เรียนได้เผชิญกับสถานการณ์ปัญหาจริง (Real-world Problems) หรือสถานการณ์จำลองที่ท้าทายความคิด เมื่อผู้เรียนถูกกระตุ้นด้วยปัญหาที่ไม่มีคำตอบสำเร็จรูป เช่น การต้องตรวจสอบรหัสคำสั่งที่ผิดปกติจากภาวะหลอนของปัญญาประดิษฐ์ จะเป็นการกระตุ้นให้ผู้เรียนเกิด กระบวนการคิดขั้นสูง (Higher-Order Thinking) ตามกรอบแนวคิดอนุกรมวิธานของบลูม (Revised Bloom's Taxonomy) ฉบับปรับปรุง [3] ไม่ว่าจะเป็นการวิเคราะห์ (Analyzing) การประเมินค่า (Evaluating) และการสร้างสรรค์ (Creating) นอกจากนี้ กระบวนการที่ผู้เรียนต้องใช้ความพยายามในการคิดวิเคราะห์และแก้ปัญหาด้วยตนเองนี้ จะช่วยให้เกิดการประมวลผลข้อมูลอย่างลึกซึ้ง ส่งผลให้ผู้เรียนสามารถจดจำกระบวนการแก้ปัญหาและนำไปเก็บไว้ในความจำระยะยาว (Long-term Memory) ได้อย่างมีประสิทธิภาพมากกว่าการรับฟังคำเฉลยจากผู้สอน

2.1.4 รูปแบบการจัดการเรียนรู้แบบเร้งรัด (Blitz Model)

เป็นแนวทางการจัดการเรียนรู้เชิงรุกที่มีลักษณะสั้น กระชับ แต่มีความเข้มข้นสูง (Short-duration, High-intensity Learning) มุ่งเน้นการพัฒนาสมรรถนะหรือทักษะเฉพาะด้าน (Targeted Skills) ให้เกิดขึ้นอย่างรวดเร็ว โดยกระบวนการเรียนรู้จะลดทอนการบรรยายภาคทฤษฎีที่ไม่จำเป็น แล้วเปลี่ยนเป็นการเปิดโอกาสให้ผู้เรียนได้เผชิญหน้าและลงมือแก้ปัญหา กับสถานการณ์จริงในทันที ซึ่งในบริบทของวิทยาการคอมพิวเตอร์และการเรียนรู้ยุคใหม่ รูปแบบนี้มีถูกนำมาประยุกต์ใช้เพื่อหลายข้อจำกัดด้านเวลา และกระตุ้นกระบวนการคิดเชิงวิเคราะห์เพื่อหาข้อผิดพลาดทางเทคนิค [4] โครงสร้างสำคัญของแนวคิดนี้จะเน้นการจัดการเรียนรู้แบบมุ่งเป้า (Focused Input) ร่วมกับการปฏิบัติงานภายใต้เงื่อนไขเวลาที่จำกัด (High-Intensity Practice) ซึ่งผลการวิจัยในอดีตยืนยันว่าการบีบอัดเวลาและป้อนกิจกรรมที่ท้าทายในลักษณะนี้ สามารถรักษาความสนใจของผู้เรียนให้อยู่ในระดับสูง และเร้งรัดให้เกิดทักษะการแก้ปัญหาขั้นสูงได้อย่างมีประสิทธิภาพ [1]

2.2 แนวคิดเกี่ยวกับการจัดการเรียนรู้แบบสาธิต (Demonstration Method)

การจัดการเรียนรู้แบบสาธิตเป็นยุทธวิธีทางการศึกษาที่ช่วยเปลี่ยนนามธรรมให้เป็นรูปธรรม ซึ่งมีความสำคัญอย่างยิ่งในการเรียนการสอนที่ต้องอาศัยทักษะการปฏิบัติ โดยได้แบ่งประเด็นสำคัญของการจัดการเรียนรู้แบบสาธิตออกเป็น 3 ประเด็น ดังนี้

2.2.1 ความหมายและทฤษฎีพื้นฐานของการจัดการเรียนรู้แบบสาธิต

การจัดการเรียนรู้แบบสาธิตเป็นวิธีสอนที่ผู้สอนหรือผู้เชี่ยวชาญแสดงกระบวนการทำงาน การใช้เครื่องมือ หรือการแก้ปัญหาให้ผู้เรียนดูเป็นตัวอย่างเชิงประจักษ์ [5] แนวคิดนี้มีความสอดคล้องอย่างลึกซึ้งกับทฤษฎีการเรียนรู้ทางสังคม (Social Learning Theory) ของ Bandura [6] ที่เชื่อว่ามนุษย์สามารถเรียนรู้พฤติกรรมและทักษะใหม่ ๆ ได้ดีที่สุดผ่านการสังเกตตัวแบบ (Observational Learning) การที่ผู้เรียนได้เห็นขั้นตอนการปฏิบัติจริงตั้งแต่ต้นจนจบ จะช่วยสร้างมโนภาพที่ชัดเจนใน

ระบบสมอง และนำไปสู่การเลียนแบบหรือประยุกต์ใช้พฤติกรรมเหล่านั้นได้อย่างมีประสิทธิภาพมากกว่าการอ่านจากตำราเพียงอย่างเดียว

2.2.2 การถ่ายทอดกระบวนการคิดผ่านตัวแบบ (Cognitive Modeling)

ในบริบทของการสอนเขียนโปรแกรมคอมพิวเตอร์ร่วมกับการใช้ปัญญาประดิษฐ์ การสาธิตไม่ได้จำกัดอยู่เพียงการแสดงผลลัพธ์หรือรหัสคำสั่งที่ถูกต้องเท่านั้น แต่มีความจำเป็นอย่างมากในการถ่ายทอด กระบวนการคิด (Cognitive Modeling) ซึ่งเป็นหลักการสำคัญของแนวทางการเรียนรู้แบบผู้ฝึกงานทางปัญญา (Cognitive Apprenticeship) [7] ผู้สอนจะต้องทำให้กระบวนการคิดที่ซ่อนอยู่ภายในปรากฏออกมาเป็นรูปธรรมให้ผู้เรียนเห็น ไม่เพียงแต่แสดงให้เห็นว่ารหัสคำสั่งใดถูกต้อง แต่ยังต้องสาธิตวิธีการตั้งคำถาม (Prompting) การสังเกตข้อผิดพลาด และกระบวนการคิดวิเคราะห์เพื่อแก้ปัญหา เพื่อให้ผู้เรียนเข้าใจกระบวนการเบื้องหลังความสำเร็จนั้น

2.2.3 การสาธิตสด (Live Demonstration) เพื่อรับมือกับภาวะหลอนของปัญญาประดิษฐ์

เมื่อนำการสาธิตมาประยุกต์ใช้ในยุคปัญญาประดิษฐ์ ผู้สอนจำเป็นต้องเผชิญหน้ากับสถานการณ์ที่ระบบอาจแสดงผลลัพธ์ที่ผิดปกติหรือเกิดภาวะหลอน การใช้เทคนิคการสาธิตสด (Live Demonstration หรือ Live Coding) จะทำหน้าที่เป็นสะพานเชื่อมระหว่างทฤษฎีและการปฏิบัติจริง [8] ผู้สอนสามารถแสดงให้เห็นถึงกระบวนการสืบเสาะหาความจริง (Verification) เมื่อได้รับข้อมูลที่ผิดพลาดจาก AI การสาธิตให้เห็นถึงวิธีการรับมือกับข้อผิดพลาด (Error) อย่างเป็นขั้นตอน จะช่วยลดความสับสน ชัดความตื่นตระหนก และสร้างความมั่นใจให้กับผู้เรียน ทำให้ผู้เรียนพร้อมที่จะลงมือปฏิบัติและตรวจสอบข้อผิดพลาดด้วยตนเองในขั้นต่อไป

2.3 แนวคิดเกี่ยวกับภาวะหลอนของปัญญาประดิษฐ์ (AI Hallucination)

ในยุคที่แบบจำลองภาษาขนาดใหญ่ (Large Language Models: LLMs) เข้ามามีบทบาทสำคัญและพลิกโฉมวงการเทคโนโลยี ปรากฏการณ์ที่เรียกว่า ภาวะหลอนของปัญญาประดิษฐ์ (AI Hallucination) ได้กลายเป็นความท้าทายหลักทางการศึกษา โดยเฉพาะอย่างยิ่งในสาขาวิทยาการคอมพิวเตอร์ จึงได้แบ่งประเด็นการศึกษาที่เกี่ยวข้องกับภาวะหลอนของปัญญาประดิษฐ์ออกเป็น 3 ประเด็นหลัก ดังนี้

2.3.1 ความหมายและลักษณะของภาวะหลอนของปัญญาประดิษฐ์

ภาวะหลอนของปัญญาประดิษฐ์ (AI Hallucination) หมายถึง ปรากฏการณ์ที่แบบจำลองภาษาขนาดใหญ่สร้างคำตอบที่ไม่มีความถูกต้องตามข้อเท็จจริง หรือสร้างข้อมูลที่ไม่มีความเกี่ยวข้องกับบริบทที่ได้รับ แต่กลับนำเสนอข้อมูลเหล่านั้นด้วยรูปแบบที่มีความน่าเชื่อถือและมีความถูกต้องตามหลักไวยากรณ์ทางภาษาอย่างแนบเนียน [9] ในบริบทของการเขียนโปรแกรมคอมพิวเตอร์ ภาวะหลอนมักปรากฏให้เห็นในรูปแบบของการสร้างชื่อฟังก์ชันหรือโมดูลที่ไม่มีอยู่จริง (Method Hallucination) หรือการแนะนำชุดคำสั่งที่ดูเหมือนจะสามารถทำงานได้ตามปกติ แต่แท้จริงแล้วกลับแฝงไปด้วยข้อผิดพลาดเชิงตรรกะ ซึ่งเป็นอันตรายอย่างยิ่งต่อผู้ใช้งานที่ขาดความเชี่ยวชาญในการตรวจสอบ

2.3.2 สาเหตุและปัจจัยที่ก่อให้เกิดภาวะหลอนในการเขียนโปรแกรม

จากการศึกษาพบว่า ปัจจัยที่ก่อให้เกิดภาวะหลอนมีรากฐานมาจากข้อจำกัดทางสถาปัตยกรรมของตัวแบบจำลองเอง โดยสาเหตุหลักประการแรกมาจาก กระบวนการคาดการณ์เชิงสถิติ (Probabilistic Nature) ซึ่ง AI ทำงานโดยการคาดเดาคำถัดไปที่มีความน่าจะเป็นสูงสุดตามชุดข้อมูลฝึกฝน (Training Data) ลักษณะการทำงานแบบนี้ทำให้บางครั้งระบบพยายามเติมคำให้เต็มประโยค หรือสร้างรูปแบบรหัสคำสั่งให้สมบูรณ์จนกลายเป็นการผลิตข้อมูลเท็จขึ้นมา [10] ประการที่สองคือปัญหา ข้อมูลที่ล้าสมัย (Data Cutoff) โดยปัญญาประดิษฐ์อาจจดจำรูปแบบไลบรารีหรือคำสั่งเก่าที่ถูกยกเลิกการใช้งานไปแล้ว (Deprecated) เมื่อถูกผู้ใช้อ้อนคำถาม จึงนำเสนอข้อมูลที่ไม่สามารถใช้งานได้ในสภาพแวดล้อมปัจจุบัน และประการสุดท้ายคือ การขาดความเข้าใจเชิงบริบทอย่างแท้จริง แม้ AI จะมีความสามารถในการจัดเรียงรหัสคำสั่ง แต่ก็ยังขาด

ความเข้าใจในเชิงตรรกะทางคณิตศาสตร์ที่ซับซ้อน ทำให้เกิดข้อผิดพลาดในการคำนวณหรือการจัดการเงื่อนไขเฉพาะ ที่ระบบไม่เคยเรียนรู้มาก่อน

2.3.3 ผลกระทบของการพึ่งพาปัญญาประดิษฐ์ต่อผู้เรียน

การนำปัญญาประดิษฐ์มาใช้เป็นเครื่องมือช่วยเขียนโปรแกรมโดยขาดวิจารณญาณ ได้นำไปสู่พฤติกรรมการพึ่งพาเทคโนโลยีมากเกินไป ซึ่งส่งผลกระทบอย่างรุนแรงต่อกระบวนการเรียนรู้ของผู้เรียนใน 3 มิติหลัก ได้แก่ 1) ด้านความเข้าใจ ผู้เรียนอาจจดจำไวยากรณ์หรือแนวทางการเขียนโปรแกรมที่ผิดพลาดซึ่งเกิดจากภาวะหลอนของระบบ 2) ด้านทักษะการแก้ปัญหา กระบวนการคิดวิเคราะห์เชิงคำนวณของผู้เรียนจะถูกลดทอนลง เนื่องจากความคุ้นเคยกับการคัดลอกคำสั่ง (Copy and Paste) โดยไม่ผ่านกระบวนการไตร่ตรองและการสืบเสาะหาความจริง [11] และ 3) ด้านจริยธรรมดิจิทัล ผู้เรียนมีความเสี่ยงสูงที่จะนำรหัสคำสั่งที่ไม่มีความปลอดภัย แฝงไปด้วยช่องโหว่ หรือละเมิดลิขสิทธิ์ ไปใช้งานในระบบจริง ซึ่งอาจก่อให้เกิดความเสียหายในวงกว้างได้

2.4 สมรรถนะการเขียนโปรแกรมภาษา Python และนวัตกรรมโจทย์ลวง (Trap Tasks)

การประยุกต์ใช้ปัญญาประดิษฐ์ในการเรียนการสอนวิชาคอมพิวเตอร์ได้พลิกโฉมความหมายของสมรรถนะไปอย่างสิ้นเชิง โดยได้แบ่งประเด็นการศึกษาที่เกี่ยวข้องกับสมรรถนะการเขียนโปรแกรมและนวัตกรรมที่ใช้ในการวิจัยออกเป็น 3 ประเด็น ดังนี้

2.4.1 สมรรถนะการเขียนโปรแกรมภาษา Python ในยุคปัญญาประดิษฐ์

ภาษา Python เป็นภาษาคอมพิวเตอร์ที่ได้รับความนิยมสูงสุดในปัจจุบัน เนื่องจากมีไวยากรณ์ที่อ่านเข้าใจง่ายและเป็นรากฐานสำคัญของวิทยาการข้อมูล (Data Science) และการพัฒนาปัญญาประดิษฐ์ อย่างไรก็ตาม สมรรถนะการเขียนโปรแกรม (Programming Competency) ในยุคปัจจุบันไม่ได้จำกัดอยู่เพียงความสามารถในการจดจำไวยากรณ์หรือการเขียนรหัสคำสั่งจากศูนย์อีกต่อไป [12] เมื่อผู้เรียนสามารถใช้ปัญญาประดิษฐ์แบบสร้างสรรค์ (Generative AI) ในการผลิตรหัสคำสั่งได้อย่างรวดเร็ว สมรรถนะที่สำคัญที่สุดจึงเปลี่ยนผ่านไปสู่ทักษะการคิดเชิงคำนวณ (Computational Thinking) โดยเฉพาะอย่างยิ่งในด้านการประเมินและการตรวจสอบรหัสคำสั่ง (Code Verification) ผู้เรียนจำเป็นต้องมีวิจารณญาณในการอ่าน ทำความเข้าใจ และวิเคราะห์ตรรกะเบื้องหลังรหัสคำสั่งที่ AI สร้างขึ้น เพื่อป้องกันข้อผิดพลาดที่อาจนำไปสู่ความเสียหายในการทำงานจริง [13]

2.4.2 นวัตกรรมโจทย์ลวง (Trap Tasks) และการเรียนรู้ผ่านข้อผิดพลาด

เพื่อพัฒนาสมรรถนะการตรวจสอบดังกล่าว งานวิจัยนี้ได้พัฒนานวัตกรรม โจทย์ลวง (Trap Tasks) ซึ่งเป็นชุดสถานการณ์จำลองที่จงใจออกแบบให้มีข้อผิดพลาดแฝงอยู่ นวัตกรรมนี้มีรากฐานมาจากแนวคิดความล้มเหลวที่สร้างสรรค์ (Productive Failure) ของ Kapur [14] ที่ระบุว่า การปล่อยให้ผู้เรียนเผชิญหน้ากับปัญหาที่ซับซ้อนและประสพกับความล้มเหลวในช่วงแรกภายใต้สภาพแวดล้อมที่ปลอดภัย จะช่วยกระตุ้นให้เกิดความขัดแย้งทางปัญญา (Cognitive Dissonance) และนำไปสู่การเรียนรู้ที่ลึกซึ้งขึ้น นอกจากนี้ การให้นักศึกษาตรวจสอบโจทย์ลวงยังสอดคล้องกับทฤษฎีการเรียนรู้เชิงประสบการณ์ (Experiential Learning Theory) ของ Kolb [15] ที่ส่งเสริมให้ผู้เรียน เรียนรู้ผ่านการสังเกตความผิดพลาด (Reflective Observation) และทดลองแก้ไขรหัสคำสั่งด้วยตนเอง (Active Experimentation)

2.4.3 การประยุกต์ใช้โจทย์ลวง 5 รูปแบบเพื่อรับมือกับภาวะหลอนของ AI

ในการวิจัยครั้งนี้ ได้ออกแบบโจทย์ลวง 5 รูปแบบเพื่อครอบคลุมลักษณะภาวะหลอนของ AI ที่พบบ่อยที่สุด ได้แก่ 1) Library Hallucination การอ้างอิงโมดูลที่ไม่มีอยู่จริง ซึ่งฝึกให้ผู้เรียนตรวจสอบแหล่งข้อมูลอ้างอิงทางการ 2) Logic Hallucination การวางกับดักเงื่อนไขเฉพาะ (Edge Cases) เช่น การหารด้วยศูนย์ เพื่อทดสอบความรอบคอบเชิงตรรกะ 3) Method Hallucination การสร้างชื่อฟังก์ชันที่ AI มโนขึ้นมาเอง 4) Syntax Mixing การผสมไวยากรณ์ของภาษาอื่น (เช่น วงเล็บปีกกาของ C++ หรือ Java) เข้ามาในภาษา Python และ 5) Algorithm Precedence ข้อผิดพลาดด้านลำดับ

ความสำคัญของตัวดำเนินการทางคณิตศาสตร์ การให้ผู้เรียนได้สวมบทบาทเป็น ผู้ตรวจสอบ (Critical Validator) ผ่านการแก้ โจทย์ลวงทั้ง 5 รูปแบบนี้ จะช่วยเสริมสร้างภูมิคุ้มกันทางดิจิทัลและยกระดับสมรรถนะการเขียนโปรแกรมได้อย่างเป็นรูปธรรม

2.5 สรุปความเชื่อมโยงและช่องว่างการวิจัย (Research Gap)

จากงานวิจัยที่กล่าวมาข้างต้น พบว่า แม้จะมีงานวิจัยต่างประเทศที่พูดถึงข้อจำกัดและภาวะหลอนของ AI รวมถึง งานวิจัยในประเทศที่ยืนยันความสำเร็จของการสอนแบบปฏิบัติการเชิงรุก แต่ยังไม่มีการวิจัยใดที่นำประเด็นทั้งสองมาบูรณา การเข้าด้วยกันเพื่อสร้างเป็นชุดกิจกรรมการเรียนรู้แบบเร่งรัด (Blitz Model) ที่มีเป้าหมายเฉพาะเจาะจงในการจับผิด AI ด้วย เหตุนี้ จึงนำข้อค้นพบดังกล่าวมาเป็นฐานคิดในการออกแบบช่องว่างการวิจัย โดยนำจุดแข็งของการสาธิตสดมาใช้ในการเปิดมุมมอง เรื่องความผิดพลาดของ AI และนำจุดแข็งของการเรียนรู้เชิงรุกมาสร้างเป็นนวัตกรรม โจทย์ลวง (Trap Tasks) จำนวน 5 รูปแบบ จัดการเรียนรู้แบบกระชับ เรียนภายในเวลา 180 นาที เพื่อมุ่งแก้ปัญหาการขาดวิจรณ์ญาณในการใช้ AI และ ยกระดับสมรรถนะการเขียนโปรแกรมภาษา Python ของผู้เรียนให้พร้อมรับมือกับความท้าทายในโลกยุคดิจิทัลอย่างเป็น รูปธรรม

3. วิธีดำเนินการวิจัย

3.1 ประชากรและกลุ่มตัวอย่าง

ประชากรที่ใช้ในการวิจัยครั้งนี้ คือ นักศึกษาระดับปริญญาตรี ชั้นปีที่ 1 คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัย ราชภัฏนครปฐม ที่ลงทะเบียนเรียนในรายวิชา การเขียนโปรแกรมคอมพิวเตอร์ จำนวน 35 คน เพื่อเข้าร่วมกิจกรรมการเรียนรู้ แบบเร่งรัดภายในระยะเวลาที่กำหนด ดังแสดงในภาพที่ 1

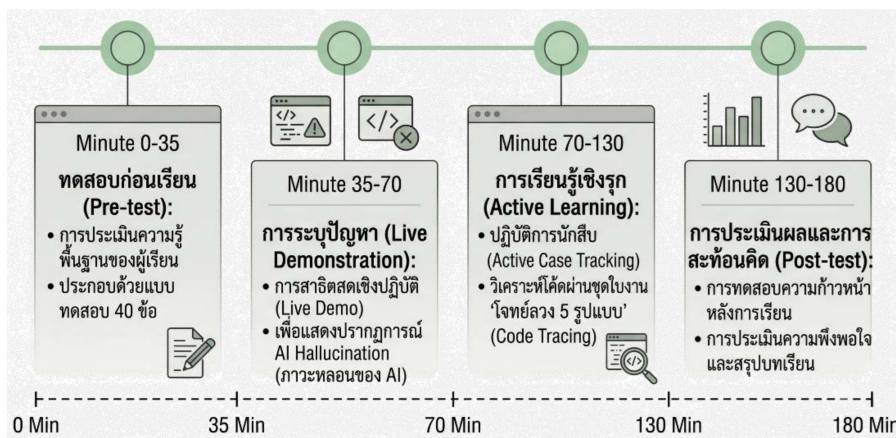
3.2 เครื่องมือที่ใช้ในการวิจัย

เครื่องมือที่ใช้ในการวิจัยและเก็บรวบรวมข้อมูลในครั้งนี้ แบ่งออกเป็น 3 ประเภทหลัก ได้แก่

- เครื่องมือที่ใช้ในการทดลองปฏิบัติการ ผ่านการรันชุดคำสั่งใน Google Colab และ การสอนแบบเชิงรุกโดยใช้ สไลด์ประกอบการสอนปูพื้นฐานเรื่อง "ภาวะหลอนของ AI (AI Hallucination)" และเทคนิคการจับผิดกับดัก 5 รูปแบบกับการจัดการเรียนรู้แบบสาธิต ซึ่งการเรียนรู้ใช้เวลาจัดการเรียนรู้อรวม 180 นาที (ภาพที่ 2) บูรณาการ พร้อมกับชุดใบงานกิจกรรมโจทย์ลวง (Trap Tasks) จำนวน 5 รายการ ("แกะรอยโค้ด (Code Tracing)" และ ระบุตำแหน่งข้อผิดพลาด) ดังแสดงในตารางที่ 1



ภาพที่ 1 บรรยากาศในห้องเรียน



ภาพที่ 2 รูปแบบการเรียนรู้แบบเร่งรัด

ตารางที่ 1 โจทย์ลวง 5 รูปแบบเพื่อครอบคลุมลักษณะภาวะหลอนของ AI

รูปแบบ	Error Name	Description	Detection Skill
1.	Library Hallucination	อ้างอิงโมดูลที่ไม่มีอยู่จริง	ฝึกการตรวจสอบแหล่งข้อมูลอ้างอิงทางการ
2.	Library Hallucination	วางกับดักเงื่อนไข (Edge Cases) เช่น หารด้วยศูนย์	ทดสอบความรอบคอบเชิงตรรกะ
3.	Method Hallucination	สร้างชื่อฟังก์ชันที่ AI ไม่ขึ้นขึ้นมาเอง	ฝึกการทวนสอบคำสั่ง (Syntax Verification)
4.	Syntax Mixing	ผสมไวยากรณ์ภาษาอื่น (เช่น ใส่วงเล็บปีกกาใน Python)	สร้างความเข้าใจโครงสร้างภาษาที่ถูกต้อง
5.	Algorithm Precedence	ลำดับความสำคัญตัวดำเนินการผิด	ฝึกวิเคราะห์ตรรกะทางคณิตศาสตร์

- เครื่องมือที่ใช้ในการวัดผลสัมฤทธิ์ ได้แก่ แบบทดสอบวัดผลสัมฤทธิ์ทางการเรียนก่อนเรียนและหลังเรียน (Pre-test /Post-test) เรื่องสมรรถนะการเขียนโปรแกรมภาษา Python และการตรวจจับภาวะหลอนของปัญญาประดิษฐ์ ลักษณะเป็นข้อสอบปรนัยชนิดเลือกตอบ 4 ตัวเลือก จำนวน 40 ข้อ
- เครื่องมือที่ใช้ในการประเมินความรู้สึกรู้ ได้แก่ แบบประเมินความพึงพอใจของผู้เรียนที่มีต่อการจัดกิจกรรมการเรียนรู้ ลักษณะเป็นมาตราส่วนประมาณค่า (Rating Scale) 5 ระดับ ตามวิธีของลิเคิร์ต (Likert)

3.3 การสร้างและหาคุณภาพเครื่องมือ

งานวิจัยนี้ ได้ดำเนินการสร้างและหาคุณภาพของเครื่องมือแต่ละชนิดอย่างเป็นระบบ โดยมีขั้นตอนดังนี้

3.3.1 แผนการจัดการเรียนรู้และชุดใบงานโจทย์ลวง

ผู้วิจัยเริ่มต้นจากการศึกษาหลักสูตร คำอธิบายรายวิชา และทฤษฎีที่เกี่ยวข้องกับการเรียนรู้เชิงรุก (Active Learning) และการสอนแบบสาธิต (Demonstration Method) จากนั้นจึงดำเนินการยกร่างแผนการจัดการเรียนรู้แบบเร่งรัด (Blitz Model) จำนวน 180 นาที ประกอบด้วย 1) ชี้นำและการสาธิตสด 2) ขั้นปฏิบัติการผ่านชุดโจทย์ลวง 5 รูปแบบ และ

3) ขึ้นสรุปและประเมินผล นำแผนการจัดการเรียนรู้และใบงานเสนอต่อผู้เชี่ยวชาญจำนวน 3 ท่าน เพื่อตรวจสอบความเที่ยงตรงเชิงเนื้อหา (Content Validity) โดยใช้ดัชนีความสอดคล้อง (Index of Item-Objective Congruence: IOC) ผลการประเมินพบว่า แผนการจัดการเรียนรู้และใบงานมีค่าความสอดคล้องอยู่ระหว่าง 0.67 ถึง 1.00 ซึ่งผ่านเกณฑ์ที่กำหนดและสามารถนำไปใช้ได้จริง

3.3.2 แบบทดสอบวัดผลสัมฤทธิ์ทางการเรียน

ผู้วิจัยสร้างข้อสอบปรนัย 4 ตัวเลือก จำนวน 40 ข้อ ตามตารางวิเคราะห์ข้อสอบ (Test Specification) ที่ครอบคลุมระดับพฤติกรรมการณ์การเรียนรู้ด้านความจำ ความเข้าใจ การนำไปใช้ และการวิเคราะห์ จากนั้นนำข้อสอบเสนอต่อผู้เชี่ยวชาญทั้ง 3 ท่าน หลังจากนั้น ผู้วิจัยได้นำข้อสอบไปทดลองใช้ (Try-out) กับนักศึกษาที่ไม่ใช่กลุ่มตัวอย่าง จำนวน 35 คน เพื่อนำผลคะแนนมาวิเคราะห์ความก้าวหน้าการเรียนรู้ (r)

3.3.3 แบบประเมินความพึงพอใจ

ผู้วิจัยสร้างแบบประเมินความพึงพอใจจำนวน 10 ข้อคำถาม ก่อนนำไปจัดทำเป็นรูปแบบออนไลน์ผ่าน Google Forms เพื่อใช้เก็บข้อมูลจากกลุ่มตัวอย่าง

3.4 รูปแบบการวิจัยและการเก็บรวบรวมข้อมูล

การวิจัยครั้งนี้เป็นการวิจัยเชิงทดลองเบื้องต้น (Pre-experimental Research) โดยใช้แผนการทดลองแบบกลุ่มเดียวทดสอบก่อนเรียนและหลังเรียน (One-Group Pretest-Posttest Design) ผู้วิจัยดำเนินการเก็บรวบรวมข้อมูลด้วยตนเองกับกลุ่มตัวอย่างในภาคการศึกษาที่ 2/2568 ซึ่งใช้ระยะเวลาในการดำเนินกิจกรรมรวมทั้งสิ้น 180 นาที มีขั้นตอนและรายละเอียดในการเก็บรวบรวมข้อมูลตามลำดับดังนี้

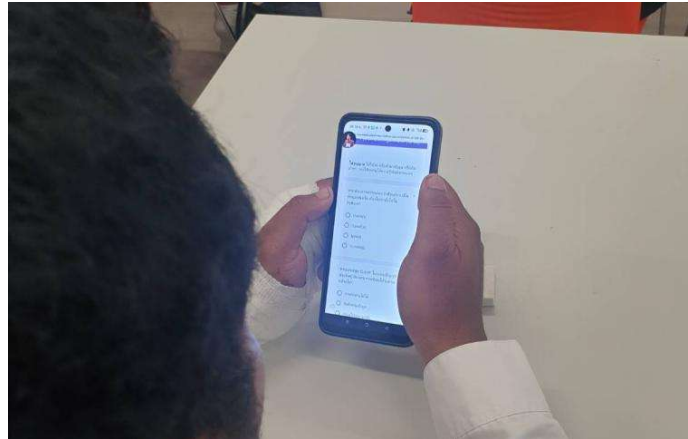
- ขั้นทดสอบก่อนเรียน เริ่มจาก ผู้วิจัยชี้แจงวัตถุประสงค์และขั้นตอนการจัดกิจกรรม จากนั้นให้นักศึกษากลุ่มตัวอย่างทำแบบทดสอบวัดผลสัมฤทธิ์ทางการเรียนก่อนเรียน ซึ่งเป็นแบบทดสอบปรนัยจำนวน 40 ข้อ ผ่านระบบออนไลน์ โดยใช้เวลา 35 นาที เพื่อเก็บคะแนนพื้นฐานและประเมินความรู้เดิมของผู้เรียน
- ขั้นดำเนินการทดลอง ผู้วิจัยดำเนินการจัดกิจกรรมการเรียนรู้ตามแผนที่กำหนดไว้ โดยเริ่มจากการบรรยายและสาธิตเพื่อชี้ให้เห็นข้อผิดพลาดและข้อจำกัดของปัญญาประดิษฐ์ เป็นเวลา 35 นาที จากนั้นให้นักศึกษาลงมือปฏิบัติกิจกรรมกลุ่มเชิงรุก (Active Learning) ผ่านชุดใบงานโจทย์ลวง (Trap Tasks) เพื่อฝึกทักษะการตรวจสอบโค้ด แยกโค้ด และจับผิดปัญญาประดิษฐ์ เป็นเวลา 60 นาที
- ขั้นทดสอบหลังเรียน ทำเมื่อสิ้นสุดกิจกรรมการเรียนรู้ ผู้วิจัยให้นักศึกษากลุ่มตัวอย่างทำแบบทดสอบวัดผลสัมฤทธิ์ทางการเรียนหลังเรียน ซึ่งเป็นแบบทดสอบชุดเดิมกับก่อนเรียนผ่านระบบออนไลน์เป็นเวลา 35 นาที เพื่อวัดพัฒนาการและผลสัมฤทธิ์ทางการเรียนหลังได้รับการจัดกิจกรรม จากนั้นทำการเฉลย สรุปและสะท้อนความคิดและให้นักศึกษาตอบแบบประเมินความพึงพอใจที่มีต่อการจัดกิจกรรมการเรียนรู้ในท้ายที่สุด เป็นเวลา 15 นาที ดังภาพที่ 3

3.5 การวิเคราะห์ข้อมูลและสถิติที่ใช้

งานวิจัยนำข้อมูลคะแนนผลสัมฤทธิ์ทางการเรียนและผลการประเมินความพึงพอใจที่เก็บรวบรวมได้ มาดำเนินการตรวจสอบความถูกต้องและวิเคราะห์ทางสถิติ โดยมีรายละเอียดของสถิติที่ใช้ดังนี้

- สถิติพื้นฐานที่ใช้ คือ ค่าเฉลี่ย ($\bar{x}$) และส่วนเบี่ยงเบนมาตรฐาน (S.D.) สำหรับอธิบายข้อมูลทั่วไปและระดับความพึงพอใจของผู้เรียน

- สถิติที่ใช้ทดสอบสมมติฐาน ผู้วิจัยได้วิเคราะห์เปรียบเทียบความแตกต่างของคะแนนเฉลี่ยผลสัมฤทธิ์ทางการเรียนก่อนเรียนและหลังเรียนของนักศึกษาในกลุ่มตัวอย่าง โดยใช้สถิติทดสอบค่าทีแบบกลุ่มตัวอย่างไม่เป็นอิสระต่อกัน โดยกำหนดระดับนัยสำคัญทางสถิติที่ระดับ 0.01
- คำนวณหาขนาดอิทธิพล (r) เพื่อประเมินขนาดและระดับความแตกต่างของผลลัพธ์ที่เกิดขึ้นจากการจัดกิจกรรมการเรียนรู้ในทางปฏิบัติ



ภาพที่ 3 ทดสอบหลังเรียน

4 ผลการศึกษา

จากข้อมูลที่เก็บรวบรวมจากกลุ่มตัวอย่างจำนวน 35 คน ได้นำมาวิเคราะห์ด้วยวิธีการทางสถิติ คะแนนจากแบบทดสอบวัดผลสัมฤทธิ์ทางการเรียนก่อนเรียนและหลังเรียน จำนวน 40 ข้อ ได้นำมาวิเคราะห์เปรียบเทียบความแตกต่างโดยใช้สถิติทดสอบค่าทีแบบกลุ่มตัวอย่างไม่เป็นอิสระต่อกัน (t-test for Dependent Samples) ดังแสดงในตารางที่ 2

จากตารางที่ 1 พบว่า คะแนนเฉลี่ยผลสัมฤทธิ์ทางการเรียนหลังเรียนของผู้เรียนที่เรียนด้วยชุดกิจกรรมการเรียนรู้ มีค่าเท่ากับ 29.77 คะแนน (S.D. = 5.67) คิดเป็นร้อยละ 74.43 ของคะแนนเต็ม ซึ่งสูงกว่าคะแนนเฉลี่ยก่อนเรียนที่มีค่าเท่ากับ 22.31 คะแนน (S.D. = 4.76) คิดเป็นร้อยละ 55.78 ของคะแนนเต็ม เมื่อทำการทดสอบความแตกต่างของค่าเฉลี่ยด้วยสถิติ Paired Samples t-test พบว่า คะแนนเฉลี่ยหลังเรียนสูงกว่าก่อนเรียนอย่างมีนัยสำคัญทางสถิติที่ระดับ .001 ($t(34) = -7.114, p < 0.001$) และเมื่อพิจารณาขนาดอิทธิพล (Effect Size) พบว่ามีค่า $r = 0.773$ ซึ่งจัดอยู่ในระดับสูงมาก (Large Effect Size) ผลลัพธ์ดังกล่าวสามารถอธิบายผลตามหลักการและทฤษฎีได้ดังนี้

ตารางที่ 2 การเปรียบเทียบคะแนนผลสัมฤทธิ์ทางการเรียนก่อนเรียนและหลังเรียน

การทดสอบ	จำนวน (n)	คะแนนเต็ม	$\bar{X}$	S.D.	t(34)	p-value
ก่อนเรียน (Pre-test)	35	40	22.31	4.76	-7.114	0.00*
หลังเรียน (Post-test)	35	40	29.77	5.12		

*มีนัยสำคัญทางสถิติที่ระดับ .001

การที่คะแนนเฉลี่ยเพิ่มขึ้นอย่างก้าวกระโดดถึง 7.46 คะแนน และมีขนาดอิทธิพล (r) ที่สูงมาก เป็นหลักฐานเชิงประจักษ์ว่า ชุดกิจกรรมการเรียนรู้ขั้นนี้ได้รับการออกแบบและจัดลำดับเนื้อหาอย่างมีระบบ มีความเหมาะสมกับระดับความสามารถของผู้เรียน ทำให้ผู้เรียนสามารถพัฒนาองค์ความรู้ ทักษะ หรือกระบวนการคิดจากเดิมได้อย่างมีนัยสำคัญ ไม่ได้

เกิดจากความบังเอิญหรือปัจจัยภายนอก เมื่อพิจารณาค่าส่วนเบี่ยงเบนมาตรฐาน (S.D.) พบว่าหลังเรียนมีค่าเพิ่มขึ้นเล็กน้อยจาก 4.76 เป็น 5.12 แสดงให้เห็นว่า ภายหลังการใช้ชุดกิจกรรมการเรียนรู้ ผู้เรียนกลุ่มสูง (High-achievers) สามารถตั้งศักยภาพของตนเองออกมาจนทำคะแนนได้เกือบเต็ม (สูงสุด 39 คะแนน) ขณะเดียวกันผู้เรียนกลุ่มอ่อนก็มีคะแนนที่ขยับสูงขึ้นตามเกณฑ์ขั้นต่ำ ส่งผลให้ช่วงของคะแนนกว้างขึ้นเล็กน้อย ผลการวิจัยนี้บ่งชี้ว่า ชุดกิจกรรมการเรียนรู้ดังกล่าวประสบความสำเร็จในการพัฒนาผลสัมฤทธิ์ทางการเรียนของผู้เรียนอย่างแท้จริง และสมควรแก่การนำไปประยุกต์ใช้ในการจัดการเรียนการสอนในเนื้อหาหรือบริบทที่ใกล้เคียงกันต่อไป

การจัดกิจกรรมการเรียนรู้ ภายใต้กรอบเวลา 180 นาที ได้รับเสียงสะท้อนเชิงบวกจากผู้เข้าร่วมกิจกรรม โดยในภาพรวมผู้เรียนมีความพึงพอใจอยู่ในระดับมาก (Mean = 4.15, S.D. = 0.49) และเมื่อนำผลประเมินความพึงพอใจนี้ไปพิจารณาร่วมกับผลสัมฤทธิ์ทางการเรียน (Pre-test / Post-test) ที่เคยวิเคราะห์ไว้ก่อนหน้านี้ จะทำให้เห็นมิติความสำเร็จในเชิงโครงสร้าง ดังตารางที่ 3

ตารางที่ 3 ผลการประเมินความพึงพอใจของผู้เรียนที่มีต่อชุดกิจกรรมการเรียนรู้

ข้อที่	รายการประเมินความพึงพอใจต่อกิจกรรมการเรียนรู้	ค่าเฉลี่ย (Mean)	S.D.	ระดับความพึงพอใจ
1	เนื้อหามีความเหมาะสมกับระดับพื้นฐานของผู้เรียน	4.17	0.62	มาก
2	เนื้อหากรณี AI Hallucination มีประโยชน์ต่อการใช้งานจริง	4.26	0.70	มาก
3	การสาธิต (Demonstration) ช่วยให้เข้าใจเนื้อหาได้ชัดเจน	4.17	0.66	มาก
4	กิจกรรมโจทย์ลวง (Trap Tasks) ช่วยฝึกทักษะการแก้ปัญหา	4.29	0.75	มาก
5	การใช้เครื่องมือ AI ช่วยให้การเรียนรู้สนุกสนานมากขึ้น	3.94	0.84	มาก
6	เครื่องมือ AI ที่ใช้เข้าถึงง่ายและสะดวกต่อการฝึกปฏิบัติ	4.40	0.55	มากที่สุด
7	มีความมั่นใจในการตรวจสอบความถูกต้องของ Code มากขึ้น	3.97	0.75	มาก
8	สามารถระบุจุดที่ AI เกิดภาวะหลอนได้อย่างถูกต้อง	4.11	0.68	มาก
9	ระยะเวลา 180 นาทีมีความเหมาะสมกับกิจกรรม	4.06	0.84	มาก
10	ความพึงพอใจในภาพรวมต่อกิจกรรมการเรียนรู้	4.11	0.72	มาก

ในจุดแข็งเชิงกระบวนการ สิ่งที่ประสบความสำเร็จสูงสุดของกิจกรรมนี้คือการเลือกสรรเครื่องมือและการออกแบบรูปแบบการจัดกิจกรรมการเรียนรู้ (Active Learning) ที่ตอบโจทย์พฤติกรรมของผู้เรียน โดยผู้เรียนมีความเห็นเป็นเอกฉันท์และพึงพอใจในระดับมากที่สุดต่อข้อความที่ว่า เครื่องมือ AI ที่ใช้เข้าถึงง่ายและสะดวกต่อการฝึกปฏิบัติ (Mean = 4.40, S.D. = 0.55) ค่าความแปรปรวน (S.D.) ที่ต่ำที่สุดในแบบประเมินนี้ บ่งชี้ว่าสภาพแวดล้อมทางเทคโนโลยีที่ผู้สอนจัดเตรียมไว้มีความเสถียร ซึ่งช่วยลดความยุ่งยากทางเทคนิคลงได้อย่างมีประสิทธิภาพ ส่งผลให้ผู้เรียนสามารถมุ่งเป้าไปที่การทำกิจกรรมได้อย่างเต็มที่ ยิ่งไปกว่านั้น นวัตกรรมการสร้างกิจกรรมโจทย์ลวง (Trap Tasks) ยังสามารถทำคะแนนความพึงพอใจได้สูงเป็นอันดับสอง (Mean = 4.29, S.D. = 0.75) ซึ่งสอดคล้องอย่างมีนัยสำคัญกับการเพิ่มขึ้นของคะแนน Post-test ที่เคยทำได้ก่อนหน้านี้ สะท้อนให้เห็นว่าการเรียนรู้ผ่านการเผชิญหน้ากับความผิดพลาดจริง (Error-based Learning) โดยมีผู้สอนคอยทำหน้าที่สาธิตอย่างชัดเจน (Mean = 4.17) สามารถดึงดูดใจ กระตุ้นทักษะการคิดวิเคราะห์ และทำให้ผู้เรียนเข้าใจธรรมชาติของภาวะ AI หลอน (AI Hallucination) ได้อย่างเป็นรูปธรรมมากกว่าการท่องจำทฤษฎี

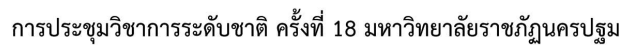
5. บทสรุป

จากการดำเนินงานวิจัยสามารถสรุปผลลัพธ์ที่สำคัญได้ว่า การพัฒนาชุดกิจกรรมการเรียนรู้เชิงรุกร่วมกับการจัดการเรียนรู้แบบสาธิตประสบความสำเร็จในการยกระดับสมรรถนะของผู้เรียนอย่างเป็นรูปธรรม แม้จะใช้กระบวนการเรียนรู้แบบเร่งรัดภายในระยะเวลาที่จำกัด ผู้เรียนได้รับการพัฒนาทั้งทักษะการเขียนโปรแกรมภาษาไพธอน และทักษะการตรวจจับภาวะหลอนของปัญญาประดิษฐ์ (AI Hallucination) เพิ่มขึ้นอย่างชัดเจนเมื่อเทียบกับก่อนเรียน ผลลัพธ์ที่เห็นชัดที่สุดคือการเปลี่ยนผ่านพฤติกรรมการเรียนรู้ จากเดิมที่ผู้เรียนมีแนวโน้มจะเป็นเพียงผู้คัดลอกคำสั่งที่พึ่งพาเทคโนโลยีมากเกินไป สู่การเป็นผู้ตรวจสอบ (Critical Validator) ซึ่งรู้จักตั้งคำถามและสืบเสาะหาความจริงก่อนนำรหัสคำสั่งไปใช้งาน นอกจากนี้ กิจกรรมโจทยลวง (Trap Tasks) ที่มอบบทบาทให้ผู้เรียนเป็นเสมือนนักสืบจับผิดปัญญาประดิษฐ์ ยังช่วยสร้างทัศนคติเชิงบวกต่อการเรียนรู้ และทำให้ผู้เรียนรู้สึกสนุกกับการเรียนรู้ผ่านความผิดพลาด (Productive Failure) ส่งผลให้เกิดความพึงพอใจต่อรูปแบบการเรียนการสอนในระดับสูง

ในด้านการนำผลการวิจัยไปใช้ประโยชน์ ชุดกิจกรรมนี้ได้สร้างภูมิคุ้มกันทางดิจิทัลให้แก่ผู้เรียน ทำให้สามารถนำทักษะการตรวจสอบไปประยุกต์ใช้กับการทำงานร่วมกับปัญญาประดิษฐ์แบบสร้างสรรค์ (Generative AI) ในชีวิตประจำวันและสายอาชีพในอนาคตได้อย่างปลอดภัย ช่วยป้องกันความเสียหายจากข้อผิดพลาดเชิงตรรกะหรือข้อมูลที่คลาดเคลื่อน ยิ่งไปกว่านั้น แนวคิดการสร้างโจทยลวงทั้ง 5 รูปแบบ ยังสามารถนำไปเป็นต้นแบบนวัตกรรมการสอนสำหรับครูและอาจารย์ผู้สอน เพื่อนำไปประยุกต์ใช้กับภาษาคอมพิวเตอร์อื่น ๆ หรือรายวิชาทั่วไปที่ใช้ปัญญาประดิษฐ์ช่วยสอน ท้ายที่สุด รูปแบบการสอนแบบสาธิตสอดคล้องกับการลงมือปฏิบัติในระยะเวลาที่จำกัด (Blitz Model) นี้ ยังเป็นข้อพิสูจน์ให้สถานศึกษาเห็นว่า การสร้างทักษะการคิดวิเคราะห์ไม่จำเป็นต้องใช้ระยะเวลานานเสมอไป หากแต่ขึ้นอยู่กับการออกแบบสถานการณ์ปัญหาที่ท้าทายและตรงจุดเป็นสำคัญ

6. เอกสารอ้างอิง (References)

- [1] Prince, M. (2004). Does active learning work? A review of the research. *Journal of Engineering Education*, 93(3), 223-231.
- [2] Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., & Wenderoth, M. P. (2014). Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences*, 111(23), 8410-8415.
- [3] Krathwohl, D. R. (2002). A revision of Bloom's taxonomy: An overview. *Theory Into Practice*, 41(4), 212-218.
- [4] Thayer, K., & Ko, A. J. (2017). Barriers faced by adult learners in computer science education "bootcamps". *Proceedings of the 2017 ACM Conference on International Computing Education Research (ICER)*, 245-253.
- [5] Tisana Khaemmanee. (2017). *Science of teaching: Knowledge for effective learning process management* (21st ed.). Chulalongkorn University Press. (In Thai)
- [6] Bandura, A., & Walters, R. H. (1977). *Social learning theory* (Vol. 1, pp. 33-52). Englewood Cliffs, NJ: Prentice-hall.
- [7] Collins, A., Brown, J. S., & Newman, S. E. (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In L. B. Resnick (Ed.). *Knowing, learning, and instruction: Essays in honor of Robert Glaser*, 453-494.
- [8] Rubin, M. A. (2013). The effectiveness of live-coding to teach introductory programming. *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, 651-656.



- นวัตกรรมการวิจัยสู่สังคมอัจฉริยะและการพัฒนาอย่างยั่งยืน