



A Multi-Tier Modular Architecture for Designing AI Curricula for Non-CS Majors: Integrating CS2023 Standards and Technical Governance Frameworks

Fawzeyia Elhashmi Mohamad Salah^{1*}, Nesrin Muftah Elrgibe²

^{1,2} Computer Department, Faculty of Education-Tripoli, Tripoli University, Libya.

^{1*}Corresponding Author's Email: F.Salah@uot.edu.ly

Received: 30/8/2026 Revised: 1/9/2026 Accepted: 30/9/2026 Publication: 08/10/2026

ABSTRACT

The growing integration of generative artificial intelligence into non technical disciplines has highlighted a gap in how higher education institutions cultivate computational literacy among students outside computer science (CS) programmes. Current educational approaches generally fall into two extremes, they either impose syntax-heavy programming tracks that alienate non-CS cohorts or focus strictly on theoretical overviews with little technical depth. This paper proposes a multi-tier modular architecture that re-encodes the ACM/IEEE Computer Science Curricula 2023 (CS2023) standards into four functionally independent tiers: Data Foundations, Algorithmic Logic, Machine Learning and Generative Models, and Algorithmic Governance and Security. Each operationalised through no code practical instruments such as retrieval augmented generation (RAG), prompt engineering, and prompt-injection red-teaming. Using a Design Science Research (DSR) approach, the framework incorporates UNESCO's AI Competency Framework and the European DigComp 2.2 framework as a core structural layer rather than an isolated ethics topic. A Standards Traceability Matrix demonstrates one-to-one alignment between each tier, its practical tasks, and the constituent competencies of the three reference frameworks. The paper addresses four key research questions regarding architectural abstraction, technical governance, non-coding practical integration, and cross disciplinary scalability. The contribution is a reusable framework from which domain tailored AI course packages can be derived without losing technical depth, thereby repositioning governance, security, and applied comprehension as co-equal pillars of AI literacy for non CS majors.

Keywords: *AI curriculum design; AI literacy; CS2023; multi-tier architecture; technical governance; prompt engineering; non-CS majors; retrieval-augmented generation.*

نموذج معياري متعدد المستويات لتصميم مناهج الذكاء الاصطناعي لغير المتخصصين في

وأطر الحوكمة التقنية CS2023 علوم الحاسوب: دمج معايير

فوزية الهاشمي محمد صلاح^{1*}، نسرين مفتاح الرقيب²

^{1,2} قسم الحاسوب، كلية التربية طرابلس، جامعة طرابلس، ليبيا

المستخلص

يتناول هذا البحث الفجوة الهيكلية في تعليم الذكاء الاصطناعي للطلاب غير المتخصصين في علوم الحاسوب. تتباين النماذج التعليمية القائمة بين مسارات برمجية مثقلة بالصياغة اللغوية تُنفّر غير المتخصصين، ومفاهيم نظرية تتنازل عن الصرامة الهندسية. هذه الورقة تقترح بنية متعددة الطبقات المعيارية تعيد ترميز معايير ACM/IEEE CS2023 إلى أربع طبقات وظيفية مستقلة: أسس البيانات، والمنطق الخوارزمي، ونماذج التعلم الآلي والتوليد، والحوكمة الخوارزمية

والأمن. تُشغّل كل طبقة عبر أدوات عملية غير برمجية كالتوليد المعزّز بالاسترجاع (RAG) وهندسة الأوامر واختبار اختراق الأوامر. بالاستناد إلى منهجية تصميم العلوم (DSR)، يدمج الإطار توصيات اليونسكو لكفاءات الذكاء الاصطناعي وإطار DigComp 2.2 الأوروبي كطبقة حوكمة هيكلية لا كملحق نظري. تُبين مصفوفة التتبع القياسي التناظر واحدًا لواحد بين الطبقات والمهام العملية وكفاءات الأطر المرجعية الثلاثة. تُعالج الورقة أربعة أسئلة بحثية تتعلق بالتجريد المعماري، والحوكمة التقنية، والدمج التطبيقي غير البرمجي، وقابلية التوسع عبر التخصصات. تتمثل المساهمة في إطار تعريف قابل لإعادة الاستخدام لاشتقاق حزم مقررات مُخصّصة مجاليًا دون تمبيع الصرامة الحاسوبية، بما يعيد تموقع الحوكمة والأمن والفهم التطبيقي بوصفها أعمدة متكافئة لمعرفة الذكاء الاصطناعي لدى غير المتخصصين. **الكلمات المفتاحية:** تصميم مقررات الذكاء الاصطناعي؛ ثقافة الذكاء الاصطناعي؛ SC2023؛ بنية متعددة الطبقات؛ الحوكمة التقنية؛ هندسة الأوامر؛ غير المتخصصين في علوم الحاسوب؛ التوليد المعزّز بالاسترجاع.

1. Introduction

1.1 Context & Technical Challenge

Generative Artificial Intelligence's spread into realms traditionally impervious to computation has reformulated an age-old educational question: What is it that lawyers, businessmen, or humanities students ought to know about the technical underpinning of the technology that now shapes their decision-making processes? This is not a matter of digital literacy in the inherited sense. Interaction has moved to a new site. While past generations came to know technology as mysterious but clearly-defined tools spreadsheets, databases, and statistical programs the present generation has been born into a world in which technology is capable of producing text, writing contracts, evaluating qualifications, and performing other professional tasks on an industrial scale, thus making indistinct the line dividing consumption from production (Brown et al., 2020; Knoth et al., 2024). Technology has become conversational in form but not in substance.

Nonetheless, there is an important shortcoming of the approach taken here. Students who learn how to use LLMs only through the interface would mistake conversational ability for correctness of reasoning, thinking that fluent answers always mean proper reasoning and ignoring such aspects as the intricacies of the transformer inference procedure (Vaswani et al., 2017), the intricacies of the data training process, and such well-known problems as hallucinations and prompt injection issues (Greshake et al., 2023). Thus, the main goal of all this education process, is not to make philosophers and financiers to become Python programmers but to create a course that will help students get from using the interface to developing algorithms without any previous programming skills.

1.2 Research Significance

However, this research paper seeks to fill an architectural, not an incremental gap. The traditional models used in computing sciences, most notably the ACM/IEEE-CS/AAAI Joint Task Force CS2023 report on Computer Science Curricula 2023 (ACM/IEEE-CS/AAAI Joint Task Force, 2024), implement a competency model that assumes accumulated programming experience through several semesters. At the opposite pole, general introductory AI courses often deployed as breadth-first surveys substitute ethical reflection and sandbox tool use for any engagement with the data, model, and inference pipeline that constitutes the actual engineering substrate (Shehu, 2026; Tadimalla & Maher, 2025). Neither pole serves the non-CS student adequately. The first excludes them; the second under-prepares them.

This paper proposes a framework : a structural translation mechanism that decomposes the CS body of knowledge into functionally independent tiers, each capable of being taught through no-code practical instruments while remaining traceable to the international standards that legitimate CS as a discipline. This approach provides institutions with a framework to develop core skills in system security, reliability, and decision making mechanisms competencies that the UNESCO Recommendation on the Ethics of Artificial Intelligence (UNESCO, 2021) and its derivative competency frameworks (Miao et al., 2024) increasingly demand, which standalone ethics modules often fail to provide (Titareva et al., 2026).

1.3 Research Scope

The scope of the present work is deliberately meta-architectural. The contribution is not a single syllabus but a scalable framework. An underlying risk lurks beneath this superficial simplicity. Those students who are only introduced to LLMs through the conversational interface will form a behavioral model of intelligence, where fluency of response is equated with sound reasoning, while forgetting about the technical details of the inference process of transformers (Vaswani et al., 2017), the vagaries of the data training process, and the well known pitfalls of hallucinations and prompt injection vulnerabilities (Greshake et al., 2023). Consequently, the challenge of the learning process is not to turn all philosophers and financiers into Python programmers but rather to provide a learning path from the passive consumption of interfaces to the structural knowledge of algorithm engineering without requiring any programming skills as a prerequisite. capable of guiding the derivation of multiple AI courses tailored to diverse non-technical disciplines business, law, humanities, public administration while supporting progressive course levels spanning foundational and advanced tiers. By treating the curriculum as a derivable artefact rather than a fixed deliverable, the architecture accommodates disciplinary heterogeneity without abandoning a common structural spine. A business analytics course and a legal-technology seminar can thus share the same four tier scaffold while diverging in the datasets, governance case studies, and prompt engineering scenarios they instantiate.

1.4 Research Questions

This study addresses four research questions (RQs), each isolating a distinct engineering tension. Considered together, they circumscribe the design space within which the architecture must operate.

RQ1 (Architectural Design & Abstraction). How can such a framework be used to convert the complex technical specifications of CS2023 into a computer science specific structure that is understandable for non computer science majors?

RQ2 (Technical Governance & Security). What is the method by which the technical governance framework can be incorporated to integrate the UNESCO AI Competency Framework and DigComp 2.2, as a level of structure, in addressing the risk of AI?

RQ3 (Non Coding Practical Integration). How can institutions embed no-code tools including prompt engineering, RAG, and security red teaming across various architectural layers to promote applied understanding among non-CS students?

RQ4 (Scalability & Reusability). What is the degree to which the architecture proposed is scalable and reusable when deriving and designing different academic courses not in technical areas?

The remainder of the paper proceeds as follows. Section 2 reviews the related literature and articulates the gap the architecture addresses. Section 3 sets out the Design Science Research methodology. Section 4 specifies the three reference frameworks and the rationales for their integration. Section 5 presents the proposed multi tier architecture. Section 6 develops the Standards Traceability Matrix. Section 7 details the intended learning outcomes. Finally, Section 8 provides recommendations for institutional adoption.

2. Related Work & Literature Gap Analysis

2.1 Thematic Review of Existing Literature

2.1.1 AI Curricula for Non-Majors

The literature on AI curricula for non majors shows a clear divide . Shehu (2026) describes the UNIV 182 course at George Mason University which uses a “prerequisite free” design that organizes instruction around a unifying conceptual pipeline traversed repeatedly at increasing technical sophistication a laudable departure from surface-level surveys. Yet the course still relays on programming assignments and artifact building projects that, while scaffold-ed, presuppose a degree of coding tolerance not uniformly distributed across non CS

cohorts. Tadimalla and Maher (2025), in their interdisciplinary socio technical framework, adopt a four pillar structure spanning technical scope, interaction, critical ethical use, and societal implication. Their design is adjustable across CS and non CS audiences but treats governance primarily as a discursive pillar rather than an engineered, testable layer. The result is a recurring compromise, courses either retain enough code to deter the target population or strip out enough engineering to forfeit rigor.

Other studies examine generative AI integration in programming courses for non CS STEM students. De Oliveira et al. (2026), in a two semester study at UNICAMP, restructured their curriculum to include an introductory programming course around problem understanding, prompt formulation, and evaluation of AI generated code. Their findings sustained engagement and perceived relevance coupled with emergent debugging difficulties suggest that prompt-mediated interaction can substitute for manual implementation, but only when the underlying curriculum retains a structural account of how the AI tool arrives at its output. Without that account, students oscillate between over-trust and brittle dependence.

2.1.2 CS Standards & Architectural Abstraction

The CS2023 guidelines (ACM/IEEE-CS/AAAI Joint Task Force, 2024) represent the latest international framework on undergraduate CS education, organizing the discipline into seventeen Knowledge Areas and pairing a backward compatible knowledge model with a competency framework influenced by Computing Curricula 2020. Eaton and Epstein (2024), writing at AAAI-24, detail the rationale for elevating Artificial Intelligence (AI), Security (SEC), and Society, Ethics, and the Profession (SEP) to first-order curricular concerns. Aly et al. (2024) echo this emphasis, identifying the same three areas as the pressure points of contemporary CS education.

What this body of work conspicuously lacks is a structural translation mechanism. CS2023 specifies what competencies a CS graduate should possess, it does not specify how those competencies can be conveyed to learners who will never enrol in a CS degree. The principle of abstraction long recognized within CS itself as the discipline's central intellectual lever offers an under exploited route. By abstracting away from syntax while preserving the data algorithm model governance pipeline, it becomes possible to convey core computing principles to non CS students without the barrier of programming-language acquisition. The literature has yet to systematise this translation; the present architecture takes up that task.

2.1.3 Technical Governance & AI Ethics Integration

A third body of literature focuses on governance and AI ethics.. Hackl et al. (2025) propose the “AI Literacy Heptagon,” a structured competency model for higher education that synthesises prior literacy frameworks into seven interrelated dimensions. While their model offers structural clarity, it treats governance as a literacy dimension rather than an engineered artefact. Knoth et al. (2024) demonstrate empirically that higher quality prompt engineering skills predict higher-quality LLM output, and that facets of AI literacy condition prompting behaviour, a finding that reinforces the case for embedding governance within the practical tier rather than quarantining it in a standalone ethics module. Titareva et al. (2026), surveying AI governance across sixteen countries, note that universities often struggle to implement UNESCO's ethical guidelines; ethics and governance are treated as isolated theoretical discussions rather than functional technical layers such as input-security guardrails and output filtering.

Consequently, governance in current curricula is either taught theoretically without practical testing, or restricted to computer science majors. The architecture proposed here treats governance as a structural tier with its own practical instrumentation, bridging the gap between theory and practice often found in existing literature.

2.2 Literature Comparative Analysis

Table 1 consolidates the comparison. Four archetypal models are evaluated against four metrics: CS2023 baseline alignment, no-code applied execution, governance as a functional tier, and meta-framework scalability. The symbol set encodes degree of fulfilment: ●● full, ● partial, ○ absent.

within the practical tier rather than quarantining it in a standalone ethics module. Titareva et al. (2026), surveying AI governance across sixteen countries, note that universities often struggle to implement UNESCO's ethical guidelines, ethics and governance are treated as isolated theoretical discussions rather than functional technical layers such as input-security guardrails and output filtering.

Consequently, governance in current curricula is either taught theoretically without practical testing, or restricted to computer science majors. The architecture proposed here treats governance as a structural tier with its own practical instrumentation, bridging the gap between theory and practice often found in existing literature.

2.2 Literature Comparative Analysis

Table 1 consolidates the comparison. Four archetypal models are evaluated against four metrics: CS2023 baseline alignment, no-code applied execution, governance as a functional tier, and meta-framework scalability. The symbol set encodes degree of fulfilment: ●● full, ● partial, ○ absent.

Table 1. *Comparative Evaluation of Pedagogical Models Against the Four Architectural Metrics*

Evaluation Metric / Feature	Traditional Pedagogical Models	Direct Coding-Based Courses	Isolated Ethics Modules	Proposed Architecture (Our Framework)
CS2023 Baseline Alignment	○ Absent	● Partial (high code complexity)	○ Absent	●● Present (via structural abstraction)
No-Code Applied Execution	○ Absent	○ Requires programming	○ Absent	●● Present (Prompt Engineering & RAG)
Governance as a Functional Tier	○ Theoretical essay	○ Absent	● Theoretical essay	●● Technical security & guardrail tier
Meta-Framework Scalability	○ Single course	○ Single course	○ Unspecified	●● Multi-course derivative capability

Three observations follow. First, traditional pedagogical models and isolated ethics modules fail on every metric except, in the latter case, a thin theoretical gesture toward governance. Second, direct coding-based courses satisfy the CS2023 baseline but at a code complexity that excludes the non-CS population and offer no governance tier whatsoever. Third, no existing model attains all four metrics simultaneously. The proposed architecture is positioned to do so by combining structural abstraction, no-code practical execution, an engineered governance tier, and a meta-framework capable of multi-course derivation.

3. Research Methodology

3.1 Research Approach: Design Science Research

This study uses Design Science Research (DSR) as its methodological framework. DSR, articulated in its modern form by Hevner et al. (2004) and structured as a six-step process by Peffers et al. (2007), is concerned with the design, construction, and evaluation of artefacts that address heretofore unsolved problems. The proposed artifact in this study is the multi-tier modular architecture itself a computing construct intended to guide curriculum design rather than to execute computations, but no less an engineered object for that. DSR is appropriate precisely because the research question is not whether a phenomenon occurs but whether a designed structure can satisfy a set of engineering requirements under explicit constraints.

The Peffers et al. (2007) process problem identification and objectives definition, design and development, demonstration, evaluation, and communication maps cleanly onto the framework-development phases described below. The problem has been identified in Section 2: no existing model satisfies all four evaluation metrics simultaneously. The objective is a meta-framework that does. The design and development phases produce the architecture; the demonstration and evaluation phases instantiate the Standards Traceability Matrix and the engineered learning outcomes as evidence of compliance and reusability.

3.2 Framework Development Phases

The architecture emerged from four iterative phases, each producing a verifiable intermediate artifact.

Phase 1 : Requirements Analysis. The structural and applied AI competencies required of non-CS majors in technical work environments were catalogued. Sources included the CS2023 Knowledge Areas most relevant to non-specialist literacy (AI, SEC, SEP, Data Management, and Foundations of Programming Languages), the UNESCO student and teacher competency frameworks (Miao et al., 2024; Miao & Cukurova, 2024), and the DigComp 2.2 competence areas concerning data literacy and digital content interaction (Vuorikari et al., 2022). The requirements converged on four functional responsibilities: representing data, reasoning about algorithmic logic, understanding generative models, and governing their behaviour.

Phase 2: Structural Decomposition. The CS and AI body of knowledge was partitioned into four independent functional tiers, Data Foundations, Algorithmic Logic, Machine Learning and Generative Models, and Algorithmic Governance and Security each defined by a theoretical dimension and a practical dimension. Independence is deliberate: each tier is teachable as a self contained module, yet the four compose into a coherent pipeline that mirrors the actual engineering lifecycle of an intelligent system.

Phase 3 : Standards Mapping. Each tier was mapped directly to specific sub-domains in CS2023, UNESCO, and DigComp 2.2. The mapping is not aspirational, it is enumerated in the Standards Traceability Matrix (Section 6) and verifiable by inspection. A tier that cannot be mapped is, by construction, extraneous; a standard outcome that no tier addresses is a coverage defect requiring revision.

Phase 4: Operationalisation. Visual workflow engines, prompt engineering protocols, and hands-on laboratory exercises were integrated into each tier to ensure that the theoretical dimension is always paired with a no-code practical task. This operational phase allows abstract concepts to be tested empirically, a student who has designed a flowchart decision tree, implemented a RAG pipeline over custom documents, and executed a prompt-injection red-team has, in each case, demonstrated a competency that a written examination could not capture.

3.3 Validation & Reliability Mechanism

Validation involves two main components. The first is the Standards Traceability Matrix, a matrix that maps structural alignment with international computing and governance standards by linking each tier's theoretical principles and practical tasks to specific learning outcomes in CS2023, UNESCO, and DigComp 2.2. The second is a reusability evaluation: the architecture's capacity to generate derivable course packages across disciplines is demonstrated by deriving two hypothetical instantiation a business analytics foundational course and a legal technology advanced course from the same four-tier framework. If an example fails to meet the matrix criteria, the architecture is revised to ensure scalability across domains. This iterative loop is characteristic of DSR evaluation, which privileges artefact refinement over statistical generalisation (Hevner et al., 2004).

4. Reference Frameworks & Integration Rationales

The architecture draws on three reference frameworks. These frameworks are integrated functionally, each framework supplies a dimension that the others lack, and the architecture's tiers are the loci at which the three intersect.

4.1 ACM/IEEE CS2023 Guidelines

CS2023 (ACM/IEEE-CS/AAAI Joint Task Force, 2024) supplies the technical computing foundation. Its seventeen Knowledge Area of which Artificial Intelligence (AI), Security (SEC), Society, Ethics, and the Profession (SEP), Data Management (DM), and Algorithmic Foundations (AL) are most pertinent here ensure academic rigor in data structures, algorithmic logic, and machine learning architectures. The competency framework's emphasis on outcomes rather than topic coverage permits the architectural abstraction the present design requires, a non-CS student need not master every CS2023 topic, but every competency the architecture claims to teach must be traceable to a CS2023 statement. Eaton and Epstein (2024) note that CS2023 elevates AI, SEC, and SEP precisely because contemporary CS education had under served them; the architecture inherits this emphasis by distributing those three areas across all four tiers rather than confining them to one.

4.2 UNESCO AI Competency Framework

The UNESCO AI Competency Framework for Students (Miao et al., 2024) covers technical governance, algorithmic accountability and organisational risk management. Its four dimensions a human-centred mindset, ethics of AI, AI techniques and applications, and AI system design align directly with the framework's governance tier. The companion framework for teachers (Miao & Cukurova, 2024) informs the pedagogical scaffolding that operationalises each tier. UNESCO's normative instrument on AI ethics (UNESCO, 2021) supplies the values substrate; Titareva et al. (2026) document both its global reach 194 member states and the implementation gap that an engineered governance tier is designed to close.

4.3 European DigComp 2.2 Framework

DigComp 2.2 (Vuorikari et al., 2022) connects data processing and algorithmic interaction with advanced digital competencies. Its five competence area information and data literacy, communication and collaboration, digital content creation, safety, and problem solving bridge the technical and the civic in a manner that neither CS2023 nor UNESCO fully achieves on its own. The framework's emphasis on the safety of digital content creation aligns directly with the architecture's treatment of prompt injection defence and output guardrails as competencies rather than afterthoughts.

4.4 Integration Rationales

Combining the three frameworks establishes a balanced curriculum architecture that unifies technical depth, regulatory responsibility, and practical execution. CS2023 alone would produce a competent CS graduate irrelevant to non-CS cohorts. UNESCO alone would yield a governance literate citizen unable to evaluate the underlying technical components of

AI systems. DigComp alone would produce a digitally competent user without the structural understanding that distinguishes literacy from fluency. The architecture's tiers are the points at which the three frameworks' complementary demands are reconciled: Data Foundations aligns with CS2023's. Data Management and DigComp's information and data literacy Algorithmic Logic instantiates CS2023's Algorithmic Foundations and DigComp's problem solving, Machine Learning and Generative Models instantiates CS2023's AI and DigComp's digital content creation Algorithmic Governance and Security instantiates CS2023's SEC and SEP, UNESCO's ethics and system design dimensions, and DigComp's safety area. The integration is thus not a concatenation of frameworks but a structural composition in which each tier is the intersection of three legitimate demands.

5. Proposed Multi-Tier Modular Architecture

The proposed architecture consists of four tiers. Each tier is defined by a structural theoretical dimension and a no-code practical application. While the tiers follow a logical sequence (data, algorithm, model, governance), they remain modular in design, each can be taught as a self contained unit and the sequence can be compressed, expanded, or re-weighted to suit a derived course's discipline and level. Table 2 presents the full architecture; the subsections that follow elaborate each tier.

Table 2. *The Proposed Multi-Tier Modular Architecture: Theoretical and Practical Dimensions*

Architectural Tier	Structural Theoretical Dimension	Proposed Practical Application (No-Code)
Tier 1 Data Foundations	Structured vs. unstructured data representation, digital matrices, and data lifecycle pipelines.	No-code dataset processing and classification; retrieval-augmented generation (RAG) implementation linking custom documents to LLMs.
Tier 2 Algorithmic Logic	Decision trees, conditional logic, and control-flow processing mechanisms.	Flowchart design and interactive decision-tree modelling using visual workflow-engineering platforms.
Tier 3 ML & Generative Models	Transformer architectures, training vs. inference mechanics , and LLM foundations.	Prompt-engineering techniques (zero/few-shot, chain-of-thought) and model benchmarking and evaluation.
Tier 4 Algorithmic Governance & Security	Data bias, transparency, algorithmic reliability, and security vulnerabilities.	Prompt-injection defence testing, input-safety constraint design, and output guardrail filtering.

5.1 Tier 1: Data Foundations

The first tier introduces the data foundation required for subsequent analysis, structured versus unstructured data representation, digital matrices, and the data lifecycle pipeline. Students examine data as a structured input whose representation choices constrain every downstream operation. The practical component applies these concepts using no-code dataset processing and classification, and through retrieval augmented generation (Lewis et al., 2020) implementations that link custom documents to LLMs. The RAG exercise allows students to evaluate data impact directly by retrieving over a corpus they themselves assemble, students

observe directly how the content and structure of the data layer shape the model's output a lesson that no lecture on bias conveys as efficiently.

5.2 Tier 2: Algorithmic Logic

The second tier abstracts the control-flow substrate. Decision trees, conditional logic, and control-flow processing mechanisms are conveyed not through code but through flowchart design and interactive decision tree modelling on visual workflow-engineering platforms. By externalising algorithmic structure into a visual notation that the student manipulates directly, students develop algorithmic reasoning skills without focusing on programming syntax. For example, a law student modeling a legal eligibility workflow applies algorithmic reasoning, the absence of a compiler does not diminish the structural achievement.

5.3 Tier 3: Machine Learning & Generative Models

The third tier addresses the model layer transformer architectures (Vaswani et al., 2017), the distinction between training and inference, and the foundations of large language models. The theoretical component balances foundational knowledge with applied technical concepts, moving beyond superficial introductory surveys that characterises the superficial pole of the bifurcation. The practical component applies these concepts through structured prompt engineering (Knoth et al., 2024), encompassing zero-shot and few-shot techniques (Brown et al., 2020) and chain-of-thought reasoning (Wei et al., 2022), and through model benchmarking and evaluation. Instead of training models from scratch, students learn to evaluate and control them through the natural language interface that constitutes, for non CS practitioners, the primary channel of control.

5.4 Tier 4: Algorithmic Governance & Security

The fourth tier is the architecture's distinctive contribution. Where prior curricula treat governance as a discursive appendix, this tier engineers it. The theoretical dimension addresses data bias, transparency, algorithmic reliability, and security vulnerabilities. The practical dimension operationalises each through prompt-injection defence testing (Greshake et al., 2023), input-safety constraint design, and output-guardrail filtering. A student who has attempted to extract a restricted output from a guarded model, and who has then redesigned the guardrails to resist the attack, gains a practical understanding of governance that complements theoretical coursework. The tier treats security as a core design requirement rather than an optional topic, in line with the SEC and SEP emphases that CS2023 elevated and that Aly et al. (2024) identify as the contemporary pressure points of CS education.

6. Standards Alignment & Traceability Matrix

The Standards Traceability Matrix demonstrates the architecture's alignment with structural compliance with the three reference frameworks. Each row pairs a tier with its theoretical principles and practical tasks; each column maps the pairing to specific learning outcomes in CS2023, UNESCO, and DigComp 2.2. The matrix is comprehensive in the sense that every tier task pairing is traceable to at least one outcome in each framework, without claiming unmapped or redundant outcomes. Table 3 presents the matrix.

Table 3. *Standards Traceability Matrix: Mapping Architectural Tiers to CS2023, UNESCO, and DigComp 2.2*

Tier	Theoretical Principles & Practical Tasks	CS2023 Knowledge Area	UNESCO AI Competency	DigComp 2.2 Competence
Tier 1 Data Foundations	Structured/unstructured data; digital matrices; data lifecycle; no-code dataset processing; RAG implementation.	Data Management (DM); Algorithmic Foundations (AL)- data representation.	AI techniques & applications- data as system input.	Area 1: Information & data literacy (browsing, searching, filtering, evaluating data).
Tier 2 Algorithmic Logic	Decision trees; conditional logic; control-flow mechanisms; flowchart design; interactive decision-tree modelling.	Algorithmic Foundations (AL); Software Development Fundamentals (SDF).	AI techniques & applications; AI system design- problem decomposition.	Area 5: Problem solving - identifying needs and technical responses.
Tier 3 ML & Generative Models	Transformer architectures; training vs. inference; LLM foundations; prompt engineering; benchmarking & evaluation.	Artificial Intelligence (AI)- ML, inference, generative models.	AI techniques & applications; AI system design-model interaction.	Area 3: Digital content creation; Area 5: Problem solving.
Tier 4 Governance & Security	Data bias; transparency; reliability; vulnerabilities; prompt-injection defence; input-safety constraints; output guardrails.	Security (SEC); Society, Ethics & the Profession (SEP).	Human-centred mindset; Ethics of AI; AI system design-accountability & risk.	Area 4: Safety - protecting devices, data & privacy; Area 5: Problem solving.

The matrix performs three functions. First, it provides empirical evidence of structural rigor, compliance is verified by inspection, not asserted. Second, it renders the architecture auditable: an external reviewer can trace any claimed competency to its standard of origin. Third, it functions as a design constraint during derivation: when a new course is instantiated from the architecture, the matrix helps ensure comprehensive topic coverage. A derived course that leaves a cell empty is, by construction, incomplete.

7. Engineered Learning Outcomes

The learning outcomes derived from this framework are measurable and applied. Each maps to one or more tiers and to the traceability matrix, and each is associated with a specific assessment task. The verbs are deliberately Bloom aligned and operationally specific; “analyse,” “apply,” “evaluate,” and “design” denote cognitive operations whose execution leaves a tangible artefact a pipeline, a benchmark, an evaluation report, a guardrail configuration available for assessment.

LO1. Analyse the structural pipeline of data flow within intelligent computing systems.

Tiers 1 & 3 · CS2023: DM, AI · DigComp: information & data literacy. Measurable through RAG-pipeline construction and data-representation exercises.

LO2. Apply advanced prompt engineering techniques, zero-shot, few-shot, and chain-of-thought to programmatically control generative model outputs using natural-language interface protocols.

Tier 3 · CS2023: AI · DigComp: digital content creation. Measurable through benchmarked output quality comparisons across prompting strategies.

LO3. Evaluate model performance, verify accuracy, and identify instances of hallucination and algorithmic bias.

Tiers 3 & 4 · CS2023: AI, SEP · UNESCO: ethics of AI. Measurable through structured evaluation reports on assigned model outputs.

LO4. Design technical guardrails for user inputs to defend AI deployments against adversarial prompt injection attacks.

Tier 4 · CS2023: SEC, SEP · UNESCO: AI system design · DigComp: safety. Measurable through red-team exercises in which students attack and then defend a guarded deployment.

The learning outcomes are directly linked to the practical instrument through which it is demonstrated. A student who satisfies LO4 has not merely recited a definition... they have built a guardrail, broken it, and rebuilt it. That artifact is the evidence, and the assessment is the artifact. This practical approach allows the architecture to translate the architecture converts the abstraction of CS2023 competencies into demonstrable, non-coding practice.

8. Conclusion & Recommendations

8.1 Conclusion

The multi tier modular architecture proposed in this paper provides a structured framework to address the divide in that has constrained AI education for non-CS majors. By decomposing the CS body of knowledge into four functionally independent tiers by applying each tier through hands-on no-code tasks, and by integrating UNESCO and DigComp 2.2 into an applied governance tier, the architecture satisfies the four research questions posed at the outset. RQ1 is addressed through architectural abstraction CS2023 competencies are conveyed without syntactic barriers. RQ2 is addressed through the governance tier, technical frameworks are engineered as functional layers with practical, testable exercises. RQ3 is addressed through the embedding of prompt engineering, RAG, and red-teaming across the tiers, each reinforcing applied comprehension. RQ4 is addressed through the framework's adaptability, the same four-tier scaffold supports disciplinary and level variation without structural revision.

Consequently, the framework contributes by integrating governance, security, and applied knowledge as core components of AI literacy for non-CS majors, and it supplies a reusable meta-framework from which domain-tailored course packages can be derived while maintaining core technical standards.. In doing so, it narrows the gap between the complex programming requirements of traditional CS curricula and the superficiality of general introductory courses—a gap that prior models have widened rather than closed.

8.2 Recommendations

Based on these findings, two key recommendations are proposed. First, academic institutions should adopt the architecture as a guiding framework for developing tailored AI courses across departments. Adoption should be accompanied by the Standards Traceability Matrix as a mandatory design instrument, ensuring all new course proposals include a completed traceability matrix, verifying curriculum alignment during initial course design.

This approach maintains departmental autonomy while ensuring consistent technical standards.

Second, higher education institutions can establish specialized Generative AI Labs focused on no-code toolsets for interactive instruction and technical validation. The lab's function is not to teach programming but to host the practical instrumentation visual workflow engines, RAG-assembly environments, prompt engineering benches, red-team sandboxes that the architecture's tiers require. By concentrating these tools in a shared facility, institutions lower the per-department cost of adoption and create a locus for cross-disciplinary pedagogical innovation that isolated departmental initiatives struggle to sustain. These labs embody the framework's core objective combining technical rigor with accessible, non-coding methodologies.

9. References

- ACM/IEEE-CS/AAAI Joint Task Force. (2024). Computer science curricula 2023 (CS2023): The final report. Association for Computing Machinery; IEEE Computer Society; Association for the Advancement of Artificial Intelligence. <https://doi.org/10.1145/3664191>
- Aly, S. G., Becker, B. A., Kumar, A. N., & Raj, R. K. (2024). Computer science curricula 2023 (CS2023): Rising to the challenges of change in AI, security, and society. In *Proceedings of the 2024 Innovation and Technology in Computer Science Education V. 2 (ITiCSE 2024)*. Association for Computing Machinery. <https://doi.org/10.1145/3649405.3659537>
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., & Amodei, D. (2020). Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*. <https://arxiv.org/abs/2005.14165>
- De Oliveira, R. M., da Silva, E. P., Wainer, J., & Azevedo, R. (2026). Generative AI in introductory programming for non-CS STEM students: A two-semester study. *IEEE Access*, 14, 88219. <https://doi.org/10.1109/ACCESS.2026.3701610>
- Eaton, E., & Epstein, S. L. (2024). Artificial intelligence in the CS2023 undergraduate computer science curriculum: Rationale and challenges. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-24)*. Association for the Advancement of Artificial Intelligence.
- Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., & Fritz, M. (2023). Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security (AISec '23)* (pp. 79–90). Association for Computing Machinery. <https://doi.org/10.1145/3605764.3623985>
- Hackl, V., Müller, A. E., & Sailer, M. (2025). The AI literacy heptagon: A structured approach to AI literacy in higher education. *arXiv*. <https://arxiv.org/abs/2509.18900>
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1), 75–105. <https://doi.org/10.5555/2017212.2017217>
- Knoth, N., Tolzin, A., Janson, A., & Leimeister, J. M. (2024). AI literacy and its implications for prompt engineering strategies. *Computers and Education: Artificial Intelligence*, 6, 100225. <https://doi.org/10.1016/j.caeai.2024.100225>
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented

- generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems* 33 (NeurIPS 2020). <https://arxiv.org/abs/2005.11401>
- Miao, F., & Cukurova, M. (2024). AI competency framework for teachers. UNESCO. <https://doi.org/10.54675/ZJTE2084>
- Miao, F., Shiohira, K., & Lao, N. (2024). AI competency framework for students. UNESCO. <https://doi.org/10.54675/JKJB9835>
- Peffer, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3), 45–77. <https://doi.org/10.2753/MIS0742-1222240302>
- Shehu, A. (2026). From understanding to creation: A prerequisite-free AI literacy course with technical depth across majors. *arXiv*. <https://arxiv.org/abs/2604.09634>
- Tadimalla, S. Y., & Maher, M. L. (2024a). AI literacy for all: Adjustable interdisciplinary socio-technical curriculum. *arXiv*. <https://arxiv.org/abs/2409.10552>
- Tadimalla, S. Y., & Maher, M. L. (2024b). Sociotechnical AI education course design for CS majors and non-majors. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 2 (SIGCSE TS 2024)*. Association for Computing Machinery. <https://doi.org/10.1145/3641555.3705252>
- Tadimalla, S. Y., & Maher, M. L. (2025). AI literacy as a core component of AI education. *AI Magazine*, 46, e70007. <https://doi.org/10.1002/aaai.70007>
- Titareva, T., Thomson, G., Barkane, I., Csonka-Stambekova, A., Dai, Y., Glass, C., & Roumate, F. (2026). Operationalising AI ethical principles in higher education with the UNESCO recommendation on the ethics of artificial intelligence by policy makers and global universities. AI Policy Lab, Umeå University. <https://aipolicylab.se/aipex/>
- UNESCO. (2021). Recommendation on the ethics of artificial intelligence (Document SHS/BIO/PI/2021/1). UNESCO. <https://unesdoc.unesco.org/ark:/48223/pf0000381137>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems* 30 (NeurIPS 2017). <https://arxiv.org/abs/1706.03762>
- Vuorikari, R., Kluzer, S., & Punie, Y. (2022). DigComp 2.2: The digital competence framework for citizens, with new examples of knowledge, skills and attitudes (EUR 31006 EN). Publications Office of the European Union. <https://doi.org/10.2760/490274>
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems* 35 (NeurIPS 2022). <https://arxiv.org/abs/2201.11903>