Exploring Artificial Intelligence as a Tool for Differentiation and Inquiry in IB Computer Science

Rajesh Kumar Jha

University College Fairview

Abstract

This mixed-methods study examined the use of artificial intelligence (AI)-supported prompts to enhance the ability of IBDP Computer Science students to work more independently towards the outcomes of a selected programming unit on arrays. Grounded in a problem of practice concerning student dependence on direct teacher explanation while supporting mixed levels of readiness, this bounded single-case study with an embedded intervention used AI-supported prompts designed for inquiry-based and differentiated learning. Quantitative data were collected from 36 students (22 DP1, 14 DP2). Qualitative data were collected through interviews with three teachers, supported by student reflections, classroom observations, and learning artifacts. Findings indicate that AI-supported prompts improved conceptual understanding, supported a partial shift toward greater autonomy, and enabled more personalized pacing. However, these gains were conditional: teacher mediation remained essential for verification, clarification, and responsible use. The study contributes subject-specific evidence on how AI can function as a tutor-like scaffold in IBDP Computer Science when embedded within structured, differentiated, and teacher-mediated learning environments.

Keywords: *Artificial Intelligence in education, IBDP Computer Science, differentiated instruction, inquiry-based learning, computational thinking*

Introduction

Artificial intelligence is transforming education through feedback provision, task personalization, and assistance with inquiry-based learning processes. In the International Baccalaureate Diploma Programme (IBDP), which promotes student-centered learning, conceptual understanding, and academic integrity, AI has particular relevance. IBDP Computer Science uniquely blends conceptual content, problem-solving skills, and independent inquiry expectations. The recently revised DP Computer Science curriculum (first teaching 2025, first assessment 2027) emphasizes computational thinking, modern application of subject content, and student justification of responses (International Baccalaureate Organization [IBO], 2025).

As a teacher of this subject, an ongoing classroom challenge has been that students often approach difficult programming concepts with strong reliance on teacher explanation and direction. Some students enter with prior programming experience, while others encounter important concepts for the first time. In mixed-ability classes, teachers struggle to support novices without slowing the pace for more experienced students, all while maintaining rigor and student ownership. This issue becomes especially visible when students are expected to ask questions, debug code, test algorithms, and explain concepts rather than simply reproduce answers provided by the teacher.

AI may help address this tension. Rather than using AI to generate answers quickly, students could use AI to generate structured questions, explanations, and personalized assistance. By designing AI-supported prompts, educators could help students with concept exploration, questioning, idea testing, and perseverance. However, the use of AI in computer science education raises concerns about over-reliance, superficial understanding, authorship, and the visibility of student thinking (Holmes et al., 2019; Kasneci et al., 2023).

This study explores the use of AI-supported prompts to enable students to achieve curriculum goals in IBDP Computer Science through inquiry-based approaches and differentiated learning. The study is bounded within a selected unit on arrays and involves three teachers and 36 students (22 DP1, 14 DP2). Three research questions guide the study: how AI-supported prompts influence student autonomy and personalization of learning; how AI-supported prompts support inquiry-based and differentiated learning; and what are the perceived affordances and limitations of AI-supported prompts in helping students meet curriculum objectives.

Literature Review

Artificial Intelligence in Education (AIED) refers to the intentional use of computational systems that classify, predict, generate, or engage in discourse to support teaching and learning within justified pedagogical and ethical boundaries. Ouyang and Jiao (2021) describe three modes of human-AI joint arrangement: AI-Directed (instructional AI that adapts to learner models), AI-Supported (decision-support tools for teacher professional judgment), and AI-Empowered (students using AI to plan investigations and critique AI-generated artifacts).

The most common AI tools in education fall within four families. Intelligent Tutoring Systems provide hints and adapt tasks to skill levels, showing moderate improvements with constructive tasks and immediate feedback (VanLehn, 2011). Automated feedback engines provide grading and feedback on programming errors, but their educational impact depends on ability to justify feedback (Keuning et al., 2018). Learning Analytics Dashboards provide visualizations of student activities to guide differentiation (Jivet et al., 2020). Conversational and generative AI can produce programs, test cases, and explanations on demand, with studies cautioning on blind acceptance of plausible but incorrect results (Denny et al., 2023).

International literature reveals that AI tools are novel and powerful, but pedagogy and governance models need to evolve to keep reasoning visible. U.S. scholarship emphasizes teacher-driven changes in specific learning tasks with verification-first pedagogy (Barke et al., 2023; Denny et al., 2023). European guidance favors program- or department-level governance with non-negotiables such as viva voce, process portfolios, and declarations regarding AI use (Jisc, 2023; UK Department for Education, 2023). Both converge on the notion that AI is not a substitute for human expertise but a pedagogical mediator whose usefulness must be explicitly framed.

Research gaps identified include conceptual gaps where autonomy and personalization are defined too broadly for IBDP Computer Science; pedagogical and methodological gaps where cross-sectional surveys cannot show impact of specific AI-supported prompting; practical and contextual gaps where implementation influences are not explored; and lack of subject-specific intervention evidence.

Theoretical Framework

An integrated theoretical framework guides this study, consisting of constructivist inquiry, self-determination theory, differentiated instruction, and TPACK. Constructivist perspectives view learners as active questioners, testers, explainers, and revisers of ideas (Vygotsky, 1978). Self-determination theory emphasizes autonomy, competence, and relatedness as drivers of intrinsic motivation (Ryan & Deci, 2020). Differentiated instruction provides a pedagogical lens for student support regarding readiness, pace, and learning needs (Tomlinson, 2017). TPACK explores the integration of technological, pedagogical, and content knowledge (Mishra & Koehler, 2006).

Figure 2.0: Theoretical Framework

add in your TF here from your thesis

A layered diagram showing constructivism, SDT, differentiated instruction, TPACK, and sociocultural perspectives as design constraints at the top; teacher practices (prompt design, scaffolding, verification routines) as the mediating layer in the middle; and student processes (thinking, questioning, debugging, testing, explaining, revising) and learning outcomes (conceptual understanding, computational thinking, autonomy) at the bottom. The figure reinforces the claim that AI does not enhance learning just by being there; it depends on how the teacher uses prompt and task design, scaffolds, checking processes, and accountability.

The framework suggests that any improvement from the use of AI-supported prompts in IBDP Computer Science is mediated, not direct. The theories positioned at the top of the model serve as design constraints. Teacher practices in the middle section constitute the key mediating layer. Student processes, thinking evidence, and learning outcomes are positioned below as the domain where effects will be seen.

Conceptual Framework

Figure: Conceptual Framework

Add in the figure of CF here from your thesis

diagram showing teacher facets (IB-aligned TPACK, assessment literacy, readiness to use AI critically) and implementation elements (well-designed prompts, differentiated scaffolds, *check-first routines, reflection opportunities) influencing student engagement (attention, confidence, persistence, quality of questions, reflective judgment), which in turn influence program-relevant outcomes (conceptual understanding, computational thinking in Theme B, ATL development, clearer authorship of student work).*

The conceptual framework moves the theoretical discussion into a logic of the study. It conceptualizes AI-supported prompting in IBDP Computer Science as a process of effects by design where teacher factors and implementation choices influence student engagement, questioning, and independence, which in turn impact the student learning-related outcomes valued in the programme.

Methodology

Research Design

A mixed-methods bounded single-case design was used with an embedded classroom intervention. The implementation of AI-supported arrays mini-units within IBDP Computer Science in one school constituted the bounded case. This design was selected because the research problem required both explanatory evidence and evidence of patterns: measurable evidence of change in student confidence, understanding, and independence, alongside qualitative explanation of how students and teachers experienced the intervention (Creswell & Plano Clark, 2018).

Participants and Sample

Participants comprised 36 students (22 DP1, 14 DP2) and 3 teachers. Students were distributed equally across two differentiated pathways: 18 completed the Beginner Arrays Mini-Unit and 18 completed the Advanced Arrays Mini-Unit. Prior programming experience varied: 14 students (38.9%) had learned a little programming before, 13 (36.1%) had studied programming as part of a previous course, 4 (11.1%) were entirely new, and 5 (13.9%) had significant prior experience.

Intervention Materials

The intervention included two differentiated one-week array mini-units. The beginner pathway was designed for students newer to programming, containing more scaffolding, straightforward language, smaller conceptual leaps, and tightly controlled AI prompts. The advanced pathway targeted more confident students, requiring more independent work, open-ended problem solving, tracing, debugging, testing, and explicit comparison of approaches. In both pathways, AI was presented as a tutor for learning, not as an answer engine.

Differentiated Pathway Structure:

- **Beginner Arrays Mini-Unit (Appendix 1):** Step-by-step explanations, simpler examples, guided tracing, structured debugging support, frequent self-check questions
- **Advanced Arrays Mini-Unit (Appendix 2):** Open-ended inquiry, independent code refinement, comparative solution analysis, complex testing and debugging scenarios

Data Collection and Analysis

Quantitative data were collected through pre-unit and post-unit student surveys measuring confidence, understanding, teacher dependence, and perceptions of AI support. Qualitative data were collected through semi-structured teacher interviews, student reflections, classroom observations, and student work samples. Quantitative analysis employed descriptive statistics (frequencies, percentages, mean responses). Qualitative analysis employed thematic analysis following Braun and Clarke (2022). Integration of quantitative and qualitative findings occurred through comparison and synthesis.

Findings

Quantitative Results

Before the intervention, students reported low confidence in key areas. Arrays understanding had a pre-unit mean of 2.36 (on a 1-5 scale), tracing confidence 2.36, testing confidence 2.53, and debugging confidence 2.50. Teacher dependence was high (M=3.56).

After the intervention, students reported substantially improved outcomes. Arrays understanding showed the clearest positive outcome, with a post-unit mean of 4.00 and 77.8% of students agreeing or strongly agreeing that they understood arrays more clearly. Tracing confidence increased to M=4.14 (77.8% agreement). Debugging confidence increased to M=3.81 (69.4% agreement). Students agreed that AI helped them understand array concepts more clearly (M=3.81), learn at a pace suited to their level (M=4.00), and support tracing, testing, debugging, or improving code (M=4.14).

At the same time, 69.4% of students agreed that teacher support was needed even while using AI, and 86.1% agreed that AI responses needed to be checked carefully before being trusted. Additionally, 80.6% indicated they would like to use this kind of AI-supported learning again for other Computer Science topics.

Table: Selected pre-post mean comparison (1-5 scale, N=36)

Construct	Pre-unit mean	Post-unit mean	Change

Arrays understanding	2.36	4.00	+1.64
Tracing confidence	2.36	4.14	+1.78
Debugging confidence	2.50	3.81	+1.31
Teacher dependence	3.56	4.25	+0.69

Comparison across learning pathways revealed that students in the advanced pathway reported stronger post-unit arrays understanding (M=4.28) and tracing confidence (M=4.39) than students in the beginner pathway (M=3.72 and M=3.89 respectively). Students in the beginner pathway reported a stronger perception that the pathway matched their readiness (M=4.11 versus 4.00).

Qualitative Themes from Teacher Interviews

Three teachers participated in semi-structured interviews. Thematic analysis revealed several recurring patterns.

All three teachers described wide variation in students' prior programming experience as a major challenge. Teachers identified decomposition, tracing, debugging, testing, and problem initiation as recurring difficulties. Arrays were described as particularly revealing because the unit required students to manage both conceptual and procedural demands simultaneously.

Teachers viewed the AI-supported intervention positively when AI was embedded in a structured learning routine. They valued that students could ask for another explanation or a different way of understanding a concept without immediately waiting for the teacher. The two differentiated pathways were seen as appropriate and useful.

Teachers indicated that AI supported inquiry most effectively when students used it to ask better follow-up questions, compare explanations, and explore alternative approaches. However, inquiry was not automatically produced by the presence of AI; when students used AI mainly to retrieve answers, the quality of inquiry weakened.

Teachers identified affordances including accessible explanation, flexible pacing, support for tracing and debugging, increased willingness to persist, and greater opportunity for differentiated access. Limitations included students trusting AI too quickly, accepting plausible explanations without checking, and confusing support with genuine understanding. Teacher mediation remained central for clarifying misconceptions, verifying AI responses, and reinforcing responsible use.

Qualitative Themes from Student Reflections

Student reflections (N=36) revealed several recurring patterns. Half of the students (50%) described uncertainty about how to begin programming tasks. All students (100%) described some form of teacher dependence before the intervention.

About two-fifths (39%) highlighted step-by-step explanation, repeated examples, or paced clarification as especially helpful in understanding arrays more clearly. A very large majority (94%) referred to tracing, testing, debugging, or code improvement as major areas in which AI supported their learning. Students stated that AI was useful because it could help trace code step by step, suggest better test cases, or offer debugging hints without fully replacing the task.

More than one-third (39%) explicitly linked the unit to stronger confidence or greater willingness to attempt programming tasks more independently. However, this increased independence was described as partial rather than complete. Students did not describe AI as replacing the teacher but as giving them more confidence to engage with difficulty before depending on teacher explanation.

Three-quarters (75%) referred to the need to verify AI responses or continuing uncertainty about whether AI explanations were fully correct. Students repeatedly noted that teacher support remained important for confirming whether an AI response was correct and relevant.

Triangulated Findings

Integration of quantitative and qualitative data revealed five triangulated patterns. First, improved conceptual clarity in arrays and related programming processes was supported across all data sources. Second, greater confidence and partial movement towards independence were evident, though teacher mediation remained important. Third, differentiated pathways supported mixed-readiness learners in different ways, with advanced students reporting stronger outcomes and beginner students valuing pathway fit. Fourth, AI functioned most effectively as a tutor-like support for tracing, testing, debugging, and iterative improvement. Fifth, teacher mediation and critical verification remained essential, with 86.1% of students agreeing that AI responses needed to be checked carefully.

Discussion

Interpretation of Findings

The findings indicate that AI-supported prompts contributed meaningfully to student understanding, supported autonomy, and enabled personalized pacing in IBDP Computer Science. The post-unit gains in arrays understanding (from M=2.36 to M=4.00) and tracing confidence (from M=2.36 to M=4.14) represent substantial perceived improvement. Student reflections consistently described step-by-step explanations, repeated examples, and debugging hints as valuable forms of support.

However, these gains were conditional rather than automatic. Teacher mediation remained essential for verification, clarification, and responsible use. The finding that 86.1% of students agreed that AI responses needed to be checked carefully underscores that AI was not experienced as a replacement for human judgment. Instead, AI functioned as a scaffold whose educational value depended on how it was integrated into structured learning routines (Holmes et al., 2019; Ouyang & Jiao, 2021).

The differentiated pathway design proved important in responding to variation in learner readiness. Students in the advanced pathway reported stronger arrays understanding and tracing confidence, while students in the beginner pathway more strongly valued the structured progression and pathway fit. This aligns with Tomlinson's (2017) emphasis that differentiation should provide different levels of scaffolding while maintaining common learning goals.

Relationship to Theoretical Framework

The findings align with constructivist perspectives, as AI-supported prompts functioned as scaffolding that helped students construct understanding through explanation, testing, and revision (Vygotsky, 1978). Self-determination theory is reflected in the partial movement towards greater autonomy, though this autonomy remained supported rather than absolute, consistent with Ryan and Deci's (2020) emphasis that autonomy develops through structured support that builds competence over time. TPACK is evident in the finding that effective AI integration depended on teacher orchestration of prompts, tasks, feedback processes, and guidance for responsible use (Mishra & Koehler, 2006).

Affordances and Limitations

Affordances of AI-supported prompts included improved conceptual understanding, real-time feedback and explanation, increased student autonomy, and personalized learning through differentiated pathways. Students particularly valued AI for tracing, testing, debugging, and code improvement.

Limitations included the need for critical verification, risk of superficial learning and overreliance (Kasneci et al., 2023), variability in AI response quality, and the continuing necessity of teacher mediation. Some students accepted AI responses without adequate checking, and AI-generated explanations could sometimes be incomplete or not fully appropriate for the specific task.

Implications for Practice

For IBDP Computer Science teachers, several practical implications emerge. AI-supported prompts should be embedded within structured learning routines that include understanding, writing, tracing, testing, debugging, improvement, and reflection. Verification-first processes should be established prior to incorporating automation. Assessment should reward process and reasoning, not just final output (Jisc, 2023). Clear expectations for disclosure and responsible use should be taught explicitly.

Differentiation remains important. The beginner and advanced pathways demonstrated that different levels of scaffolding and challenge can support learners with different readiness levels while maintaining common learning goals (Tomlinson, 2017). Schools should consider department-level coherence in AI expectations and shared language for discussing responsible use (UK Department for Education, 2023).

Limitations

This study has several limitations. The bounded single-case design means findings are closely tied to one school setting, limiting generalizability. The sample size (N=36) was sufficient for descriptive analysis within the case but relatively small. The intervention duration was one week, capturing short-cycle change rather than long-term development. Self-reported survey data are limited by subjectivity, though triangulation with teacher interviews, student reflections, and work samples partially addressed this. The absence of a control group means the study cannot claim causal effects with certainty.

Conclusion

This mixed-methods study examined AI-supported prompts in a selected arrays unit within IBDP Computer Science. The findings indicate that AI-supported prompts can contribute positively to conceptual understanding, supported autonomy, and personalized learning. Students reported clearer understanding of arrays, stronger confidence in tracing, testing, and debugging, and greater willingness to continue working through programming tasks rather than depending immediately on teacher explanation. AI was most effective when it functioned as a tutor-like support for explanation, tracing, testing, debugging, and iterative improvement.

However, these benefits were conditional. Teacher mediation remained essential for validation, clarification, and responsible use. AI did not replace the teacher but changed the form of support available. The differentiated pathways supported learners with different readiness levels in different ways, with the beginner pathway offering access through stronger scaffolding and the advanced pathway supporting refinement and greater independence.

The study contributes subject-specific evidence to a field often discussed in generic terms. For IBDP Computer Science, AI-supported prompts can make a meaningful contribution to inquiry-based and differentiated programming learning, but only when embedded within structured, teacher-mediated environments that preserve verification, explanation, and critical thinking. As AI continues to evolve, educators must be prepared to use it responsibly and effectively to enhance student learning outcomes while maintaining the principled values of the IB Diploma Programme.

References

Barke, S., James, M. B., & Polikarpova, N. (2023). Grounded copilot: How programmers interact with code-generating models. *Proceedings of the ACM on Programming Languages, 7*(OOPSLA), 1-27.

Braun, V., & Clarke, V. (2022). *Thematic analysis: A practical guide*. SAGE.

Creswell, J. W., & Plano Clark, V. L. (2018). *Designing and conducting mixed methods research* (3rd ed.). SAGE.

Denny, P., Kumar, V., & Giacaman, N. (2023). Conversing with Copilot: Exploring prompt engineering for solving CS1 problems using natural language. *Proceedings of the 54th ACM Technical Symposium on Computer Science Education*, 1136-1142.

Holmes, W., Bialik, M., & Fadel, C. (2019). *Artificial intelligence in education: Promises and implications for teaching and learning*. Center for Curriculum Redesign.

International Baccalaureate Organization. (2025). *Computer science guide: For first assessment in 2027*. IBO.

Jisc. (2023). *Generative AI in education: A guide for teaching and learning*. Jisc.

Jivet, I., Scheffel, M., Specht, M., & Drachsler, H. (2020). License to evaluate: Preparing learning analytics dashboards for educational practice. *Proceedings of the Tenth International Conference on Learning Analytics & Knowledge*, 31-40.

Kasneci, E., Sessler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., & Kasneci, G. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences, 103*, 102274.

Keuning, H., Jeuring, J., & Heeren, B. (2018). A systematic literature review of automated feedback generation for programming exercises. *ACM Transactions on Computing Education, 19*(1), 1-43.

Mishra, P., & Koehler, M. J. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers College Record, 108*(6), 1017-1054.

Ouyang, F., & Jiao, P. (2021). Artificial intelligence in education: The three paradigms. *Computers and Education: Artificial Intelligence, 2*, 100020.

Ryan, R. M., & Deci, E. L. (2020). Intrinsic and extrinsic motivation from a self-determination theory perspective: Definitions, theory, practices, and future directions. *Contemporary Educational Psychology, 61*, 101860.

Tomlinson, C. A. (2017). *How to differentiate instruction in academically diverse classrooms* (3rd ed.). ASCD.

UK Department for Education. (2023). *Generative artificial intelligence in education*. DfE.

VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist, 46*(4), 197-221.

Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Harvard University Press.