MODUL-AR.JOURNAL

RESEARCH ARTICLE
Vol. 9, No. 1, 2026, 1-20
DOI: 10.59389/modular.1853822

Implementing the 4C/ID Model for Parametric Design in Architectural Education

Mimarlık Eğitiminde Parametrik Tasarım İçin 4B/ÖT Modelinin Uygulanması

Hasan TAŞTAN[1*]

Received: 01.01.2026 - Accepted: 24.02.2026

Abstract

Parametric design (PD) education requires an algorithms-based logic that imposes considerably high cognitive load. The aim of this study, therefore, is to explore and determine the efficiency of the four-component instructional design model (4C/ID). The 4C/ID has a focus on authentic, whole-task learning, which was tested on a group of 14 senior architecture students on a 15-week undergraduate course on PD within mixed-design research. There were pre- and post-assessments of computational thinking (CT), CT self-efficacy (CTSE), and design thinking (DT), along with analyses of final projects and qualitative feedback. The quantitative data showed that there were significant gains for algorithmic thinking and problem-solving skills after taking the course. There were significant gains for CTSE factors such as abstraction and reasoning. On correlation analysis, there was a strong relationship between course performance and these technical skills, although there were no significant correlations with DT scores, which were found not to have changed much. This implies that though the 4C/ID model facilitated the transition of students from passive understanding to confident performance of procedures, achieving competencies in higher order design skills took longer. Qualitative feedback confirmed the effectiveness of the 4C/ID model but also highlighted two issues: visual complexity in the interface and data complexity, which could hinder students who are not sufficiently prepared. Therefore, effective PD education needs careful task design and support for logical reasoning to balance technical requirements with creative objectives.

Keywords: Parametric design education, 4C/ID model, Computational thinking, Algorithmic reasoning, Instructional design.

Özet

Parametrik tasarım (PT) eğitimi, oldukça yüksek bilişsel yük getiren algoritma tabanlı bir mantık gerektirir. Bu nedenle bu çalışmanın amacı, dört bileşenli öğretim tasarım modelinin (4B/ÖT) verimliliğini araştırmak ve belirlemektir. 4B/ÖT; otantik, bütünsel görev öğrenimine odaklanmaktadır ve karma tasarım araştırması kapsamında 15 haftalık bir PT lisans dersinde 14 son sınıf mimarlık öğrencisi üzerinde test edilmiştir. Bilgi işlemsel düşünme (BİD), bilgi işlemsel düşünme öz-yeterliği (BİDÖ) ve tasarım odaklı düşünme (TOD) ön ve son değerlendirmeleri ile, final projeleri ve nitel geri bildirim analizleri yapılmıştır. Nicel veriler, kursun ardından algoritmik düşünme ve problem çözme becerilerinde önemli kazanımlar olduğunu göstermiştir. Soyutlama ve akıl yürütme gibi BİDÖ faktörlerinde önemli kazanımlar olmuştur. Korelasyon analizinde, öğrenme performansı ile bu teknik beceriler arasında güçlü bir ilişki olduğu görülmüştür; ancak TOD puanları ile önemli bir korelasyon bulunmamıştır ve bu puanların fazla değişmediği tespit edilmiştir. Bu; 4B/ÖT modelinin, öğrencilerin pasif anlayıştan prosedürlerin kendinden emin bir şekilde uygulanmasına geçişini kolaylaştırmasına rağmen üst düzey tasarım becerilerinde yetkinlik kazanmanın daha uzun sürdüğünü ima etmektedir. Niteliksel geri bildirimler, 4B/ÖT modelinin etkinliğini doğrulamakta, ancak aynı zamanda iki sorunu da vurgulamaktadır: arayüzdeki görsel karmaşıklık ve veri karmaşıklığı. Bu da yeterince hazırlıklı olmayan öğrencileri engelleyebilir. Bu nedenle etkili PT eğitimi, teknik gereksinimleri yaratıcı hedeflerle dengelemek için dikkatli görev tasarımı ve mantıksal akıl yürütme desteği gerektirir.

Anahtar Kelimeler: Parametrik tasarım eğitimi, 4B/ÖT modeli, Bilgi işlemsel düşünme, Algoritmik akıl yürütme, Öğretim tasarımı.

Citation: Taştan, H. (2026). Implementing the 4C/ID model for parametric design in architectural education. *Modular Journal*, *9*(1), 1-20.

[1] Yıldız Technical University, Faculty of Architecture, Department of Architecture, Istanbul, Türkiye. hasantastn@gmail.com
* Corresponding Author

RESEARCH ARTICLE
Vol. 9, No. 1, 2026, 1-20
DOI: 10.59389/modular.1853822

Extended Abstract

Introduction: The design processes in today's digital world are complex and rely heavily on data. This requires a significant shift in how we educate architects. Unlike traditional approaches that focus on form, parametric design (PD) emphasizes a process-oriented mindset. In PD, form comes from the connections between algorithmic logic, rules, and variables. This approach can be mentally demanding, as it asks students to combine conceptual knowledge, procedural skills, and design reasoning. However, existing pedagogical approaches frequently prioritize technical skills, potentially overlooking the development of algorithmic thinking and metacognitive strategies.

Purpose: In this context, the aim is to assess the applicability and effectiveness of the four-component instructional design (4C/ID) model, which was developed for teaching complex skills in an undergraduate course in parametric design. Accordingly, the study uses a mixed-method approach to investigate the following three central research questions:

- RQ1. How are students' final grades (learning performance) related to their computational thinking, computational thinking self-efficacy, and design thinking scores, as indicated by the correlation analysis?
- RQ2. To what extent do students show measurable changes in computational thinking and computational thinking self-efficacy over the duration of the course, as evidenced by pre–post (t-test) comparisons?
- RQ3. What are students' perceptions of the effectiveness of the 4C/ID-based course structure, the level of task complexity, and the sufficiency of instructional support in developing their parametric design competencies?

Method: In this study, a mixed-methods approach was employed with a sample group consisting of 14 senior students in an architecture program in a 15-week parametric design class. In this 4C/ID course structure, there are four basic components:

- Learning tasks (LT) sequenced from simple to complex,
- Supportive information (SI) to facilitate the formation of cognitive schemas,
- Procedural information (PI) for routine skills, and
- Part-task practice (PTP) to ensure automation.

The learning process was divided into themes, starting from basic geometric manipulations (weeks 1-3) to data-informed systems (weeks 4-6), and finally, storytelling by moving images and montage (weeks 8-12). Both quantitative and qualitative methods were used simultaneously during the process of gathering data. In order to assess the cognitive and affective shifts of the participants, computational thinking (CT), computational thinking self-efficacy (CTSE), and design thinking (DT) scales were used as pre- and post-tests. The learning performance was assessed by two raters on the basis of learning achievements (final grades) by applying the Consensual Assessment Technique (CAT) regarding parameters like originality, appropriateness, and spatial richness. Further, there was a gathering of qualitative data pertaining to perceptions of the learning process.

Findings: According to the correlation analyses, course success (final grade) was most strongly correlated with algorithmic thinking ($r = .652$), problem-solving, and CTSE. In contrast, no meaningful relationship could be established between the sub-dimensions of DT and course performance. This may imply that technical and algorithmic rather than creative or user-centric strategies are more influential for success in the parametric design course. Another result of pre-post comparisons showed a significant development in students' algorithmic thinking and problem-solving skills. The most interesting change occurred in CTSE, where all sub-dimensions of the concept, such as abstraction, reasoning, decomposition, and generalization indicated large and significant changes. The result verifies that the step-by-step process and whole-task design of the 4C/ID model increase confidence and ability of PD learners. On the other hand, there was no statistically significant difference in DT. This result indicates that higher-order skills like DT are not easy to develop in a short duration of a technical course and require a protracted time span. The result was further supported by the qualitative analysis. The learners appreciated the pedagogical efficiency of the 4C/ID model; however, they considered the cognitive load to be substantial because of the visual nature of Rhino-Grasshopper and management of the structure of the data.

Conclusion: The results prove the 4C/ID model to be very effective for developing skills in technical fluency/algorithms in PD education. It was the whole task approach in the 4C/ID model, where the students moved from being mere users to designers who were able to design complex rule-bound systems with confidence. However, what emerged from the results is that though there was greater expertise in the latter phase, there was no direct impact on the DT scores. In conclusion, the 4C/ID model is successful in helping students internalize algorithmic logic and gain self-efficacy within PD education. Despite its success, it is strongly recommended that future learning models place more emphasis on accurate task calibration to cope with technical requirements while addressing creative design needs. The current study contributes to the existing theories because it illustrates the relevance of process-oriented learning approaches in architectural education.

Keywords: Parametric design education, 4C/ID model, Computational thinking, Algorithmic reasoning, Instructional design.

RESEARCH ARTICLE
Vol. 9, No. 1, 2026, 1-20
DOI: 10.59389/modular.1853822

INTRODUCTION

Parametric design (PD) marks a transition from form-driven architecture to a process-based mode of variables, rules, and relationships. Unlike conventional approaches to design, PD employs computational logics to drive the creation of form by interaction and parameter-based dynamics in terms of modeling rule sets to achieve flexibility and complexity (Çam & Özen, 2021). This improves students' analytical thinking skills and their capacity to analyze and react to complex situations (Alalouch, 2018). Therefore, instruction that focuses on technical skills alone in the learning process of parametric design is inadequate, and there is a need to place emphasis on parametric thinking as an instructional goal and paradigm (Alalouch, 2018; Oxman, 2008). This can only be achieved through learning models that deliver on the cognitive processes and methodological frameworks involved in systematic and algorithmic rule-based generation in design. From the cognitive perspective, the learning process of PD focuses on process-based problem-solving, abstraction, and analytical thinking (Uçar et al., 2023) and involves the ability to perform complex tasks through the integration of conceptual, procedural, and strategic knowledge (Frerejean et al., 2019). Methodologically, PD education draws on algorithmic and rule-based design approaches (Abdelmohsen & Massoud, 2021; Tan, 2024) that are conducive to sequential thinking, decision-making, and design creativity through generative exploration. Thus, PD pedagogy is an interdisciplinary approach to the development of higher-order thinking.

Recent studies focused on PD education find it a deep cognitive change, engaging higher cognitive load while developing analytical and strategic skills (Delahunty et al., 2020; Lee et al., 2013). On the cognitive dimension, it is shown that metacognitive strategies, defined as thinking about thinking, contribute to the generation and development of creative ideas in design (Kavousi et al., 2020). Strategic knowledge, encompassing the planning, monitoring, and revising of one’s cognition, is identified as a core constituent of (computer-aided design) CAD modeling skills, in contrast to other approaches, focusing on factual or procedural knowledge (Delahunty et al., 2020). In this regard, parametric design pedagogy not only promoted technical skills but also induced higher-order cognitive and metacognitive competencies through which students acquired an awareness of monitoring and improving their design thinking processes. Methodologically, a variety of pedagogical models are suggested, such as incomplete recipes designed to foster algorithmic thinking (Vazquez, 2024) and the translation of form model, which translates architectural forms into algorithmic procedures (Tan, 2024). Besides, hybrid strategies that integrate analytical deconstruction, manual sketching, and digital modeling invite students to analyze, plan, and assess their design process before actually touching computational environments (Uçar et al., 2023).

Because of its dual nature of cognitive and methodological objectives, the 4C/ID model is apt in this case, as it enables the integration of conceptual, strategic, and procedural thinking and reflection. This current study utilizes tested measurement instruments that capture the dimensions, since PD tasks engage inter-related cognitive processes like abstraction, decomposition, algorithmic reasoning, iterative exploration, and creative problem formulation. CT, CTSE, and DT scales allow for analytical reasoning and build learners' confidence in handling algorithmic and strategic problem-solving tasks. All these instruments, along with the measures of learning performance, such as final grades, provide a comprehensive means of evaluating the cognitive and attitudinal transformation expected in a 4C/ID-based parametric design course.

Research Problem

PD demands a paradigm shift in architecture design with respect to methods from tradition to logic-based approaches using algorithmic processes. This makes a boundary for architecture students because they have to understand complex visual programming software at the same time. There is no systematic

RESEARCH ARTICLE
Vol. 9, No. 1, 2026, 1-20
DOI: 10.59389/modular.1853822

learning process available in architecture education for bridging this domain with respect to developing advanced computational logic.

Research Aim

This study investigated the effectiveness of the four-component instructional design model, 4C/ID, in the context of a 15-week undergraduate PD course. The main purpose was to establish whether this task-centered whole-learning approach enhances students' learning performance in terms of final grades, computational thinking skills self-efficacy, and design thinking scores in view of the inherent complexities of visual programming environments.

Research Questions

On the basis of the above-stated foundation, the current study applies an instructional design based on 4C/ID in the PD course, and its effect is measured by pre- and post-testing using the scales. This mixed-methodology study allows for gaining an overall insight into the learning process. In this context, the research questions include:

- RQ1. How are students' final grades (learning performance) related to their computational thinking, computational thinking self-efficacy, and design thinking scores, as indicated by the correlation analysis?
- RQ2. To what extent do students show measurable changes in computational thinking and computational thinking self-efficacy over the duration of the course, as evidenced by pre–post (t-test) comparisons?
- RQ3. What are students' perceptions of the effectiveness of the 4C/ID-based course structure, the level of task complexity, and the sufficiency of instructional support in developing their parametric design competencies?

THEORETICAL BACKGROUND

To fulfill the complex demands of the digital era (Tan, 2024), in the digital age, the teaching of architecture needs to break free from the limitations of the objectives-based model. The 4C/ID model of learning provides an even better task-centered framework (van Merriënboer & Kirschner, 2012) for parametric designs in architecture by considering the concept of cognitive loads using the strategy of whole task learning. This will enable the learners to participate in the process of abstraction that computational thinking requires.

The 4C/ID-Model

Well-designed programs must consider cognitive architecture to ensure efficient learning (van Merriënboer & Kester, 2012). The 4C/ID model addresses this by integrating four core elements (Frerejean et al., 2019; Vandewaetere et al., 2015) (Figure 1):

1. Learning tasks (LT) are authentic, real-life problems forming the backbone of complex learning. To ensure transfer, tasks must offer high variability across diverse scenarios (Janesarvatan & Van Rosmalen, 2023). They are organized into simple-to-complex task classes, guided by learning needs and the intended delivery mode (van Merriënboer & Kirschner, 2012).

2. Supportive information (SI) enables the successful completion of non-routine tasks by bridging conceptual knowledge and practice (Janesarvatan & Van Rosmalen, 2023). Unlike procedural steps, it

RESEARCH ARTICLE
Vol. 9, No. 1, 2026, 1-20
DOI: 10.59389/modular.1853822

offers domain-related guidance (Limbu et al., 2018) through general mental models, illustrations (e.g., case studies), and cognitive feedback (van Merriënboer & Kirschner, 2012).

3. Procedural information (PI) facilitates schema automation for routine tasks (Janesarvatan & Van Rosmalen, 2023) by providing just-in-time (JIT) rules, demonstrations, and corrective feedback during execution (van Merriënboer & Kirschner, 2012).

4. Part-task practice (PtP) ensures the automation of recurrent task aspects through targeted, repetitive practice of specific sub-skills (van Merriënboer & Kirschner, 2012).

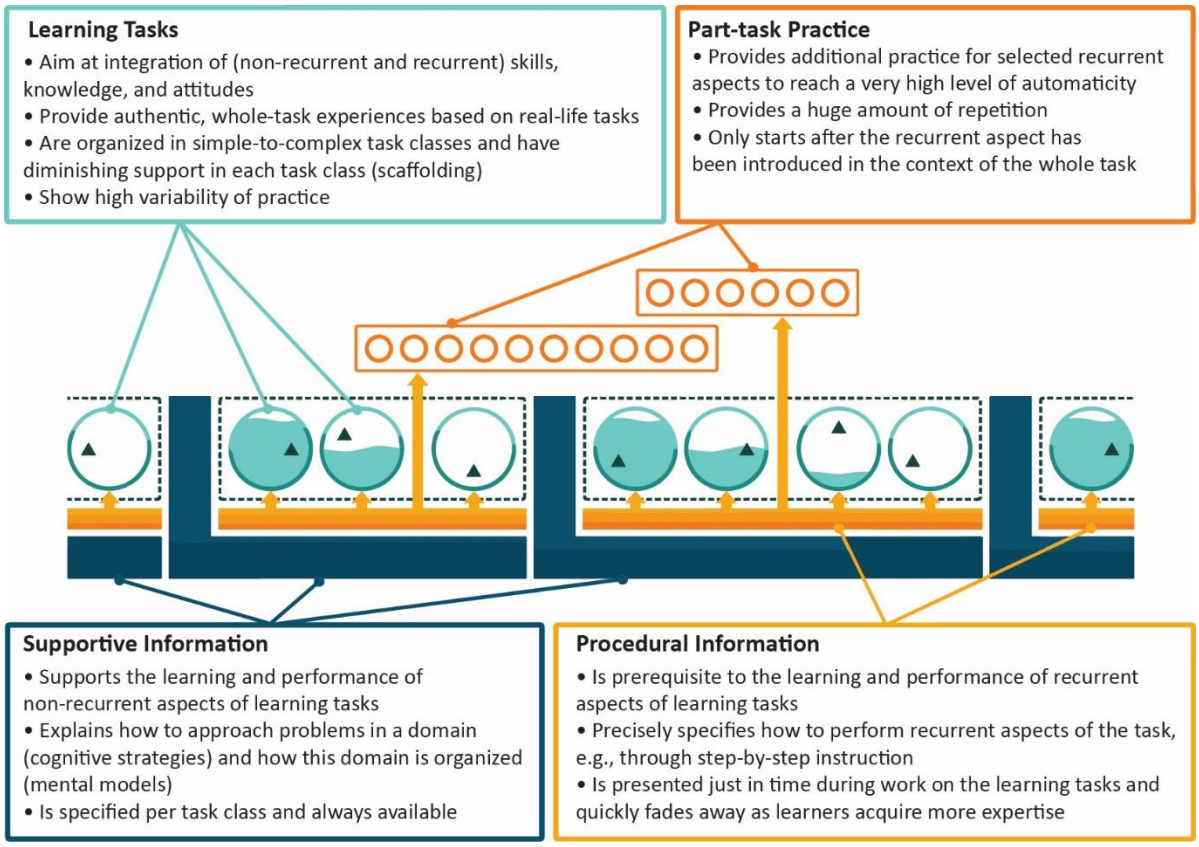


Figure 1. The 4C/ID's components (van Merriënboer & Kirschner, 2012)

Parametric Design Education through the 4C/ID Components

Parametric Design (PD), which is defined by algorithmic logic, is an intricate cognitive field that needs coordination between geometry, programming, and design thinking. In an attempt to manage these complexities, 4C/ID has developed four components.

For PD, LTs need to be authentic; students should be compelled to define the complete generative logic(rule-set and relationships) for an architectural design, not just for isolated components (Alalouch, 2018). Tasks should be progressively complex, adding in factors such as the structural properties or environmental inputs. Authentic examples include developing generative models (Alalouch, 2018), or translating some constraints from physical exploration into the digital parametric model (Abdelmohsen & Massoud, 2021).

RESEARCH ARTICLE
Vol. 9, No. 1, 2026, 1-20
DOI: 10.59389/modular.1853822

SI plays an essential role in inspiring an algorithmic mindset. This involves the education of theoretical domain models, such as viewing architecture as a dynamic rule-based system (Alalouch, 2018), systematically employing the method of reducing a form. For example, students would be able to study the geometric reduction technique methodologically through the study of practical architectural examples (Ostrowska-Wawryniuk et al., 2022). This enables the formalization of the algorithmic logic involved.

PI targets fluency in software manipulation and translating abstract thought into code. For PD, PI is the skill in manipulating software and the ability to transform thinking into code. An effective instructional material can be the uncompleted recipe, which provides visual hints or a step-by-step guide on how to build a defined parametric models in a visual programming languages (VPL) on Grasshopper (Vazquez, 2024). PI also involves teaching the students the rules to apply the rules set using simple mathematical operations. Finally, the translation of the form (ToF) model acts as a procedure, where one is expected to reverse-engineer a given form in architecture to discover the base geometry and the rules involved in the transformations (Tan, 2024).

PtP ensures automaticity in core digital skills when whole tasks provide insufficient repetition. By automating routine actions such as specific node usage or debugging, students can dedicate cognitive resources to complex, non-recurrent problem-solving aspects.

METHODOLOGY

The study was carried out using a structured four-phase approach to ensure clarity and replicability. In the first phase (preliminary assessment), pre-test data were gathered using the CT, CTSE, and DT scales to set the baseline for the 14 senior architecture students. The second phase (implementation) lasted for 15 weeks, during which the 4C/ID framework curriculum was implemented through foundational, systemic, and narrative modules as described in Table 1. In the third phase (evaluation), post-test data and final project results were collected to assess cognitive and performance changes. Finally, in the fourth phase (data analysis), the statistical analysis of quantitative data was carried out using t-tests and correlation analysis, in addition to the coding of qualitative data.

Design Course Along with the 4C/ID Model

The course was developed entirely in line with the 4C/ID model, and the curriculum was organized in the learning cycle of Prior Knowledge → Supportive Information → Procedural Information → Learning Tasks → Task Outcomes. The 15-week course allows for authentic whole-task experience approaches supported by conceptual guidance, procedural support, and targeted practice (Table 1).

Table 1. A simplified training blueprint

Week	Prior Knowledge	Supportive Information	Procedural Information	Learning Objective	Learning Task (LT/PTP)	Task Outcome	Post-Class Activity	Assignment
1–3 (Foundational)	Basic Geometric Logic	Node-based design; Parameter exploration; List logic	Interface basics; Series & List manipulation	Understand transformations & automated data selection	LT1.1-1.2: Tower Model; PTP1: List practices	Parametric tower; List fluency	Video 01–04; Self-quiz	A1–A3: Basic Geometric Maipulation and list usage
4–6 (Systemic)	List & Debugging skills	Data mapping; Hierarchy;	Graph Mapper; Multi-object	Design responsive multi-object	LT2.1-2.2: Kinetic Façade;	Responsive kinetic	Video 05–07; Reflectio	A4–A7: Kinetic Explore

RESEARCH ARTICLE
Vol. 9, No. 1, 2026, 1-20
DOI: 10.59389/modular.1853822

Week	Prior Knowledge	Supportive Information	Procedural Information	Learning Objective	Learning Task (LT/PTP)	Task Outcome	Post-Class Activity	Assignment
		Remap logic	integration	kinetic systems	PTP2: System Refinement	surface	n	
7 (Midterm)	Surface Logic	—	—	Analyze holistic movement & function	Midterm Project: Transformable Surface	50x70cm Board + Grasshopper (GH) Model	Expert review	Midterm Submission
8–10 (Motion)	Motion Basics	Temporal design; Arch. motion; Spatial composition	Sliders; C# basics; Export pipeline	Transfer modeling to synchronized visual output	LT3.1-3.2: Animation; PTP3: Export & Sync	Linked GH-Premiere montage	Video 08–11; Checklist	A8–A10: Montage Prep.
11–12 (Narrative)	Editing Principles	Visual storytelling; Rhythm; Coherence	Media sync; Timeline setup	Apply pacing principles to visual narratives	LT4.1-4.2: Timeline & Montage refinement	Polished coherent draft	Revision based on feedback	A11–A12: Final Edits
13	Midterm (Anim)	—	—	Analyze time & sequencing intent.	Final Project: Parametric Video Montage.	30–45 sec narrative video.	Submit for storytelling review.	Animation Video
14-15 (Final Project)	Narrative Intent	Reflection skills; Iterative evaluation	Final sync & polishing	Communicate design intent through cinematic flow	Final Project: Parametric Video Montage	45-sec Video + Reflective Report	Final Presentation; Self-assess	Final Submission

In accordance with the 4C/ID principle of whole task progression, the design gradually proceeds from geometric manipulation through kinetic systems to finally arrive at the synthesis of the video. This enables the students to acquire experience with the task of geometric logic, motion control, and digital communication. Learning tasks are designed based on authentic architectural design scenarios. These reflect the 4C/ID model's focus on whole-task learning professional practices. These tasks further help to strengthen the connection between conceptual understanding and application, which is a reflection of actual practices in computational design. The course also requires the application of the learned concepts in different modeling contexts (geometric, animation, and narrative). Applying concepts across different domains aids in generalizing cognitive strategies, and this further assists in establishing flexible competencies in the domain. These competencies are also established in a manner that is relevant to professionalism within the context of computational design.

RESEARCH ARTICLE
Vol. 9, No. 1, 2026, 1-20
DOI: 10.59389/modular.1853822

4C/ID Components in the Parametric Course Design

The components of the 4C/ID-based parametric design course designed for the lesson are shown in Figure 2.

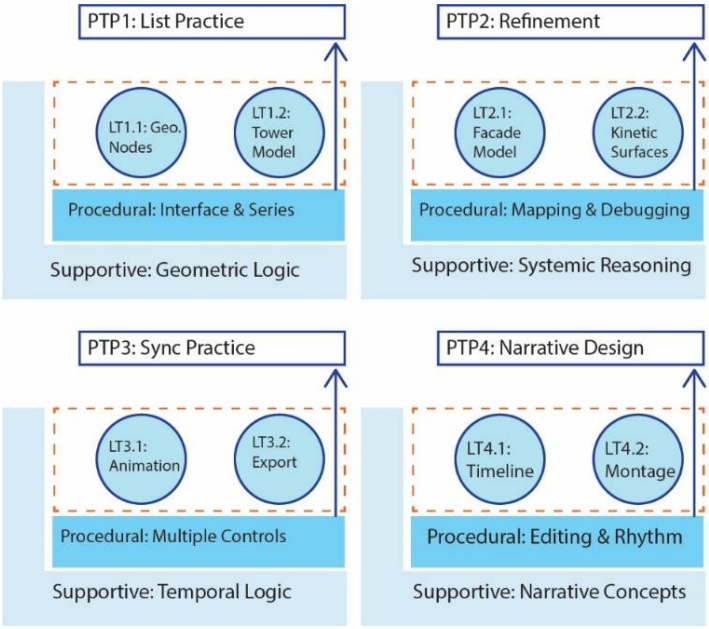


Figure 2. Structural distribution of parametric design education tasks according to the 4C/ID model

Learning Tasks

The learning tasks in the course follow a structured progression from foundational modeling challenges to advanced parametric–temporal communication, in full accordance with the 4C/ID principle of whole-task sequencing.

- *Foundational cluster (weeks 1–3):* Tasks LT1.1–LT1.2 introduce students to basic geometric logic, node manipulation, and Series-based generative structures, culminating in the construction of a parametric tower.
- *Systemic cluster (weeks 4–6):* Tasks LT2.1–LT2.2 advance into data-driven façade generation and responsive kinetic systems, asking students to reconstruct façade logic using list hierarchies and design a multi-object kinetic surface inspired by real architectural precedents.
- *Motion cluster (weeks 8–10):* After establishing core modeling competencies, tasks LT3.1–LT3.2 center on motion-based tasks. Students develop multi-surface animations through slider integration and C# scripting, and subsequently transfer animated sequences into Premiere for editing.
- *Narrative cluster (weeks 11–12):* The final cluster (LT4.1–LT4.2 and final Project) focuses on multi-layer narrative construction, prompting students to build synchronized timelines, refine montage structures, and produce coherent visual narratives.

These thematically grouped tasks mirror authentic computational design workflows and increase in complexity, ensuring deep, integrated learning across the semester.

RESEARCH ARTICLE
Vol. 9, No. 1, 2026, 1-20
DOI: 10.59389/modular.1853822

Supportive Information

Supportive Information in the course is structured around three conceptual layers that align with 4C/ID's emphasis on non-recurrent knowledge and mental-model formation.

- *Conceptual foundations layer (weeks 1–3):* The course provides the initial conceptual introduction of geometric logic, node modeling, data hierarchies, mapping, and strategic list and curve use. This layer of the course provides the necessary conceptual foundation of the mental models required to comprehend the interactions between geometric relationships, parametric constraints, and data structures in Rhino-Grasshopper.
- *Behavioral and systemic reasoning layer (weeks 4–6):* When the tasks progress into dynamic and kinetic systems, supportive information incorporates systemic thinking in the following fields of holistic surface activity, interaction of multiple objects, normalization approaches, logic of remapping, and precedents of kinetic and multifunctional architectural systems.
- *Temporal and narrative concepts layer (weeks 8–12):* As students engage in animation and montage exercises, supportive information shifts to temporal reasoning, including frame-rate logic, spatial storytelling, camera path theory, cinematic flow, rhythm, and narrative sequencing. This layer helps students develop high-level mental strategies for constructing time-based representations that communicate design intent effectively.

Procedural Information

Procedural information is provided in a structured progression of recurrent, rule-based, and just-in-time procedures to ensure students are able to execute operational tasks with increasing efficiency.

1. *Foundational operations layer (weeks 1–3):* Students are given clear procedures for constructing geometric primitives, performing transformations (moving, rotating, scaling), performing series operations, indexing list data, and manipulating graph mapper functions. These are introduced just-in-time, at the exact moment when the learner needs them, which is helpful to minimize extraneous cognitive load during initial skill acquisition.

2. *Integration and debugging layer (weeks 4–6):* As tasks become increasingly complex, procedural information must address the integration of multiple components: the combination of series, list, and remap operations, debugging faulty definitions, and coordinating multi-axis transformations.

3. *Temporal and media-editing operations layer (weeks 8–12):* At this level, procedural guidance covers the animation workflow, including slider grouping, C# scripting for time-based behaviors, and frame export processes. In the creation of montage, explicit steps about PI are provided for the importation of sequences, synchronization of media layers, adjustment of timing, and fixing timeline inconsistencies.

Part-Task Practice (PTP)

Part-task practice is implemented in the learning process at four distinct points in order to ensure automaticity of basic skills.

- *Geometric and list fluency (week 3):* Prior to the complex phase of data management, students work on some rapid drills of selection of list items, list length, and geometric constraints to

RESEARCH ARTICLE
Vol. 9, No. 1, 2026, 1-20
DOI: 10.59389/modular.1853822

automate frequent data manipulations and free up resources during subsequent kinetic façade tasks.

- *System refinement (week 6):* Before the midterm, this class asks students to revisit and extend the earlier exercises on rotations and remapping in a refinement loop for solidifying the synthesis of previous methods into coherent dynamic systems.
- *Export and sync automaticity (week 10): To* prevent technical barriers from hindering creative flow during the montage phase, focused drills on batch exporting frames and synchronizing audio-visual cuts are introduced. This ensures that the technical pipeline between Rhino-Grasshopper and Adobe-Premiere becomes error-free and automatic.
- *Final polish (week 14):* This final practice session fine-tunes montage structure, sound synchronization, and timing adjustments in order to guarantee that students have professional fluency in editing operations before the final submission.

Reserving the application of part-task practice for those tasks that do in fact require repetition in order to reach mastery (e.g., data handling activities, media export tasks) will conform to the 4C/ID frame for the application of PTP, which should enable high-frequency tasks that need to become automatic.

Course Assignment and Participants

There were fourteen students in the course, who were all seniors pursuing architecture with similar levels of educational backgrounds and design experiences. Assignment 1 required the design of a multifunctional parametric system based on motion. This required the students to design a continuous surface that changes to act as seating, storage, and lighting. For this, students were required to produce a Rhino-Grasshopper board with dimensions of 50x70cm. The output required perspective views that would demonstrate the adaptability of the system and the process of how it has been designed procedurally. For the final project, students were required to produce an animated visual story based on the motion of the system. Students created videos that showcased the motion of the system, its transitions, and its aesthetic changes based on camera movements in space and time, along with the presentation board showing the final product designed in the system. The montage had to fulfill particular requirements. First of all, it had to demonstrate seamless transitions from various functional states (from seating to storage and from storage to lighting); secondly, it had to express global transformations instead of component-level transformations; and lastly, it had to express design logic in terms of synchronized motion and cinematic storytelling. Figure 3 shows the student work from the above-mentioned assignment.

Data Collection

To evaluate the effectiveness of the 4C/ID-based parametric design course, four instruments such as computational thinking (CT), design thinking (DT), computational thinking self-efficacy (CTSE) and learning performance via the consensual assessment technique (CAT), were employed to capture the cognitive and affective dimensions. The selected scales are conceptually aligned with the cognitive processes activated by the course's learning tasks. Wing (2006) defined CT as a problem-solving activity involving abstraction, whereby it is a skill that is very applicable to design-based fields. Beyond computer science, Henderson et al. (2007) argued about its quintessential nature within a variety of fields. Although it has no general definition, programming has been generally recognized as the key paradigm for cultivating CT, which is based on algorithmic logic (Buitrago Flórez et al., 2017). CT in this study was measured by a modified form of a scale by Korkmaz & Bai, 2019, which consists of 20 items that were measured by five dimensions: evaluation, abstraction, decomposition, generalization, and algorithmic thinking. Moreover, the ethical approval for conducting this study has been obtained from the Yildiz Technical University Social and Human Sciences Research Ethics Committee (Meeting Date: 03.05.2025, Decision No: 2025.05).

RESEARCH ARTICLE
Vol. 9, No. 1, 2026, 1-20
DOI: 10.59389/modular.1853822

DT has originated in professional circles and has since developed into one of the main learning approaches in dealing with many complex problems. It can act as both a mindset and methodological processes, emphasizing empathy, ideation, and iterative testing (Brenner et al., 2016; Razzouk & Shute, 2012). According to the model developed by the Stanford School, there should be five main stages involved in DT—empathize, define, ideate, prototype, and test. This model has been extensively applied in learning environments in order to measure creative problem-solving capabilities of learners in varied learning situations (Tsai & Wang, 2021). To evaluate students' DT, the present research employed an adapted version of the DT disposition scale developed by Tsai & Wang, 2021, which corresponds to these phases and includes four subscales (empathy, define, ideate, and prototype) comprising 19 items. In addition, the CTSE test, as designed by Kukul & Karatas, 2019's CTSE, which has 18 items classified across four domains: reasoning, abstraction, decomposition, and generalization, was also integrated. Furthermore, the four-item data on self-confidence, as indicated by Li et al. (2023) were also collected. Both scales have five points for the data, which range from strongly disagree to strongly agree.



Figure 3. Examples of student assignments

To measure domain-specific design performance, the consensual assessment technique (CAT), originally developed by Amabile (1982), was utilized. Learning performance for the tasks was assessed using criteria

RESEARCH ARTICLE
Vol. 9, No. 1, 2026, 1-20
DOI: 10.59389/modular.1853822

adapted from prior design studies (Cho & Suh, 2019; Suh & Cho, 2020). Learning performances for three tasks were represented in terms of 3D richness**,** originality, appropriateness, understandability, and overall creativity (Cho & Suh, 2019). Since robust design outcomes typically require both novelty and utility (Runco, 1988), originality and appropriateness served as key evaluation criteria. Additionally, given the requirement to produce a 3D object, 3D spatial richness was included. Finally, the comprehensibility of the product was measured through understandability.

Two expert evaluators assessed the design outputs based on five specific dimensions, which were presented in a randomized order to prevent bias. The first dimension was originality, wherein innovation referred to the level to which the design solution presented a novel, unpredictable, and/or inventive solution to the problem, notably evident in a distinct 3D design pattern. Secondly, there was appropriateness, which assessed how well the design solution fully addressed every programmatic and functional requirement. Third came 3D spatial richness, which centered on the spatial complexity inherent in the design in relation to the size, volumes, and differences in the depth, width, and height in three dimensions, respectively. Follows was the dimension, understandability, which measured the level to which the design solution presented easily comprehensible rules and processes with respect to the solution and its graphical presentation, respectively. Lastly, overall performance encompasses the collective subjective design evaluation on the part of the expert evaluator, in line with the principles employed in the CAT method, where the evaluators independently assessed the designs produced by the students on a 5-point Likert scale, with 1 indicating poor performance and 5 a performance graded as *excellent* respectively.

To ensure the reliability of the evaluation process, the two expert evaluators participated in a formal training and calibration session before conducting the CAT. During this session, the raters reviewed a standardized rubric defining the five dimensions. To ensure that the two raters used the same benchmarks in scoring, the raters were required to score three example projects independently and then compare the scores to reach a consensus on the levels of performance. After the calibration process, the raters were required to score the final projects of the students independently using a 5-point Likert scale. The inter-rater reliability was measured using the Intraclass Correlation Coefficient (ICC). The analysis gave an ICC of 0.953 ($p < .001$), which is an excellent level of agreement between the two raters.

In order to address these quantitative aspects, there had to be other components in place to collect qualitative information on behalf of students and with respect to the third research question: *what are students' perceptions of the effectiveness of 4C/ID-based course structure, level of task complexity, and sufficiency of support provided in order to develop students' parametric design skills?.* The qualitative method used would allow for an understanding of cognitive and affective aspects in learning in a specified course structure design.

FINDINGS

Pearson correlation coefficients were calculated to examine the relationship between final grade and the post-instruction scores on all subscales. As shown in Table 2, final grade demonstrated significant positive correlations with several computational thinking (CT) dimensions, including algorithmic thinking ($r = .652$, $p = .012$) and problem Solving ($r = .551$, $p = .041$). Additional CT components which are generalization ($r = .567$, $p = .034$), reasoning ($r = .531$, $p = .051$), and abstraction ($r = .524$, $p = .054$) also showed moderate correlations, some marginally significant. The total CT score approached significance ($r = .522$, $p = .055$).

For computational thinking self-efficacy (CTSE), the total score was significantly correlated with final grade ($r = .549$, $p = .042$), indicating that students with higher computational self-confidence tended to achieve higher performance in the course. In contrast, none of the Design Thinking (DT) subscales (Empathize,

RESEARCH ARTICLE
Vol. 9, No. 1, 2026, 1-20
DOI: 10.59389/modular.1853822

Define, Ideate, Prototype) nor the DT total score exhibited significant correlations with Final Grade (all p > .62). This pattern suggests that course performance was most closely associated with technical and computational constructs rather than with higher-level design-strategic dispositions.

Table 2. Correlation analysis between final grade and other factors

Variable (After)	r	p	95% CI (Lower–Upper)	Fisher's z	SE(z)
Algorithmic Thinking	0.652	0.012	0.185 – 0.879	0.778	0.302
Problem Solving	0.551	0.041	0.029 – 0.837	0.62	0.302
Creativity	0.4	0.156	–0.166 – 0.768	0.424	0.302
Cooperativity	–0.035	0.904	–0.556 – 0.505	–0.035	0.302
Critical Thinking	0.477	0.084	–0.071 – 0.804	0.52	0.302
CT Total	0.522	0.055	–0.012 – 0.824	0.579	0.302
Generalization	0.567	0.034	0.053 – 0.844	0.644	0.302
Abstraction	0.524	0.054	–0.009 – 0.825	0.582	0.302
Decomposition	0.321	0.263	–0.253 – 0.728	0.333	0.302
Reasoning	0.531	0.051	0.0005 – 0.828	0.591	0.302
CTSE Total	0.549	0.042	0.025 – 0.836	0.616	0.302
Empathize	0.056	0.85	–0.489 – 0.570	0.056	0.302
Ideate	0.09	0.758	–0.462 – 0.593	0.091	0.302
Define	0.116	0.692	–0.442 – 0.609	0.117	0.302
Prototype	0.142	0.629	–0.421 – 0.625	0.143	0.302
DT Total	0.118	0.688	–0.440 – 0.610	0.119	0.302
CI : Confidence Interval, SE : Standard Error					

Paired-sample t-tests (Table 3) were conducted to assess changes from pre- to post-instruction across CT, CTSE, and DT subscales. Results showed substantial improvements in several CT dimensions. Algorithmic thinking increased significantly from pre (M = 2.80, SD = 0.64) to post (M = 3.84, SD = 0.79), t(13) = −5.65, p < .001, d = −1.43, and problem solving also improved significantly, t(13) = −3.54, p = .004, d = −1.04. The CT general score showed a trend toward significance, t(13) = −2.12, p = .054. Even greater changes were observed in computational thinking self-efficacy (CTSE). All CTSE subdimensions showed statistically significant improvements accompanied by large effect sizes. Students' abstraction scores increased markedly, $t(13) = -3.53$, $p = .004$, $d = -0.95$, while reasoning demonstrated an even stronger gain, $t(13) = -4.25$, $p < .001$, $d = -1.14$. Significant improvements were also found in decomposition, $t(13) = -4.21$, $p = .001$, $d = -1.12$, and generalization, $t(13) = -3.20$, $p = .007$, $d = -0.86$. Results of the CTSE total score revealed the overall strength of the pattern and indicated a highly significant positive change, t(13) = −4.32, p < .001, d = −1.16. Taken together, these findings clearly demonstrate that students' confidence and feeling of competence regarding computational and algorithmic work had significantly improved throughout the duration of the course. In contrast, design thinking scores were found to be statistically equivalent for all subdimensions (empathize, define, ideate, prototype) and the DT total score (all p > .38). Design thinking scores were very stable from the beginning to the end of the test.

RESEARCH ARTICLE
Vol. 9, No. 1, 2026, 1-20
DOI: 10.59389/modular.1853822

Table 3. Paired sample t test results

Variable	Before Mean (SD)	After Mean (SD)	t	df	p	Cohen's d	95% CI
Computational Thinking Factors							
Creativity	4.38 (0.47)	4.17 (0.99)	0.77	13	0.456	0.27	−0.27, 0.80
Algorithmic Thinking	2.80 (0.64)	3.84 (0.79)	−5.65	13	< .001	−1.43	−2.17, −0.66
Cooperativity	3.75 (0.92)	3.59 (0.86)	0.78	13	0.449	0.18	−0.35, 0.71
Critical Thinking	4.14 (0.56)	3.93 (0.90)	1.19	13	0.254	0.26	−0.28, 0.79
Problem Solving	2.89 (0.59)	3.50 (0.60)	−3.54	13	0.004	−1.04	−1.68, −0.37
CT (General)	3.59 (0.39)	3.80 (0.63)	−2.12	13	0.054	−0.33	−0.86, 0.21
Computational Thinking Self Efficacy Factors							
Abstraction	2.94 (0.41)	3.91 (0.71)	−3.53	13	0.004	−0.95	−1.57, −0.30
Reasoning	3.07 (0.50)	3.96 (0.61)	−4.25	13	< .001	−1.14	−1.80, −0.44
Decomposition	2.59 (0.36)	3.68 (0.95)	−4.21	13	0.001	−1.12	−1.79, −0.44
Generalization	3.48 (0.55)	4.16 (0.84)	−3.20	13	0.007	−0.86	−1.46, −0.23
CTSA	3.02 (0.28)	3.93 (0.67)	−4.32	13	< .001	−1.16	−1.83, −0.46
Design Thinking (DT) Factors							
Empathize	4.48 (0.53)	4.48 (0.77)	0	13	0.998	–	−0.53, 0.52
Define	4.37 (0.52)	4.34 (0.72)	0.13	13	0.895	0.04	−0.49, 0.56
Ideate	4.23 (0.77)	4.00 (0.78)	0.91	13	0.38	0.24	−0.29, 0.77
Prototype	3.89 (0.80)	3.93 (0.94)	0.25	13	0.807	0.07	−0.59, 0.46
Design Thinking (DT)	4.24 (0.49)	4.19 (0.70)	0.31	13	0.766	0.08	−0.45, 0.61

DISCUSSION

Correlation analysis for RQ1 indicates that students' performance is most strongly correlated with algorithmic thinking (r = .652, p = .012), problem solving (r = .551, p = .041), and CTSE total scores (r = .549, p = .042). Moreover, the CT dimensions of generalization, abstraction, and reasoning were found to have moderate-to-strong correlations with success. It can be said that these findings confirm the 4C/ID model's focus on rule-based logic, which enables a transition from passive software use to active logic construction. This transition is in line with the parametric design pedagogy; more specifically, it confirms Vazquez's (2024) emphasis on rule-based construction and resonates with Tan's (2024) findings on the cognitive dimension of form translation via coding as a critical success factor in high-complexity environments such as Rhino-Grasshopper.

From an instructional design perspective, these correlation results further illustrate the importance that the current course-design and evaluation structure places more heavily on computational logic, procedural fluency, and technical problem-solving processes, compared to design strategic processes. Though correlation results can never truly measure causation, it nevertheless fits well with what kinds of performance-related factors would be considered to highly impact higher-order, computationally based processes as described by the 4C/ID model.

The pre-post test findings for RQ2 indicate that the instructional treatment resulted in positive developments on particular core aspects of computational thinking (CT) instead of a general positive

RESEARCH ARTICLE
Vol. 9, No. 1, 2026, 1-20
DOI: 10.59389/modular.1853822

development on all aspects. More specifically, the findings indicate that large positive effects were observed on algorithmic thinking ($p < .001$, $d = -1.43$) and problem-solving ($p = .004$, $d = -1.04$). These particular developments may indicate that the 4C/ID design activities support learners in dealing with complex logical tasks. Nevertheless, it should be noted that other aspects of CT, such as creativity and critical thinking, did not present statistically significant positive developments, which may indicate that these aspects need more extensive treatment periods or alternative instructional strategies.

The strongest positive trend across the whole category was found in CTSE, where all subscales showed significant improvement ($p < .01$, $d = -0.86$ to -1.16). The broad improvement in self-efficacy beliefs might be a consequence of the model's focus on JIT information and PtP practice. The model seems to have contributed to overcoming negative affective barriers by allowing students to learn routine procedures through repetition, which might have facilitated a shift from survival mode to a more confident approach to parametric modeling. Although these findings are promising, they should not be considered as preliminary proof of the model's universal effectiveness but rather as preliminary evidence of its effectiveness in this particular setting.

Analysis of the correlation findings (RQ1) and pre-post gains (RQ2) indicates a possible empirical trend: those variables like algorithmic thinking, computational problem-solving, and technical self-efficacy, which seemed most directly related to the final outcome, also seemed to demonstrate the greatest progress. These results could suggest that the course structure was likely conducive to the skills necessary for complex parametric modeling. Moreover, these results seem to support the hypothesized mechanisms of the 4C/ID model. For example, authentic learning tasks may be the catalyst for improvements in algorithmic thinking, while supportive information could be the facilitator of the underlying conceptual shifts in abstraction. Likewise, procedural information may boost efficacy perceptions via directed learning, while part-task practice is likely integral to the processes of decomposition and generalization via the repetition of recurrent skills. However, although the large effect sizes discovered in the CT and CTSE provide some initial validation of the model, the small sample size ($n=14$) is still a notable consideration. This may restrict the power of the study and its generalizability to a wider student base. Therefore, perhaps rather than providing conclusive evidence of the universal applicability of the model in design education, the current study could be considered more of a case study of senior architecture students.

The result that DT scores did not show significant change or correlation with the final grade was as expected in the initial projections of the study. Although DT involves high-level competencies like empathy, strategic problem framing, and prototyping, these creative mindsets can usually take long maturation times or low-load situations that are not easily accommodated in a 15-week technical subject offering. Notably, the findings indicate a competition for cognitive resources where the design of the course inadvertently gave higher priority to algorithmic correctness over design strategic exploration. As Li et al. (2023) suggest, CT and DT are parallel but distinct systems that do not necessarily support each other in high-complexity situations. The qualitative data about the Grasshopper interface indicates a high level of intrinsic cognitive load that probably drained the students' mental resources, leaving insufficient capacity for the empathy-driven ideation needed in DT. This mental overload led to a necessity to focus on getting the code right (algorithmic correctness), which in effect limited the fluid ideation and user empathy required for DT development. Thus, these results imply that future scaffolding in instruction needs to move beyond the mere provision of procedural information. For DT development and technical proficiency to be achieved, future iterations of the course need to include modular scaffolding in the LT, such as pre-existing code clusters.

Student feedback offers a crucial qualitative perspective on the quantitative trends identified in RQ1 and RQ2, providing a more complex understanding of the role of instructional features in improving

RESEARCH ARTICLE
Vol. 9, No. 1, 2026, 1-20
DOI: 10.59389/modular.1853822

performance. Although the statistical results clearly indicate substantial improvement, the qualitative narratives underscore a striking imbalance in students' initial computational experience. This imbalance indicates that the correlations identified in RQ1 were probably mediated by students with a stronger initial foundation, enabling them to better capitalize on the 4C/ID framework. Nevertheless, a striking paradox arose: although the 4C/ID framework was rated as conceptually clear, the implementation of parametric tasks was often experienced as overwhelming. This suggests that, despite the whole-task approach's focus on authenticity (van Merriënboer & Kirschner, 2012), the sheer number of concurrent operations in Rhino-Grasshopper might have come close to the limits of students' working memory capacity. The *spaghetti problem*, or the visual complexity of dense node definitions (Harding & Shepherd, 2017; Peteinarelis & Yiannoudes, 2018), was often mentioned as a hindrance, suggesting that the current Supportive Information (SI) and Procedural Information (PI) should be more carefully focused on basic data hierarchy and parametric logic to avoid cognitive overload.

Based on these results, future studies should continue to build upon the implementation of the 4C/ID model and investigate more adaptive scaffolding techniques. One of the key recommendations is to incorporate modular scaffolding, which would allow students to be given documented code clusters for routine tasks. This would enable students to avoid the low-level *data-wiring* tasks and concentrate on the high-level *narrative logic* tasks, which could potentially create the cognitive space needed for Design Thinking (DT) techniques to flourish. Moreover, in light of the competition for cognitive resources, future studies should investigate the use of AI-based scaffolding.

For example, the application of large language models or AI plugins may offer real-time, just-in-time procedural advice, assisting in de-cluttering complex definitions and illustrating data structures as students develop them. Long-term, it is imperative to monitor whether the achievement of technical proficiency ultimately results in a *plateau* where cognitive load is reduced sufficiently to permit creative inquiry in subsequent semesters. Lastly, increasing the sample pool from the current population (n=14) through multi-institutional research would be vital in confirming these observations across a wider population, assisting in ascertaining whether the 4C/ID model can be codified as a universal paradigm for parametric design education or whether it necessitates certain modifications according to the level of architectural education.

CONCLUSION

The application of the 4C/ID model in parametric design education represents a paradigm shift from tool-centered learning to a logic-based learning paradigm. Although traditional learning approaches have difficulty dealing with the high cognitive load of visual programming, this study showed that a task-centered and whole-task approach can provide the necessary technical skills for students to move from passive software users to active designers of rule-based systems.

The results indicate that in the digital era, architectural pedagogy needs to focus on procedural fluency and algorithmic thinking as the preconditions for creative exploration. The high correlation between course success and the dimensions of CT indicates that understanding the logic of the form is the first hurdle to overcome in order to be competent in high-complexity domains such as Rhino-Grasshopper. However, the stability in the scores of DT highlights an important pedagogical point that technical competence and creative maturity have different time constants. Therefore, curricula should be designed to first stabilize technical self-efficacy before expecting higher-order design synthesis.

The implication of this research is that there is a need for an adaptation of the 4C/ID model in design-related fields to address the spaghetti problem and the cognitive overload problem that comes with visual

programming. The use of pre-documented code clusters for repetitive tasks, or modular scaffolding, is believed to be a critical component in addressing this problem. This is because instructors may give students the cognitive space they need to engage with the narrative and spatial parts of a task by solving the data wiring problem.

The area of integrating new technologies presents a very promising area of further research. In particular, the application of AI-based scaffolding and LLMs could be investigated as a means of providing just-in-time procedural guidance, possibly uncluttering complex definitions in real-time. Moreover, although this case study represents initial validation for senior architecture students, inter-institutional research with a larger sample size is necessary to ascertain whether the 4C/ID model can be formalized as a paradigm for architectural education in general.

Author Contribution Statement

Order	Name Surname	ORCID	Contribution to Writing*
1	Hasan TAŞTAN	0000-0001-6231-4388	1, 2, 3, 4, 5
* The number(s) corresponding to the relevant explanation in the contribution section have been written.			
1. Designing the study 2. Collecting the data 3. Analysis and interpretation of the data 4. Writing the article 5. Critical revision			

Ethics Committee Approval

This research was carried out with the permission of the Yıldız Technical University Rectorate Social and Humanities Scientific Research and Publication Ethics Board dated May, 03, 2025 and numbered 2025-05.

Conflict of Interest Statement

There is no personal and/or financial conflict of interest within the scope of the study.

REFERENCES

Abdelmohsen, S., & Massoud, P. (2021). A material-based computation framework for parametric design education. *Open House International*, *46*(3), 459–475. https://doi.org/10.1108/OHI-02-2021-0043

Alalouch, C. (2018). A pedagogical approach to integrate parametric thinking in early design studios. *Archnet-IJAR: International Journal of Architectural Research*, *12*(2), 162–181. https://doi.org/10.26687/archnet-ijar.v12i2.1584

Amabile, T. M. (1982). Social psychology of creativity: A consensual assessment technique. *Journal of Personality and Social Psychology*, *43*(5), 997. https://doi.org/10.1037/0022-3514.43.5.997

Brenner, W., Uebernickel, F., & Abrell, T. (2016). Design thinking as mindset, process, and toolbox: Experiences from research and teaching at the University of St. Gallen. In W. Brenner & F. Uebernickel (Eds.), *Design thinking for innovation: Research and practice* (pp. 3–21). Springer. https://doi.org/10.1007/978-3-319-26100-3

RESEARCH ARTICLE
Vol. 9, No. 1, 2026, 1-20
DOI: 10.59389/modular.1853822

Buitrago Flórez, F., Casallas, R., Hernández, M., Reyes, A., Restrepo, S., & Danies, G. (2017). Changing a generation's way of thinking: Teaching computational thinking through programming. *Review of Educational Research*, *87*(4), 834–860. https://doi.org/10.3102/0034654317710096

Cho, J. Y., & Suh, J. (2019). Understanding spatial ability in interior design education: 2D-to-3D visualization proficiency as a predictor of design performance. *Journal of Interior Design*, *44*(3), 141–159. https://doi.org/10.1111/joid.12143

Çam, E., & Özen, Y. A. (2021). Mimarlık temel eğitiminde kavramsal tasarım sürecinde algoritmik düşünme sisteminin etkisini incelemeye yönelik bir çalışma. In A. Çağlar (Ed.), *Mimarlık Alanında Farklı Yaklaşımlar* (pp. 59–74). Nobel Bilimsel Eserler.

Delahunty, T., Seery, N., Dunbar, R., & Ryan, M. (2020). An exploration of the variables contributing to graphical education students' CAD modelling capability. *International Journal of Technology and Design Education*, *30*(2), 389–411. https://doi.org/10.1007/s10798-019-09503-x

Frerejean, J., van Merriënboer, J. J. G., Kirschner, P. A., Roex, A., Aertgeerts, B., & Marcellis, M. (2019). Designing instruction for complex learning: 4C/ID in higher education. *European Journal of Education*, *54*(4), 513–524. https://doi.org/doi.org/10.1111/ejed.12363

Harding, J. E., & Shepherd, P. (2017). Meta-parametric design. *Design Studies*, *52*, 73–95. https://doi.org/10.1016/j.destud.2016.09.005

Henderson, P. B., Cortina, T. J., Hazzan, O., & Wing, J. M. (2007). Computational thinking. *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education*, 195–196. https://doi.org/10.1145/1227310.1227378

Janesarvatan, F., & Van Rosmalen, P. (2023). Instructional design of virtual patients in dental education through a 4C/ID lens: a narrative review. *Journal of Computers in Education*, *11*, 523–556. https://doi.org/10.1007/s40692-023-00268-w

Kavousi, S., Miller, P. A., & Alexander, P. A. (2020). Modeling metacognition in design thinking and design making. *International Journal of Technology and Design Education*, *30*(4), 709–735. https://doi.org/10.1007/s10798-019-09521-9

Korkmaz, Ö., & Bai, X. (2019). Adapting computational thinking scale (CTS) for Chinese high school students and their thinking scale skills level. *Participatory Educational Research*, *6*(1), 10–26. https://doi.org/10.17275/per.19.2.6.1

Kukul, V., & Karatas, S. (2019). Computational thinking self-efficacy scale: Development, validity and reliability. *Informatics in Education*, *18*(1), 151–164. https://doi.org/10.15388/infedu.2019.07

Lee, J. H., Gu, N., Ostwald, M. J., & Jupp, J. (2013). Understanding cognitive activities in parametric design. In J. Zhang & C. Sun (Eds.), *CCIS* (Vol. 369, pp. 38–49). Springer. https://doi.org/10.1007/978-3-642-38974-0_4

Li, Q., Liu, Z., Wang, P., Wang, J., & Luo, T. (2023). The influence of art programming courses on design thinking and computational thinking in college art and design students. *Education and Information Technologies*, *28*(9), 10885–10902. https://doi.org/10.1007/s10639-023-11618-7

RESEARCH ARTICLE
Vol. 9, No. 1, 2026, 1-20
DOI: 10.59389/modular.1853822

Limbu, B. H., Jarodzka, H., Klemke, R., & Specht, M. (2018). Using sensors and augmented reality to train apprentices using recorded expert performance: A systematic literature review. *Educational Research Review*, *25*, 1–22. https://doi.org/10.1016/j.edurev.2018.07.001

Ostrowska-Wawryniuk, K., Strzała, M., & Słyk, J. (2022). Form follows parameter: Algorithmic-thinking-oriented course for early-stage architectural education. *Nexus Network Journal*, *24*(2), 503–522. https://doi.org/10.1007/s00004-022-00603-1

Oxman, R. (2008). Performance-based design: Current practices and research issues. *International Journal of Architectural Computing*, *6*(1), 1–17. https://doi.org/10.1260/147807708784640090

Peteinarelis, A., & Yiannoudes, S. (2018). Parametric models and algorithmic thinking in architectural education. *International Conference on Education and Research in Computer Aided Architectural Design in Europe*, 401–410. https://doi.org/10.52842/conf.ecaade.2018.2.401

Razzouk, R., & Shute, V. (2012). What is design thinking and why is it important? *Review of Educational Research*, *82*(3), 330–348. https://doi.org/10.3102/0034654312457429

Runco, M. A. (1988). Creativity research: Originality, utility, and integration. *Creativity Research Journal*, *1*(1), 1–7. https://doi.org/10.1080/10400418809534283

Suh, J., & Cho, J. Y. (2020). Linking spatial ability, spatial strategies, and spatial creativity: A step to clarify the fuzzy relationship between spatial ability and creativity. *Thinking Skills and Creativity*, *35*. https://doi.org/10.1016/j.tsc.2020.100628

Tan, F. (2024). Translation of form: A model for teaching design coding to undergraduate students. *Nexus Network Journal*, *26*(4), 897–924. https://doi.org/10.1007/s00004-024-00778-9

Tsai, M.-J., & Wang, C.-Y. (2021). Assessing young students' design thinking disposition and its relationship with computer programming self-efficacy. *Journal of Educational Computing Research*, *59*(3), 410–428. https://doi.org/10.1177/0735633120967326

Uçar, U. E., Gökdemir, G., Güzelci, O. Z., & Garip, E. (2023). Integrating digital and manual modes of design. *Periodica Polytechnica Architecture*, *54*(1), 37–49. https://doi.org/10.3311/ppar.21353

Vandewaetere, M., Manhaeve, D., Aertgeerts, B., Clarebout, G., Van Merriënboer, J. J. G., & Roex, A. (2015). 4C/ID in medical education: How to design an educational program based on whole-task learning: AMEE Guide No. 93. *Medical Teacher*, *37*(1), 4–20. https://doi.org/10.3109/0142159X.2014.928407

Van Merriënboer, J. J. G., & Kester, L. (2012). The four-component instructional design model: Multimedia principles in environments for complex learning. In *The Cambridge Handbook of Multimedia Learning* (pp. 71–94). Cambridge University Press. https://doi.org/10.1017/cbo9780511816819.006

Van Merriënboer, J. J. G., & Kirschner, P. A. (2012). *Ten steps to complex learning: A systematic approach to four-component instructional design*. Routledge. https://doi.org/10.4324/9780203096864

Vazquez, E. (2024). Teaching parametric design: fostering algorithmic thinking through incomplete recipes. *Open House International*, *49*(4), 736–751. https://doi.org/10.1108/OHI-06-2023-0135

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35. https://doi.org/10.1145/1118178.1118215

RESEARCH ARTICLE
Vol. 9, No. 1, 2026, 1-20
DOI: 10.59389/modular.1853822

AUTHOR BIOGRAPHY

Hasan TAŞTAN

Received his BS degree in 2014 and an MS degree in 2018, each in the Department of Architecture from Yildiz Technical University. He earned his PhD in 2023 from the same institution. He is currently serving as an Assistant Professor in the Department of Architecture at Yildiz Technical University. His research interests include computational design, artificial intelligence, and human–computer interaction with a focus on learning and education.