ITiCSE 2025:Innovation and Technology in Computer Science Education
June 27 - July 2, 2025
Nijmegen, Netherlands

Conference Sponsors:
SIGCSE

RESEARCH-ARTICLE

The Rest of the Robots: Generative AI in Post-introductory Computing Education

DENNIS J. BOUVIER, United States Air Force Academy, Colorado Springs, CO, United States

BRUNO PEREIRA CIPRIANO, Lusophone University of Humanities and Technologies, Lisbon, Portugal

RICHARD GLASSEY, KTH Royal Institute of Technology, Stockholm, Stockholms, Sweden

OLGA PETROVSKA, Swansea University, Swansea, West Glamorgan, Wales, U.K.

EMMA ANDERSON, University of Northumbria, Newcastle, Tyne and Wear, U.K.

ANASTASIIA BIRILLO, JetBrains s.r.o., Prague, Czech Republic

View all

ITiCSE-WGR 2025: 2025 Working Group Reports on Innovation and Technology in Computer Science Education (June 2025)
https://doi.org/10.1145/3760545.3783970
ISBN: 9798400721670

The Rest of the Robots: Generative AI in Post-introductory Computing Education

Dennis J Bouvier[*] [†]
United States Air Force Academy
Colorado Springs, CO, USA
djb@acm.org

Bruno Pereira Cipriano[†]
Lusofona University
Lisbon, Portugal
bcipriano@ulusofona.pt

Richard Glassey[†]
KTH Royal Institute of Technology
Stockholm, Sweden
glassey@kth.se

Olga Petrovska[†]
Swansea University
Swansea, United Kingdom
olga.petrovska@swansea.ac.uk

Emma Anderson
Northumbria University
Newcastle upon Tyne, UK
emma.anderson2@northumbria.ac.uk

Anastasiia Birillo
JetBrains Research
Belgrade, Serbia
anastasia.birillo@jetbrains.com

Ryan Dougherty[‡]
United States Military Academy
West Point, NY, USA
ryan.dougherty@westpoint.edu

Raymond Pettit
University of Virginia
Charlottesville, VA, USA
raymond.pettit@virginia.edu

Nuno Pombo
Universidade da Beira Interior
Covilhã, Portugal
ngpombo@ubi.pt

Ebrahim Rahimi
Open Universiteit
Heerlen, Netherlands
ebrahim.rahimi@ou.nl

Charanya Ramakrishnan
Macquarie University
Sydney, NSW, Australia
charanya.ramakrishnan@mq.edu.au

Alexander Steinmaurer
Interdisciplinary Transformation University
Linz, Austria
alexander.steinmaurer@it-u.at

Shubbhi Taneja
Worcester Polytechnic Institute
Worcester, Massachusetts, USA
staneja@wpi.edu

Muhammad Usman
Blekinge Institute of Technology
Karlskrona, Blekinge, Sweden
muhammad.usman@bth.se

Annapurna Vadaparty
University of California, San Diego
La Jolla, CA, USA
avadaparty@ucsd.edu

Abstract

Generative AI (GenAI) is playing an increasingly influential role in computing education across all levels, offering new opportunities to support both teaching and learning. However, its effective integration raises critical concerns related to trust, academic integrity, and broader social and ethical implications. While substantial attention has been given to GenAI use in introductory programming courses (e.g., CS0/CS1), there remains a notable gap in research addressing its application in "upper-level" computing courses, such as software engineering, human-computer interaction, algorithms, operating systems, and theoretical computer science.

This working group report presents two complementary studies: a systematic literature review of GenAI interventions in upper-level computing education, and a survey of computing instructors on their practices and perspectives regarding GenAI integration in these contexts. Based on the combined findings, this report presents an overview of current practice and practical guidance for computing instructors. The report is intended to inform the design of engaging, pedagogically sound, and forward-looking curricula that align with modern educational and workforce standards and expectations.

[*]The views expressed in this article, book, or presentation are those of the author and do not necessarily reflect the official policy or position of the United States Air Force Academy, the Air Force, the Department of Defense, or the U.S. Government. Approved for public release: distribution unlimited. PA#: USAFA-DF-2025-817
[†]Co-leader
[‡]The views expressed are those of the author and do not reflect the official policy or position of the US Army, Department of Defense or the US Government.

ITiCSE-WGR 2025, Nijmegen, Netherlands

ACM ISBN 979-8-4007-2167-0/2025/06
https://doi.org/10.1145/3760545.3783970

CCS Concepts

• **Social and professional topics → Computing education**.

Keywords

Generative AI, computing education, learning goals and outcomes

ACM Reference Format:
Dennis J Bouvier, Bruno Pereira Cipriano, Richard Glassey, Olga Petrovska, Emma Anderson, Anastasiia Birillo, Ryan Dougherty, Raymond Pettit, Nuno Pombo, Ebrahim Rahimi, Charanya Ramakrishnan, Alexander Steinmaurer, Shubbhi Taneja, Muhammad Usman, and Annapurna Vadaparty. 2025. The Rest of the Robots: Generative AI in Post-introductory Computing Education. In *2025 Working Group Reports on Innovation and*

Technology in Computer Science Education (ITiCSE-WGR 2025), June 27-July 2, 2025, Nijmegen, Netherlands. ACM, New York, NY, USA, 47 pages. https://doi.org/10.1145/3760545.3783970

1 Introduction

The coming of artificial intelligence has been predicted and discussed as far back as the 1950s [77]. Prior to 2020, each generation of artificial intelligence (e.g., expert systems) has gone through the typical technology hype-cycle, beginning with growing optimism that eventually fades as the promised potential transformation fails to materialize [30].

Along with the development of the transformer model architecture [119] came the latest generation of artificial intelligence (AI), commonly referred to as "generative AI" (GenAI). One prolific example involves "large language models (LLMs)", also known as transformer-based models. Since then, thousands of peer-reviewed scholarly articles have been published describing the impact of GenAI within various societal sectors including the economy, business practices, labor, politics, and many more. This report focuses on the sector of "computing education"; in particular the impact of GenAI on "post-introductory computing courses" (i.e., beyond the introductory level, see section 1.3 for definition). This report skips the "GenAI is coming" introduction and instead points the interested reader to the relevant citations for more background.

1.1 GenAI in Education

A recurring theme in the literature surrounding GenAI and education is that instructors must "do something" to incorporate GenAI into pedagogical practices. One suggestion is outright banning GenAI's use within the classroom as it can be an easily obtained inhibitor of the learning process. Without much effort on the part of the student, GenAI can be used to accomplish most typical student assignments from primary school level up to undergraduate courses in many academic disciplines [53], and assist with graduate level research [66]. Additionally, GenAI can be used by instructors to create assignments, assess and provide feedback on student work, and potentially detect students who cheat on assignments by traditional means.

As an example of how powerful GenAI has become, consider "The AI Scientist", a system developed by Sakana AI, which can develop a research idea "into a full paper at a cost of approximately $15"[1]. Note that Sakana AI is not offering The AI Scientist as a service, and no review of its output is included here; however, the relatively low cost for producing a paper with such ease indicates that students could leverage GenAI services to complete their school assignments with little student input and/or intervention, resulting in reduced student learning. The authors did not test Sakana AI; on the other hand, whole conferences have been conducted in which the paper authors and 'peer reviewers' are AI [45].

The authors of this report do not weigh in on the debate on banning GenAI in the classroom, except to say this: employers will likely not be banning GenAI in the workplace, making proficiency with GenAI a skill graduates will be expected to have soon, if not already. This "new reality" should inspire instructors to wonder what new learning experiences will prepare graduates for the new reality, a workplace in which appropriate and efficient use of GenAI is the norm.

[1] https://sakana.ai/ai-scientist/

1.2 Focus of this Report

This report is focused on assessing and coalescing the current state-of-the-art for integration of GenAI within post-introductory computing courses. That focus manifests in two forms: a systematic literature review of peer-reviewed publications involving GenAI as a pedagogical tool in these courses, and a survey of instructors of these courses to determine how GenAI is leveraged in their courses.

1.3 Working Group Nomenclature

In an attempt to make this report easier to understand, a list of terms used throughout the report is presented here:

Generative Artificial Intelligence (GenAI) - a class of artificial intelligence systems capable of generating new content, such as prose, code, explanations, images, and other types of content, based on patterns learned from training data. These systems, which include tools such as ChatGPT, Gemini, Microsoft Copilot, Claude, and others, leverage advanced machine learning models to create outputs that resemble human-produced content.

Large Language Model (LLM) - an AI system trained on large text datasets to generate human-like language. LLMs are the backbone of many GenAI tools used in education, enabling applications such as automated feedback, content generation, and language-based tutoring.

Introductory course - a course in which the course goal is for students to build proficiency in writing computer programs. The typical introductory course assumes the student has little, or no, prior programming experience; typically set in the 1st year of a degree program but could also occur in primary or secondary school. Such courses are often referred to as 'CS0' or 'CS1' [46].

Post-introductory course - a computing course that follows a introductory computing course in a curriculum; including CS2 (course content varies, but often includes sorting algorithms and data structures [46]), algorithms, database systems, operating systems, software engineering, theory of computing, capstone, and many others.

Learning Objective (LO) - a statement of what students should know at the completion of the course; similar to, but more specific than a course goal; also known as 'Intended Learning Outcomes (ILOs)'.

1.4 Working Group Scope

1.4.1 ***In Scope***. The scope of this report is focused on the impact of GenAI on student learning and learning activities in post-introductory computing courses, and uses of GenAI by instructors of these courses.

1.4.2 ***Out of Scope***. While the list below is not comprehensive, it helps define the boundary between what this report addresses and does not address. These topics are not necessarily less important, just not addressed in this report.

- Self-directed student use of AI (e.g., exam preparation)

- Student perception of AI
- Use of AI to detect unauthorized student use of AI
- Introductory programming courses
- Non-computing courses
- Ethics of AI, including:
  - Bias in GenAI models
  - Copyright infringement / plagiarism
  - Economic impacts
  - Employment impacts
  - Environmental impact
  - Social impacts (e.g., digital / AI divide)

1.5 Report Organization

The remainder of this report is organized as follows:

Section 2 outlines the report's goals and research questions.
Section 3 reviews related work, including prior studies on GenAI, post-introductory computing, and existing literature reviews.
Section 4 describes the working group's methodology, which combines a systematic literature review with a survey of instructors teaching post-introductory computing courses.
Sections 5 and 6 present the method for, and results of, a systematic literature review on GenAI use in post-introductory computing courses.
Sections 7 and 8 present the method for, and results of, an instructor survey on GenAI integration in post-introductory computing courses.
Section 9 discusses the main findings, implications, limitations, and recommendations.
Section 10 offers concluding remarks.

2 Working Group Goals

The main goals of this report are:

Establishing a reference standard for the current state of GenAI use in post-introductory computing courses so that instructors of post-introductory computing courses, and programs involving them, can determine appropriate GenAI incorporation.
Highlighting trends across different post-introductory computing courses and GenAI model incorporation.
Highlighting recommended areas for future research in pedagogically appropriate usage of GenAI models.

2.1 Research Questions

The following research questions guide the pursuit of the aforementioned goals:

RQ1. What research literature exists on the integration of GenAI into post-introductory computing courses in terms of subjects and activities?
RQ2. How have instructors integrated GenAI into their post-introductory computing courses?
RQ3. What are the higher-level trends on the use of GenAI within post-introductory computing courses in terms of novel activities, changes in assessment, skills development, usage policies and learning objectives?

3 Background

This section presents an outline of previous publications addressing GenAI across the computing domain, publications concerning interventions within post-introductory courses, and publications detailing a literature review of either of these.

3.1 Generative AI and Computing Education

Many scholarly publications at the intersection of GenAI and computing education report GenAI's *performance* on assignments (i.e., ability to complete assignments) within computing courses: across the curriculum [54, 78], introduction to programming (aka "CS1") courses [108, 125], computer engineering [104], computer graphics [35], and discrete mathematics [91]. Often, the purpose of these publications is to identify the strengths and weaknesses of the AI tools students can use so instructors can design "AI-resistant" assessments. Some research has been devoted to exploring AI to detect the use of AI (i.e., cheating detection where student use of AI is banned) [48].

Other research reports on students' 'relationships' with GenAI, such as students' attitudes toward and perceptions of GenAI use [16, 44, 105] and how much students use GenAI [49]. Liu [70] conducted a student and faculty survey in which they offered ethical guidance for GenAI incorporation based on their responses.

As mentioned in Section 1.4.1, this report is focused on *pedagogical* integration of Generative AI within computing education beyond introductory programming courses. So while the topics mentioned above are interesting and important (especially students' relationships with GenAI), they are out of scope for this report.

3.2 Prior Literature Reviews

Several literature reviews on the topic of 'GenAI and computing education' have been published despite GenAI having being developed only a few years before this report. The closest of these to this report is that of Raihan et al. [98] where they performed a systematic literature review of large language models and computing education in general. Other literature reviews considered these models within a general pedagogical context [2], teaching [15, 109], and student learning [50].

Other literature reviews survey literature on GenAI use within specific courses, for example: algorithms [69, 72], operating systems [32, 89], theory [27], and software engineering [107, 120]. Another survey on GenAI use in software engineering courses by Damayanti et al. [28] found a subtheme of GenAI integration within these courses, namely continually evolving curricular and policy adaptation.

This report is novel and distinguished from the above surveys in that (1) it is more recent and capturing more recent publications, (2) the scope is different in that this report considers post-introductory computing courses instead of all computing courses or a course-specific subset, and (3) the literature review is juxtaposed with results of a survey of computing instructors.

4 Working Group Methodology

The overarching methodology of this working group involves combining a systematic literature review with a survey of instructors teaching post-introductory computing courses. The initial literature review helps to identify some existing practices of incorporating GenAI into courses. The consequent survey aims to shed further light into existing practices that are not currently included in published literature.

Comparing the themes found in a literature review to the themes of a survey of computing instructors about their use of GenAI in courses can benefit the overall purpose of this study in several ways. The literature review provides a more systematic overview of existing studies and research-based initiatives in this domain. On the other hand, the survey is less systematic, but provides information on potentially unpublished, and potentially more current, teaching practices of computing instructors, offering a first-hand account of instructional decisions, trends, and patterns related to the use of GenAI in computing courses.

The combination of these two perspectives enables a deeper, more complete, and realistic understanding of what is happening in computing education regarding GenAI integration. In addition, the survey elicited input from computing instructors who have not integrated GenAI into their courses, to explore their reasons for not doing so and their potential plans for future integration. In this way, the survey results can supplement the literature review by revealing trends *and barriers* to GenAI adoption in computing education. The details of the methodologies employed for each of these input channels appear in their respective subsections.

5 Systematic Literature Review

The systematic literature review (SLR) aims to discover the current research on pedagogical integration of GenAI in post-introductory computing courses, addressing RQ1. As with the broader field of CSEd, research tends to be dominated by introductory level courses, for example: Becker and Quille [9], Luxton-Reilly et al. [73], and Valentine [118]. Therefore, this literature review distinguishes between introductory and post-introductory level courses; intentionally focusing on the latter. As stated before, the goal is to describe how instructors are incorporating GenAI into their teaching, and how students are engaging with the subject through GenAI-related activities within an area that may be underrepresented in the current literature.

5.1 Method

Inspiration for this report came from previous ITiCSE working group reports for framing a SLR within computing education, including: Prather et al.'s review of current trends in GenAI research, teaching practices, and tools [95], and Clear et al.'s scoping review and interview study on AI integration in the IT professional workplace [21]. Additional inspiration came from Kitchenham's broader procedures for conducting systematic reviews in software engineering [57].

The SLR workflow proceeded in four phases, outlined below and depicted in Figure 1. A detailed description follows.

Phase 1: (a) development and (b) validation of search string
Phase 2: execution of the search on academic databases and filtering by title, abstract and time frame
Phase 3: (a) categorization of papers and (b) application of inclusion / exclusion criteria
Phase 4: reading full-texts and synthesizing paper contents

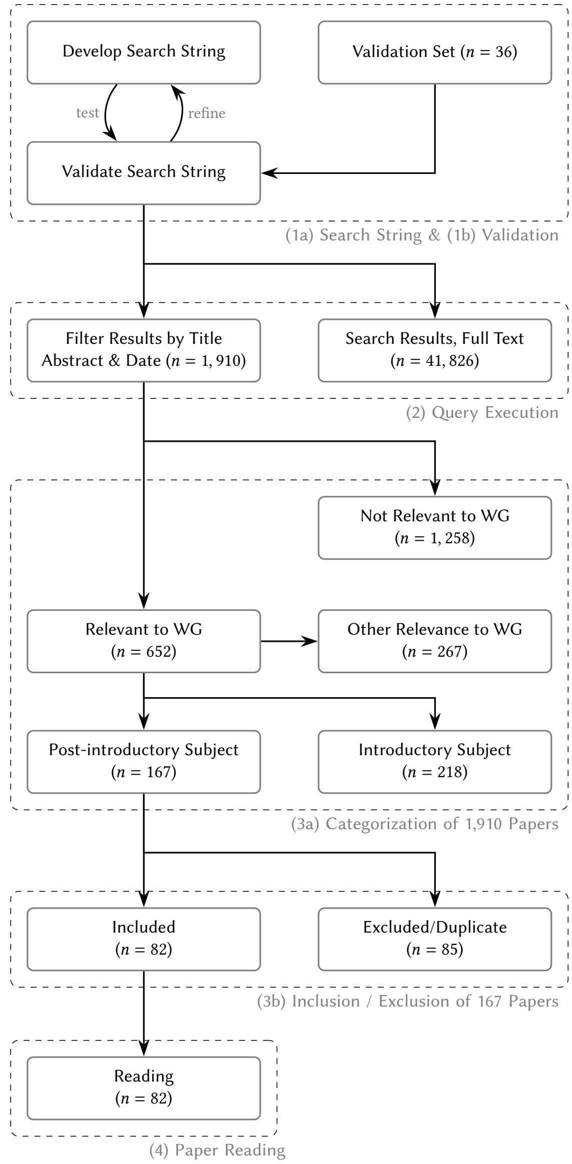

Figure 1: Workflow steps and result sets from systematic literature review.

5.1.1 ***SLR Phase 1a: Search String Development:*** In the first phase, the search string was iteratively developed. The initial search string iterations were informed by prior work [21, 95, 98], each of which presents a systematic literature review aiming to identify GenAI scholarship within the computing education domain. These studies include dedicated components in their search strings to capture a broad range of GenAI-related terminology. Examining the overlaps and variations among these formulations provided a foundation for refining and strengthening our own query.

The main challenge was in capturing papers relevant to an unknown set of post-introductory subjects (e.g., database systems, operating systems, software engineering, etc.) in search results.

Various strategies were tried, including: (1) enumerating subjects based on commonly accepted subject areas in CS (e.g., the ACM CS 2023 Curricula [60]) – this resulted in unwieldy search strings that challenged the database search engine interface limitations (i.e., maximum of 50 terms); (2) using negation filters to isolate post-introductory by excluding introductory level results – this yielded results that did not comprehensively exclude introductory computing.

Another search term tried, and later abandoned, was explicitly targeting "CSEd" and closely related ways of stating our subject area in the abstract. Through these iterations, it became apparent that the papers sought did not commonly use "CSEd" as a term in the abstract or paper keyword; however, the publication venue could be used as an indicator of CSEd. Thus, the alternative strategy adopted was to include a rich set of pedagogical aspects that tilted the results towards various educational contexts, without needing to specify or omit specific courses. This yielded the following search string:

Subject Domain: "computer science" OR "computer engineering" OR "software engineering" OR "computing education" OR "cs education" OR "csed" OR "cse"

Generative-AI: "generative" OR "large language model" OR "large language models" OR "llm" OR "llms" OR "gpt" OR "gpt-3" OR "gpt-3.5" OR "gpt-4" OR "gpt-4o" OR "o1" OR "o3" OR "chatgpt" OR "openai" OR "gemini" OR "bard" OR "claude" OR "copilot" OR "llama" OR "mixtral" OR "deepseek" OR "codex"

Pedagogical Aspects: "education" OR "teaching" OR "pedagogy" OR "student" OR "students" OR "learner" OR "learners" OR "teacher" OR "teachers" OR "curriculum" OR "course" OR "courses" OR "course design" OR "assignment" OR "homework" OR "project" OR "capstone" OR "coursework" OR "assessment" OR "grading" OR "examination" OR "exam" OR "learning outcome" OR "learning outcomes" OR "learning objective" OR "learning objectives" OR "competence" OR "competency" OR "competencies" OR "policy" OR "policies"

The resulting string is the following query, with scope restrictions for the search engines denoted square brackets:

```
Subject Domain [anywhere] AND Generative-AI [abstract]
AND Pedagogical Aspects [abstract]
```

The rationale for applying different scope restrictions is as follows. A broader scope was used for the **Subject Domain** component to ensure that the captured papers were situated within computing: although domain-relevant terms may not appear in the abstract, they often surface elsewhere in the paper. In contrast, a narrower scope was applied to the **Generative-AI** and **Pedagogical Aspects** components. This decision was informed by the observation that papers relevant to these areas typically signal their focus explicitly in the abstract, making a more restrictive search both appropriate and effective.

Multiple digital libraries were compared for suitability and consistency of executing the search string. Whilst we are aware of the emerging popularity of arXiv, and comprehensive indexing of Google Scholar, the decision was made to rely on the established and trusted databases of ACM Digital Library and IEEE Xplore.

Table 1: Validation set paper by digital library and count with percentage found by search string.

Library	DOI Prefix	Count	Found
ACM Digital Library	`10.1145/*`	24	(22 of 24) 92%
IEEE Xplore	`10.1109/*`	8	(5 of 8) 62%
		36	(27 of 36) 75%

5.1.2 ***SLR Phase 1b: Validation of the search string:*** Developing a validation set (also referred to as a *quasi-gold standard* or a *known-relevant corpus*) provides an objective way to assess the sensitivity of a search strategy. In computing-education research, where relevant work is dispersed across computer science, education, learning sciences, and engineering venues, the total population of relevant studies is never fully knowable. The validation set therefore serves as a proxy benchmark for determining how effectively the automated search retrieves known-relevant work [63, 126]. Because comprehensiveness is essential to avoid systematic omission of relevant studies, the validation process focuses primarily on maximizing search sensitivity before optimizing precision.

To construct the validation set, the team identified a group of papers expected to be found by the search. The initial list was compiled collaboratively by the research team, drawing on prior familiarity with the field and a preliminary scan of influential publication venues (e.g., SIGCSE TS, ITiCSE, ICER, and ACM TOCE). The list of papers identified by the team was augmented with the relevant post-introductory papers found in [98], a recent SLR on a related theme. As the plan was to query only the ACM DL and IEEE Xplore, the validation set was filtered by the respective DOI prefixes for those two libraries, resulting in a final set of 36 papers.

The resulting collection functioned as a validation set against which to test the performance of the automated search. After each iteration of search-string refinement, the percentage of validation papers found by the query was computed. When retrieval fell below the predefined threshold (70% of validation papers[2]), the search terms, Boolean structure, or database coverage were expanded accordingly. This iterative process ensured the final search configuration could reliably recover key studies across the diverse publication venues and terminological variations characteristic of computing-education research.

Table 1 shows the performance of the search string against the validation set; despite the difference in return rates (92% for ACM DL and 62% for IEEE Xplore), the overall success of 75% surpassed the criterion (i.e., 70%) for acceptable query performance. The validation set is included in Appendix A as a list of titles for reference to either replicate this work or provide a starting point for future research.

5.1.3 ***SLR Phase 2: Query Execution:*** Conducting a full-text search with the search string identified over 41,000 papers. The

[2]It was observed that papers that should have been included due to their fulltext would never have been matched with any search string tweaking, thus it was decided to settle on 70% as a fair threshold.

Table 2: Summary of paper sets by category.

Category	Count
Search String applied to Anywhere	41826
Search String filtered by Title and Abstract	1910
Not Relevant to WG	1258
Relevant to WG	652
— Other Relevance to WG	267
— Introductory	218
— Post-introductory	167
— Excluded / Duplicate	85
— Included	82

decision was made to proceed with the subset of 1,910 papers identified by restricting search fields to title and abstract, and within the time period from January 2022 to June 2025.

5.1.4 ***SLR Phase 3: Categorization:*** Each WG member randomly chose a subset of the 1,910 papers, and judged them based on the title and abstract if the paper was:

Not Relevant for WG – False positive results to be discarded.
Other Relevance for WG – Interesting works, such as other literature reviews, policy, vision and so on that might form important background material.
Lower-level computing – Focus on introductory/CS1/CS0 level course.
Post-introductory computing – Focus on course recognized to follow the introductory course.

After this initial pass, WG members made a second judgment to verify if the introductory and post-introductory papers were as such. In cases of of disagreement WG leaders resolved the disagreement by taking a closer look at the paper and providing a final decision with rationale for transparency of decision making among members. Table 2 shows the final counts for each of these paper sets.

5.1.5 ***SLR Phase 4: Reading Full Texts:*** As the post-introductory set had arrived at a tractable size for the WG members to read, the decision was made to advance to full reading of papers and strict application of the following inclusion and exclusion criteria:

Inclusion Criteria:

IC1: Paper targets post-introductory subject(s)
IC2: Paper explores the use of generative AI within the subject
IC3: Paper explores the use of generative AI by students, teaching assistants, or instructors

Exclusion Criteria:

EC1: Paper not about generative AI
EC2: Paper not in computing education domain
EC3: Paper only targets introductory subjects (e.g. CS1/CS0/K12)
EC4: Paper only focused on using GenAI to generate course material in a generic/non-novel way
EC5: Not a research article (e.g. books, blogs, presentations, panels, abstracts etc)
EC6: Less than 4 pages single column, 3 pages double column
EC7: Published before 2022

Table 3: Count and distribution of GenAI studies by subject from post-introductory literature.

Subject	#	Distribution
Software Engineering	30	
Databases	10	
Human Computer Interaction	10	
Algorithms and Data Structures	8	
Data Science and Visualization	5	
Object-oriented Programming	5	
Security	4	
Machine Learning	3	
Networks	2	
Theoretical Computer Science	2	
Distributed Systems	1	
Operating Systems	1	
Web Development	1	

EC8: Language is not English
EC9: Publication not within an ACM or IEEE-sponsored venue

For papers identified as other relevance or not relevant, an automated sanity check was applied[3] to label as relevant or not relevant and why; papers detected as being relevant to post-introductory or introductory level were checked manually by WG leaders and moved to the appropriate result set.

As a final note, the set of introductory level papers ($n = 218$) are included as a list of DOIs in appendix B. Whilst these papers were not used as input for this report, they are listed here for the community to serve as a starting point for an introductory level review, or as a source for papers to validate future reviews.

6 Results from the Literature

The results of the SLR are organized into four parts in the following sections.

- **Sec 6.1** a high-level summary of GenAI use as found in the literature, grouped by subject
- **Sec 6.2** a general taxonomy of aspects of GenAI use that emerged from analyzing activities across all subjects
- **Sec 6.3** integrated findings that compare how the aspects of activities varies between subjects
- **Sec 6.4** a detailed, subject-by-subject analysis of GenAI use

6.1 Summary of GenAI Activities by Subject

Working group members first organized included papers into subject clusters using common computing courses. Then, sub groups focused on each cluster, correcting any misallocation if discovered on the second pass. Table 3 summarizes the distribution of GenAI use across subjects in post-introductory courses. Software engineering accounts for the largest share ($n = 30$), followed by database systems ($n = 10$), human–computer interaction ($n = 10$), algorithms and data structures ($n = 8$), data science and visualization ($n = 5$), and object-oriented programming ($n = 5$). Smaller numbers are reported in security ($n = 4$), machine learning ($n = 3$), networks

[3] See: https://github.com/rjglasse/iticse-2025-wg2

($n = 2$), and theoretical computer science ($n = 2$), with distributed systems, operating systems, and web development, each appearing once ($n = 1$).

6.2 Taxonomy of GenAI Use in Activities

During the analysis of the literature, recurring aspects of GenAI use emerged across subject specific activities. These activities ranged from simple, single-aspect instances to complex, multi-aspect uses. To enable comparison of GenAI use between subjects and to identify potential areas for further exploration, the following taxonomy of aspects was developed:

Generate — Creating new artifacts, outputs, or ideas using available tools or resources with GenAI.

Interpret — Making sense of information, artifacts, or situations to extract meaning or understanding with GenAI.

Evaluate — Assessing the quality, accuracy, or suitability of an artifact, idea, or process generated by GenAI.

Get Feedback — Receiving a review, assessment or feedback from GenAI upon one's own work.

Refine — Iteratively improving or modifying an existing or generated artifact, idea or process based on feedback or new insights from GenAI.

Brainstorm — Supporting and inspiring ideas and approaches towards a problem or objective using GenAI.

Design — Planning and structuring a solution, process, or system to meet identified goals or needs.

Simulate — Emulating scenarios and experiences for exploration, practice, or testing using GenAI as facilitator.

Reflect — Reflecting upon the higher-level implications of GenAI use within the subject.

6.3 Analysis of GenAI Use by Subject

Table 4: Percentage of GenAI aspects across all activities.

Aspect of GenAI Use	%	Distribution
Generate	33	
Evaluate	17	
Interpret	16	
Refine	9	
Get Feedback	7	
Reflect	7	
Brainstorm	4	
Simulate	4	
Design	2	

To establish an overall picture of emerging usage patterns, Table 4 presents a summary of aspects of GenAI use across all subject areas, aggregated into percentages based on the taxonomy defined in the previous section. This provides a high-level view of which aspects are most frequently observed in the literature.

To add disciplinary context, these counts are broken down by subject area. Figure 2 visualizes this breakdown as a heatmap, illustrating how the prominence of specific aspects varies across subjects. Across subjects, the most common use of GenAI was to generate artifacts, with *Generate* activities dominating in nearly all areas, particularly in Software Engineering, Databases, Human–Computer Interaction, Object-Oriented Programming, Machine Learning, and Theoretical Computer Science. By contrast, *Evaluate* and *Interpret* appeared as secondary but widely distributed practices, varying in strength depending on the subject. More iterative and creative forms of use, such as *Refine*, *Get Feedback*, *Brainstorm*, *Simulate*, and *Design*, were comparatively rare and appeared only in specific contexts. This suggests that while GenAI is broadly adopted as a tool for production, its potential for supporting feedback, reflection, and creative iteration remains underutilized outside a few design-oriented disciplines. However, these percentages are heavily influenced by the amount of studies found per subject, which must be kept in mind.

Having established both the aggregate and subject-specific patterns of GenAI use, the following sections examine each subject area in more detail. This closer analysis is supported by a series of tables summarizing common activity types within each subject, described using the taxonomy of GenAI use.

6.4 Software Engineering

There are 30 papers in our result set that focus on Software Engineering (SE). Of these, 25 report on students' use of GenAI to complete course tasks. In most cases, instructors explicitly required students to use GenAI in addition to performing the task manually. Three papers describe instructors' use of GenAI to evaluate its appropriateness for coursework. Another three papers report where both students and instructors engaged with GenAI tools. A summary of these activities and their GenAI aspects is provided in Table 5.

Due to the relatively large number of software engineering (SE) papers, they were grouped into six specific topics areas: requirements engineering, software design and modeling, software development and engineering in general, software testing and quality assurance, project management, and software capstone, and team project courses. The following sections describe the findings for each SE topic group, highlighting both positive and negative experiences of using GenAI. The distribution of papers across these six groups is shown in Table 6.

6.4.1 ***Requirements Engineering****.* Four papers in our sample report experiences of using GenAI in courses related to Requirements Engineering. In these studies, students used GenAI to

- Generate or refine user stories [13, 100]
- Identify inconsistencies in natural language (NL) requirements [33]
- Trace requirements to their sources [23]

In three of these papers [13, 23, 33], the same tasks were also performed manually, allowing for a comparison between GenAI and human results. In a fourth paper, the user stories produced with the support of ChatGPT and other LLMs were reviewed by experts to assess their quality.

For inconsistency detection in requirements [33] and requirements traceability [23], ChatGPT did not outperform humans. When generating or improving user stories, the use of ChatGPT and other

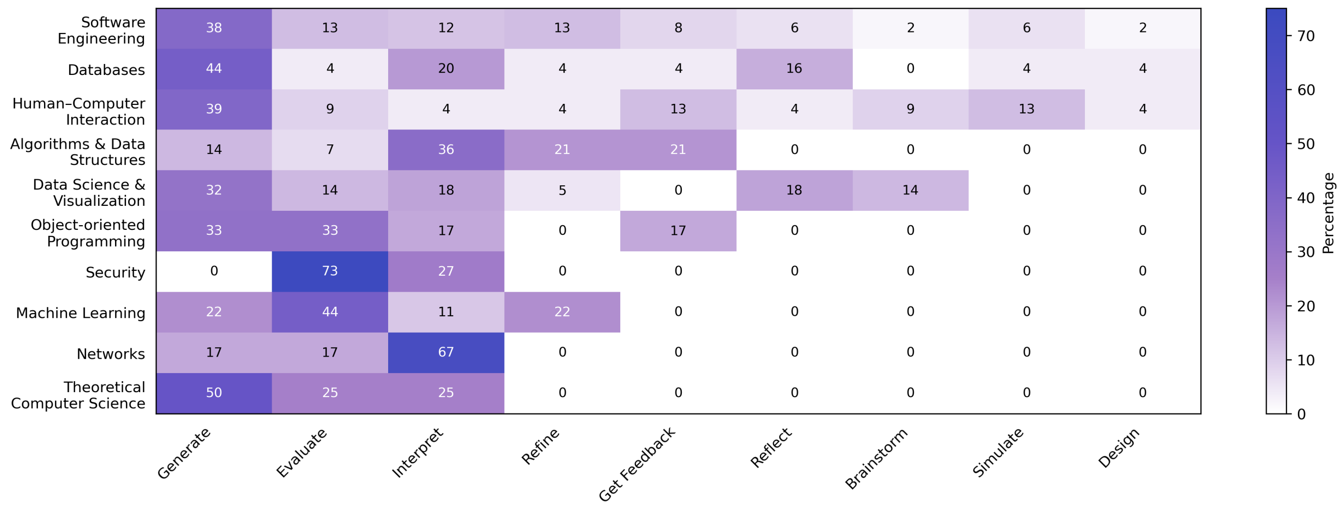

Figure 2: Heatmap of GenAI Aspects from activities occurring across different subjects, normalized by percentage.

LLMs by students generally resulted in higher-quality user stories [13, 100]. However, in one case, ChatGPT struggled with producing independent user stories, as it sometimes failed to identify and group related user feedback comments, which served as the input [13].

ChatGPT 3.5 was used in three requirements engineering papers [13, 23, 33], while one study also included ChatGPT 4.0, which showed comparatively better results [23]. A majority of students in one course expressed satisfaction with the quality of ChatGPT's responses for generating user stories [100]. Upon reflection, tudents reported that ChatGPT enhanced their problem-solving skills and supported their preparation for real-world software development.

Across multiple studies in requirements engineering, the authors recommend using GenAI as a supplementary aid for students and instructors, rather than as a replacement for tutor support [23, 33].

6.4.2 ***Software Modeling and Design.*** Five papers [5, 26, 29, 38, 121] explored the use of GenAI tools in software modeling and design courses. Four of these focused specifically on creating UML diagrams. Several studies adopted the diagram generation tool PlantUML[4] as a way to produce visual output from GenAI, leveraging its text-based input to render AI-supported modeling [5, 26, 121]. In these contexts, students used GenAI to generate initial drafts of UML class, use case, or sequence diagrams, often as part of formative or exploratory modeling tasks [5, 26, 29, 121].

While the generated outputs were generally syntactically correct and pragmatically useful, the authors consistently noted semantic shortcomings, including incorrect relationships, misused multiplicities, and inconsistent naming [26, 29]. Both students and instructors emphasized the importance of revising and critically reflecting on LLM-generated diagrams, which served not as final solutions but as scaffolds for deeper learning and model refinement. The educational value of this approach lies in its ability to promote iterative modeling, encourage meta-cognitive engagement, and strengthen students' understanding of modeling semantics through active correction and refinement.

Another reported use of GenAI for software design came from a course on enterprise design patterns [38]. Here, students engaged in structured prompting tasks with ChatGPT to explore selected patterns and reflect on their understanding. They were tasked with generating meaningful prompts, evaluating responses, and documenting their learning in experience reports and exam questions. Although students initially struggled to formulate higher-order prompts, this interaction fostered deeper engagement with design concepts and highlighted the pedagogical value of guided prompting. Overall, ChatGPT was considered helpful in scaffolding conceptual understanding but less capable of providing nuanced or contextually precise insights without structured support. These findings suggest that while LLMs can facilitate access to design knowledge, they are most effective when embedded within structured, reflective learning activities.

6.4.3 ***Software Development and Engineering.*** Seven papers describe GenAI interventions within courses referred to as software engineering or software development [11, 22, 39, 42, 79, 111, 122]. Some papers related to software development focused on altering the approach to [22], or content of, the assessment [39]. The others focused on learning and teaching interventions.

Some interventions focused on a specific software engineering activity (e.g., refactoring [79]), or a particular context (e.g., working with large code bases [111]). It can be challenging for students to fully appreciate and understand the relevance of this type of activity before they experience it in the workplace, where GenAI tools can be beneficial. In [79], students refactored code with and without GenAI to critically assess its suggestions. This approach fostered skills such as abstraction, problem formulation, and decomposition.

In multiple studies the motivation for the inclusion of GenAI tools was to provide authentic learning for post-introductory students to prepare them for the types of task they may encounter in the workplace [22, 42, 111, 122]. Clift and Petrovska [22] emphasize

[4]https://plantuml.com/

Table 5: GenAI activities found in SE literature.

GenAI Use	Illustrative Activities	#
Generate	Generate code (capstone/solo projects); Generate tests/unit tests; Generate UML diagrams/class diagrams; Generate user stories; Create documentation, exercises, project artifacts; Support agile process (e.g., user stories + code suggestions) *Instructor:* Create intentionally inconsistent requirements; Generate UML diagrams; Generate requirements from interview transcripts	17
Evaluate	Compare algorithm creation effectiveness; Evaluate generated testing materials; Evaluate inconsistent requirements; Judge quality of LLM test/-mocking suggestions *Instructor:* Evaluate generated UML diagrams; Evaluate students' planning poker estimates; Evaluate generated requirements from interview transcripts	7
Refine	Refine project backlog; Improve project management artifacts; Perform and improve code refactoring; Improve UML class diagrams; Refine user stories; Improve weekly stand-up reports	7
Interpret	Understand and navigate large code bases; Use for code comprehension; Learn about design patterns; Identify refactoring opportunities; Understand testing concepts; Use of knowledge assistant in capstone projects	6
Get Feedback	Receive feedback on UML/software models; Get feedback on design/code; Determine quality of code reviews; Generate code reviews	4
Reflect	Reflect upon GenAI use in projects (e.g., software dev, escape room); Reflect upon generated code reviews	3
Simulate	Interact with simulated team roles; Simulate team member in poker planning; Use GenAI as assistant in scrum process	3
Brainstorm	Clarify problems in finite state machine modeling	1
Design	Support GUI design	1

Table 6: Breakdown of the Software Engineering cluster.

Software Engineering Topic	#
Software Development and Engineering in General	7
Software Testing and Quality Assurance	6
Project Management	5
Software Design and Modeling	5
Requirements Engineering	4
Team and Capstone Project	3

the importance of preparing learners for technologies currently used in the SE industry by integrating GenAI technologies into the curriculum. Their students agreed their experience of working with GenAI would be valuable for their future careers.

Wang et al. [122] noted that "the complex concepts and interactive team activities involved in the SDLC pose significant challenges for students to learn and instructors to teach at scale". They created GenAI agents with personas representing roles from a software development team. Students could seek guidance, practice skills, ask questions, and receive real-time feedback from the agents. They can gain practical insights into how each decision and action influences the overall software development process. Use of GenAI here efficiently supports large-scale student learning.

Clift and Petrovska [22] and Borghoff et al. [11] gave learners free choice to use GenAI tools as they wished while working on a software development project. This allowed students to decide "if, when and how useful GenAI would be in a development situation." Clift and Petrovska [22] noted a positive trend between learners being given freedom to explore GenAI tools usage and student enjoyment and project quality.

Gorson Benario et al. [42] discussed the changes to the course objectives of a Software Development course to reflect updated software engineering practices which may now utilize GenAI tools. Prior to formal instruction as to how GenAI tools can support SE activities, students reported their most frequent use of GenAI tools was for learning new concepts. After receiving training throughout the semester, the authors found the students were more likely to understand, and actively engage with, the code outputs of the GenAI tools. The students reported the experience with GenAI tools was positive and assisted their development of programming and problem solving skills [42].

Use of DevCoach for Simulating Development Team Roles (Direct Quote from Wang et al. [122])

"When the user lands on the "Plan and Design" phase page, they would have a chat interface with a text box at the bottom where they could type their messages, and the area above it would display the history of messages.
The Team Lead greets the user and starts the design discussion for a movie database and review system. The relevant persona would provide the appropriate response depending on the user's message. For example, if the user asks, "What are the functional requirements for the project?", then the Product Manager would respond as they are responsible for any discussion related to project requirements, user experience, market needs, and business objectives.
Meanwhile, if a user asks about which programming language to use, the Senior Developer would respond as they are responsible for technical specifications, system design, data modeling, and technical feasibility of proposed features.
The Team Lead would respond to messages concerning project timelines, resource allocation, team coordination, and adherence to technical standards and best practices.
After discussion with the generative agents, the phase is deemed complete once the user comes up with a clear set of user stories of functional and nonfunctional requirements".

One study [39] involved students learning how to integrate large language models (LLMs) into a software system. Students were provided with a Java-based API that connects with OpenAI's GPT model. Students learned to manage LLM API calls, and how to construct effective LLM prompts to create 'intelligent and user-friendly applications'. The students were provided with simple starter code, allowing them creativity in extending the application. The authors were impressed with several ambitious applications developed by students with limited prior programming experience. Key themes noted in the student feedback include: engagement and enjoyment, creative freedom and autonomy, and challenges related to learning new concepts and technologies. The students reported that "they appreciated the opportunity to work with cutting-edge AI technologies while reinforcing their programming skills".

6.4.4 ***Software Testing and Quality Assurance.*** Six papers report GenAI interventions within software testing courses: [24, 43, 55, 71, 80, 117]. Students were the users of GenAI in this cluster except in Crandall et al. [24]. Most used a ChatGPT model, either 3.5 or 4; one used Bard-AI [117]. Crandall et al. [24] employed their own GPT-based system integrated into a custom Automatic Review Tool (ART). All papers were experience or evaluation-related interventions. Only two papers had explicit GenAI learning outcomes [43, 117]. The interventions had similar themes:

- Generating software testing artifacts more easily [43, 117]
- Assisting with software testing practices, such as debugging [71, 80] and code review [24, 55]
- Verifying the output software testing artifacts of GenAI models manually [43]

None of the authors of these papers report a change in the structure of the software testing course. Overall, the effectiveness and perception from students in these studies were mixed. In all cases the authors reported positive student responses of how the GenAI model was able to solve "basic" problems related to software testing, such as basic test generation, and how it can speed up existing tasks. Generally, the authors also reported negative student responses about how the models were not able to handle corner cases and more "formal" concepts such as finite state processes. Mezzaro et al. [80] reported decreased performance with GenAI usage.

6.4.5 ***Project Management.*** Five papers report GenAI use Project Management courses, including one integrating GenAI into learning activities [12] and three that use GenAI for assessments [59, 82, 88]. Key areas of focus include:

- Agile methodology, particularly Scrum, [12, 88]
- Project management practices [59]
- Planning poker estimation [81, 82]

These papers mainly aim at enhancing students' understanding and engagement in the subject matter while improving overall learning outcomes.

A recent study shows the effective integration of two GenAI tools, ChatGPT and Bard, within educational workshops [12]. These workshops leverage instructor-guided prompt interactions to convey key concepts of agile methodology, with a specific focus on the Scrum framework. In two courses, students used GenAI in their assessments to generate project management artifacts and apply project management practices in the development of applications [59, 88]. Notably, the instructors in one course restricted the use of GenAI to generating unit tests to prevent students from over-relying on the tool [88]. Furthermore, another prominent application of GenAI integration is its use in planning poker estimation, a technique used in agile planning and estimation processes. GenAI has been integrated as a learning tool for students [82] and as a means to assess the effectiveness of leveraging GenAI in evaluating student contributions during planning poker sessions [81].

The integration of GenAI in project management courses has been shown to positively impact students' learning and motivation levels. Research highlights that the incorporation of GenAI in projects management courses enhances students' comprehension of complex concepts: 97% of students reported an improvement in their understanding of essential concepts, such as the Scrum framework, when GenAI was included in their workshop lessons [12]. Additionally, 81.8% of participants indicated that allowing unrestricted access to GenAI throughout most of the course facilitated their learning process [88]. Other positive impacts identified include enhanced student motivation in utilizing the tool [59, 88], increased engagement with the content [12, 82], improved efficiency in task execution [82, 88], as well as advancements in problem-solving [82] and critical thinking skills [88]. Additionally, the papers also report a notable improvement in teamwork and collaboration [82, 88]. While students recognized GenAI as a valuable educational supplementary tool, they also highlighted the essential role of human instructors [12].

6.4.6 ***Software Capstone and Team Projects.*** Three studies explored the integration of GenAI tools in capstone and team-based software engineering projects, examining their role in improving productivity, communication, and learning outcomes [85, 86, 102]. Collectively, these studies show that GenAI can support student teams, particularly in early development phases and for routine communication and reflection tasks.

Rasnayaka et al. [102] conducted a semester-long project course in which students were encouraged to use GenAI tools such as ChatGPT and GitHub Copilot to assist with code generation, debugging, and algorithm design. The analysis of the annotated code and the students' prompts showed that GenAI usage was most frequent at the beginning of the project, supporting early scaffolding. More experienced students used GenAI more effectively, highlighting the need for foundational coding and prompting skills.

Neyem et al. [86] presented an AI knowledge assistant tool utilized in a capstone project course integrated into a Kanban project management system. This tool features a chatbot that allows students to inquire about course assignments, project guidance, and documentation. It retrieves recommendations from a local repository of lessons learned and utilizes queries as contextual information in prompts for GPT-4 and LLaMa. Additionally, the tool searches Stack Overflow for further recommendations. Feedback indicated that students favoured recommendations from LLMs over those from the local repository, while they found the external Stack Overflow recommendations less beneficial. Overall, students reported a positive experience using the tool.

6.4.7 ***Positive Sentiments.*** A positive theme across the six topics of SE is the supporting and scaffolding role of GenAI. In requirements engineering, studies reported that students using GenAI to generate or improve user stories produced outputs that were generally better than manual attempts, with students expressing satisfaction with the quality of generated user stories [13, 100]. In modeling and design courses, GenAI was helpful in producing drafts of UML diagrams as starting points for students to further reflect and refine them to improve their modeling practices [5, 26, 121].

In software development courses, students valued GenAI for its ability to prepare them for future careers. For example, code refactoring tasks with GenAI support helped foster computational thinking skills [79], role-based GenAI agents allowed students to practice communication and decision-making throughout the software development lifecycle [122], and GenAI tools helped students develop their programming and problem solving skills [42]. In testing and quality assurance courses, students reported that GenAI tools were able to solve basic testing problems, such as generating simple tests, which saved time and reduced repetitive work [43, 117].

In project management courses, GenAI was found to be useful in improving students' understanding of Scrum [12]. GenAI also had a positive impact on students' motivation, engagement, teamwork, and collaboration, with several students noting that unrestricted access to GenAI supported their learning process [82, 88]. Finally, in capstone and team project courses, GenAI tools provided valuable early scaffolding in activities such as code generation and debugging [102]. The students in another project course benefited from the integration of an AI knowledge assistant into the project management system, finding AI recommendations more helpful than other sources [86].

6.4.8 ***Negative Sentiments****.* Despite the aforementioned positive sentiments, some concerns have also been shared in the reviewed literature about the effectiveness of GenAI in software engineering courses. A recurring theme is that GenAI struggles with advanced or nuanced tasks.

In requirements engineering, ChatGPT was not better than humans at detecting inconsistencies in requirements or tracing them to their sources [23, 33]. When generating user stories, it sometimes failed to produce independent user stories [13]. In modeling and design courses, although the draft UML diagrams were useful, they frequently suffered from semantic errors, such as incorrect relationships, misused multiplicities, and naming inconsistencies [26, 29]. In testing and quality assurance courses, while GenAI could handle routine cases, students reported that it failed with corner cases and more formal testing concepts, such as finite-state processes [71, 80].

6.5 Databases

Ten papers report GenAI interventions within database courses. Across these ten papers, instructors identified several critical areas where incorporating LLMs into database systems instruction. Table 7 summarizes the GenAI activities and aspects of this cluster.

6.5.1 ***Assessments Integration:*** One of the most common ways GenAI tools were used in database courses was to assist students with various assessments.

In one study, students interacted with a custom GenAI tool simulating a client persona. The persona was knowledgeable about user stories but lacked understanding of the database specifications.

Table 7: GenAI activities found in Database literature.

GenAI Use	Illustrative Activities	#
Generate	Generate queries; Force incorrect SQL solutions; Generate visual step-throughs with query explanations; Generate personalized database schemas; Use GenAI to complete assignments; Summarize and generate report on deadlock scenarios; Generate practice questions based on lecture content; Generate synthetic data to populate the database; Assist in writing queries *Instructor:* Automatic generating database schemas and SQL questions; Generate pop quizzes	11
Interpret	Explore prepared database schemas; Implicit Query Execution with GenAI interface; Use GenAI to understand assignments; Analyze and discuss provided deadlock scenarios; Use to answer questions during lectures	5
Reflect	Reflect on generated outputs vs. student solutions; Guide self-reflection by affirming understanding and prompting thought; Reflect on use of LLMs for assignments; Assess the effectiveness of ChatGPT in facilitating database tasks	4
Evaluate	Analyze incorrect SQL solutions	1
Get Feedback	Provide feedback on Enhanced Entity Relationship Diagrams (EERs)	1
Design	Devise the SQL schema	1
Refine	Improve Entity Relation (ER) diagrams	1

The positive outcome of using this tool was that students produced high-quality ER diagrams, demonstrating the tool's effectiveness in enhancing their engagement and comprehension [3].

Most of the studies focus on incorporating GenAI access into assessments, either in completely [34, 61, 62, 65, 103], or in specific parts of the assessment [101]. These studies emphasize assessing various concepts, including the creation of SQL tasks and reflections on them [34, 62, 65], as well as theoretical database concepts related to deadlocks and transactions [61, 101, 103]. All these studies also indicate moderate to positive feedback in their integration into the assessments.

Another study focused on offering detailed feedback to students regarding the design of Enhanced Entity Relationship Diagrams (EERs). The research involved modifying the AI system to provide targeted feedback on individual relationships, ensuring the feedback more closely aligned with instructors' expectations [103]. The result notes that 84% of participants reported the feedback was beneficial in refining their contextual diagrams.

6.5.2 ***Chatbots:*** A couple of studies focus on chatbots. One proposes a framework for successfully implementing chatbots in database courses [92], while the other involves using a chatbot to answer questions during lectures and generate practice questions based on the content covered in those lectures [84]. The latter study had a favorable acceptance rate among students with positive sentiments as well.

6.5.3 ***Generation of Questions:*** Only one of the ten papers in this cluster focuses on instructors. This study aimed to analyze the impact of GenAI models, particularly ChatGPT-3.5, by identifying effective prompts to determine whether it can generate database schemas and SQL questions for instructors [1].

6.5.4 ***Positive Sentiments:*** Most studies report positive sentiments regarding the integration of GenAI in database courses. When personas were used, students engaged in critical discussions and sought information rather than relying on GenAI for quick solutions [3]. Students were asked to critically assess ChatGPT's outputs in one study; 70% of respondents reported it provided clear and constructive feedback [34]. All 58 students who participated in the activity passed the final exam, contrasting sharply with only 31% of those who did not participate. This outcome indicates a positive correlation between the engagement activity and exam success. This aligns with another study showing that students who participated in a reflective assessment based on a language model performed better on a proctored exam two weeks later compared to those who only reviewed lecture slides [62].

Another related study reports a positive impact of continuous engagement with LLMs during homework exercises. A comparison of homework scores showed that students who used problem-solving strategies independently performed worse than those who engaged continuously with LLM assistance; suggesting that independent problem-solving may be less effective for homework performance [61].

In the context of generating SQL tasks for assessments, students found ChatGPT particularly helpful for creating sample data to populate database tables [65]. When chatbots were customized to address specific lecture questions and ignore irrelevant topics, it resulted in a positive reception [84].

6.5.5 ***Negative Sentiments:*** Some negative perceptions were observed in the studies. While students viewed the chatbot as a tool that enhanced their educational journey, they still preferred human tutors. There were some accuracy issues in the chatbot's explanations, which sometimes differed from the provided notes [84]. In assessments, students noted that the chatbot's effectiveness diminished when it came to more complex tasks, such as designing database schemas and formulating difficult SQL queries [65]. Although ChatGPT can help reduce exam preparation time, some instructors remain skeptical about using large language models (LLMs) due to the need for verification [1]. In the study where GenAI was allowed partially in one of the tasks, findings indicated that students performed better on those tasks but faced challenges in other areas of the assignment and the overall course. In contrast, students who abstained from GenAI tools engaged more actively with the discussion board content and participated more than their peers who used ChatGPT. These findings suggest that unsupervised use of GenAI may not effectively enhance skill development or material comprehension and could reduce engagement [101].

6.6 Human Computer Interaction

Ten papers report GenAI interventions within human-computer interaction (HCI) courses. The interventions used across these courses were generally targeted toward students (seven of the nine papers), though there were instances of instructors using GenAI tools as well. The interventions broadly assisted with handling three core elements of HCI: user personas, creating designs and prototypes, and feedback on the design. Table 8 summarizes the GenAI activities and aspects of this cluster.

Table 8: GenAI activities found in HCI literature.

GenAI Use	Illustrative Activities	#
Generate	Generate scenarios; Generate persona, avatar and knowledge graph; Generate images for mood board; Generate visual tangible design; Generate storyboard *Instructor:* Generate application mockups using Uizard	9
Get Feedback	Get feedback on design concept; Get feedback on usability issues of web interface; Get feedback on improvements for web interface	3
Simulate	Simulate user interviews; Simulate cognitive walkthrough of web interface *Instructor:* Simulate user interviews	3
Evaluate	Evaluate generated persona, avatar and knowledge graph; Evaluate generated persona	2
Brainstorm	Brainstorm many possible design options; Brainwriting project proposal	2
Design	Iterate on design concept viability	1
Interpret	Explore market and user research	1
Refine	Refine generated personas	1
Reflect	Reflect on using GenAI in the design process	1

6.6.1 ***Generating User Personas and Simulating Interviews.*** Several courses encouraged students to use GenAI to create user personas (fictional representations of potential users of a tool being designed) [8, 40, 110, 124, 127]. Many of these involved simulating user interviews through chat-based LLM tools [8, 40, 124, 127]. One study highlights the ability of LLMs to facilitate more data-driven persona creation with a course activity in which students use LLMs to generate user personas using survey data [127]. While students generally found it helpful for generating user personas, many noted that it seemed to offer repetitive and sometimes stereotypical personas [8, 40, 124, 127], echoing issues of "regression to the mean" heard across various fields' comments on GenAI usage.

6.6.2 ***Creating Designs and Prototypes.*** GenAI supported activities across a variety of phases of the design process, including brainstorming, storyboarding, and creating prototypes [37, 106, 110, 124]. Students found it helpful, noting that it extended their creative abilities and improved design outcomes in general [110], though sometimes noted challenges with accuracy. From a teaching perspective, GenAI was effectively employed to automate the development of teaching materials, such as app mockups, which significantly enhanced instructional efficiency and allowed instructors to focus more on content planning and guidance [47]. Instructors interested in adopting GenAI for HCI stress the importance of emphasizing ethical concerns to students as they use GenAI for design, and the need for dialog with students along the way in order to navigate the balance between freedom and guidance in their use of GenAI [37, 116].

6.6.3 ***Feedback on Designs.*** Across various educational settings, GenAI tools were integrated into multiple stages of the design process and learning activities, supporting ideation, interface evaluation, visual design, and the creation of instructional materials. In fictional field studies and cognitive walkthroughs of a tool, GenAI provided helpful feedback, noting potential usability issues, and introduced new perspectives [10, 106]; however, its responses were often overly positive and lacked critical depth, limiting its effectiveness in challenging student assumptions or simulating realistic user interactions [106]. During interface evaluations, students appreciated the tool's usability and the quality of its feedback; although it had difficulty identifying visual or aesthetic issues, such as poor layout or color contrast, that human evaluators readily noticed [10].

6.6.4 ***Student Reflections on GenAI Usage.*** In some HCI courses, students were required to reflect on their process. Reflections asked students questions about their experiences with using GenAI for the various parts of the design process, and how it helped or hindered [40, 110]. One included a more structured reflection with both Likert and open ended portions [40]:

> **Persona Construction Questions (Direct Quote from [40])**
>
> - Using GPT-3 contributed to making the persona information more consistent.
> - Using GPT-3 contributed information to the persona that helped me better understand the people it describes.
> - The persona I created with GPT-3 would further improve my ability to make decisions about the user group it describes.
> - Using GPT-3 contributed to the persona seeming more like a real person.
> - Using GPT-3 contributed to presenting the persona information in a clear manner.
> - Using GPT-3 made me feel like I understood the persona more.
> - In what ways did GPT-3 help the construction of your persona?
> - In what ways did GPT-3 hinder the construction of your persona?
> - Do you have any thoughts or concerns about the use of GPT-3 for generating personas as part of the UX design process?

As reflection on design process is a usual component of many HCI courses, some courses had reflection components that did not explicitly ask about GenAI but aimed to capture student sentiments toward a GenAI-integrated design process [37, 106].

6.6.5 ***Positive Sentiments: Helpful for improving efficiency and ideation.*** GenAI integration in various components of design education was generally well received by students, with many highlighting its support for creativity, ideation, and overall learning enhancement. In fictional field studies, students engaged actively, gained useful insights, and considered the exercise pedagogically beneficial [106]. When applied to persona creation, GenAI tools were credited with improving efficiency, accuracy, and diversity [127], and with aiding elaboration and creativity during the co-creation of usable personas using GPT-3 [40]. In usability evaluation settings, students found the GenAI-based tool intuitive and effective for identifying meaningful interface issues [10].

In broader user-centered design tasks, students reported *enhanced creativity, faster ideation, and more effective design outcomes* [124], often appreciating AI-generated suggestions they might not have developed independently [110]. In simulated interview exercises, ChatGPT was perceived as generating believable and useful responses that supported persona development, achieving high alignment with instructor-defined expectations [8]. Furthermore, in UI design instruction, GenAI-produced materials improved student satisfaction, boosted technology acceptance, and increased instructional efficiency by reducing development time and enabling instructors to focus on higher-level planning and guidance [47].

6.6.6 ***Negative Sentiments: Unoriginality, Bias, and Ethics.*** Despite the overall positive reception of GenAI tools in design education, several limitations and challenges were reported. In persona generation, students encountered UI complexity, limited design diversity [127], and outputs that were sometimes vague, inconsistent, or stereotypical [40]. In simulated user interviews, GenAI tended to provide brief, overly direct answers, lacking the openness or illogical thought patterns typical of real users, which resulted in students having inauthentic interaction experiences [106]. During AI-supported cognitive walk-throughs, students expressed initial uncertainty, low confidence in addressing AI-generated errors, and frustration with hallucinated content or excessive, misaligned suggestions [10].

> **Prompt Excerpt (Direct Quote from [40])**
>
> "You should refrain from giving the student the correct solution, as that wouldn't help them become better problem solvers. You should mentor them and walk them through their logic, explaining what is good and what needs to change. If a student shows that they are confused about some aspect of the logic, you can use light pseudocode (structured comments, not actual code) to push them in the right direction. Look at the following list to know when to give light pseudocode.
>
> - If a student is close to the right answer then pseudocode isn't needed. Push them in the right direction is all that is needed
> - If a student has many key ideas but is having a hard time putting those ideas in the code, then pseudocode would mean to give them a logic outline of the correct code
> - If a student is missing key ideas, then pseudocode could be used to highlight key ideas needed.
> - If a student is very confused, then pseudocode could be used to help them understand the overall shape of the correct code and why the code is that way"

In broader user-centered design activities, challenges emerged around controlling GenAI, verifying its accuracy, and avoiding over-reliance on its output [124]. Students also reported creative stagnation, where the AI repeatedly produced similar ideas or encountered "creative blocks", with several studies noting the *limited diversity and often monotonous nature of outputs* [8, 40, 110, 124, 127]. Students and instructors raise concerns about *ethical use and concerns with GenAI models using artists' work non-consensually*, and *model bias* [110, 116].

6.7 Algorithms and Data Structures

Eight papers report GenAI interventions within algorithms and data structures courses. The interventions in these courses generally involved students (seven of the eight papers), with one study describing teaching staff using GenAI tools. Overall, the interventions assisted with understanding concepts and algorithms, and providing feedback on student solutions. Table 9 summarizes the GenAI activities and aspects of this cluster.

Table 9: GenAI activities found within the Algorithms and Data Structures literature.

GenAI Use	Illustrative Activities	#
Interpret	Asking conceptual and technical questions about linked lists; Analyze algorithm complexity; Interact with a course chatbot that provides explanations; Clarifying concepts *Instructor:* Clarifying complex concepts about shortest path algorithms	5
Get Feedback	Get formative feedback on programming quizzes involving recursion; Get structured feedback on decomposition, pattern recognition, abstraction, and algorithm design *Instructor:* Providing feedback on student submissions	3
Refine	Debug algorithms; Optimize (code) functions; Help with debugging code	3
Generate	Generate solutions *Instructor:* Generating exercises about shortest path algorithms	2
Evaluate	Compare solutions with GenAI solutions	1

6.7.1 *Understanding Concepts and Algorithms: Explanations and Hints.*

GenAI tools, particularly ChatGPT and Google Gemini, were employed to support students in understanding data structures, algorithms, and computational thinking principles. In multiple interventions, students used LLMs to decompose problems, clarify recursion, traverse data structures, and gain insights into algorithmic efficiency [25, 51, 68]. In addition, ChatGPT was also employed to deliver structured, step-by-step guidance grounded in a multi-dimensional scaffolding approach, specifically addressing five core dimensions of the computational thinking framework—decomposition, abstraction, algorithm design, debugging, and iteration to effectively support student reasoning throughout problem-solving tasks [68]. In AI-enhanced platforms such as Gurukul, LLMs were deployed to deliver retrieval-augmented explanations and concept-driven hints that avoided revealing direct answers [97].

6.7.2 *Feedback on Student Solutions.*

Several studies incorporated LLMs to generate formative feedback on student submissions. An instructor-created prompt using GPT-4 was compared with direct instructor feedback on recursion quizzes, and the GenAI responses, structured with chain-of-thought prompts, led to higher resubmission scores and were perceived as equally supportive [87].

In some courses, teaching assistants also mediated GenAI feedback, delivering refined insights from ChatGPT-4o to improve clarity and accuracy before presenting them to students [51]. Gurukul's AI module offered scaffolded feedback through a live code editor, enabling students to refine their understanding of programming logic and algorithm design within a constraint-based learning environment [97].

6.7.3 *Student Reflections on GenAI Usage.*

In some data structures and algorithms courses, instructors required students to reflect on their usage of GenAI [75, 96]. One such reflection prompted students to write their own solution and then compare its performance and efficiency with GenAI code [96]:

> Student Reflection (Direct Quote from [96])
>
> "Use a generative AI, such as ChatGPT, Bing AI, or Bard, to optimize program, function, or code segment.
>
> - Attach the screenshot of "solution" generated by generative AI
> - Summarize your findings (at least 50 words).
>
> Here are several ways that you can use generative AI to optimize program, function, or code segment for performance:
>
> - Best practices: Having been trained on a wide range of code patterns, ChatGPT can help you follow best practices for coding, leading to more efficient and optimized code.
> - Refactoring: ChatGPT can help to reorganize existing code to improve its efficiency and maintainability without affecting its functionality.
> - Code Suggestions: ChatGPT can suggest code snippets or alternative solutions to improve the performance of your existing code. (You need to have your own implementation first before asking AI tool for alternative solution.)"

Another required students to include their transcript and also included more general questions about their GenAI use with the intention of having students show evidence of learning, and how their knowledge would transfer to other problems [75]. Instructors included such reflection components to encourage students to acquire the intended skills and knowledge, and prevent them from becoming over-reliant on the tools for core competencies [75, 96]. An example of a student reflection appears below.

Student Reflection (Direct Quote from [75])

"The learning reflection form asked students to include the entire transcript of their interaction with the AI tool, even parts they didn't use. It also included questions asking students to explain whether/how they used the AI content as part of their submission and how they checked its accuracy. Finally, they were invited to reflect on their learning with the prompt "Give some evidence that shows what you learned from using the AI tool for this assignment. For example, this could be a written description showing you can explain the content in question, some new code that applies what you learned to a different problem, a new version of the code that was changed in sufficient ways to better solve the problem, etc."

6.7.4 *Positive Sentiments: Scales instructor-led scaffolding, and improves student engagement.*

In data structures and algorithms courses, GenAI can *help scale instructional strategies by encouraging consistent use of instructor-designed prompts or problem-solving steps.* For example, instructors can provide prompts that guide students to trace through data structures or think about aspects of a particular step in an algorithm, ensuring broader adoption of intended mental models for understanding complex topics.

Across courses, GenAI-integrated interventions were successful in improving student comprehension and computational thinking by way of increased engagement through real-time responses. Importantly, these positive outcomes occurred when instructors provided very hands-on and detailed guidance for students' use of GenAI [25, 51, 68, 87, 96]. This guidance included guardrails and systems to ensure that the AI did not give away answers directly [99] and providing instructor-created prompts for students to use (both for understanding content and for generating helpful feedback on their solutions with structured rubrics) [25, 68, 87, 97, 99]. One study that specifically included the perspective of teaching assistants emphasized the importance of the teaching staff role in their hybrid human-GenAI approach, since AI-generated materials frequently required corrections in order for them to be understandable and helpful to students [51].

6.7.5 *Negative Sentiments: Vague explanations do not improve understanding, and over-reliance must be proactively avoided.*

In several studies, students mentioned the GenAI explanations were *too vague or abstract for them to understand and apply concretely* to the problem or misconception at hand, and either left students confused or needing re-explanation from instructors or teaching assistants [51, 68, 87, 97]. Another concern that arose was *over-reliance on the GenAI tools.* Several instructors handled this proactively by 1) creating clear prompts or tools for student use and 2) setting expectations for or modeling intended GenAI usage [25, 51, 68, 87, 96].

6.8 Data Science & Visualization

The result set included five papers related to data science and visualization. The observed interventions in these courses primarily involved student-centered activities. The following sections provide a detailed account of the main findings. Table 10 summarizes the GenAI activities and aspects of this cluster.

Table 10: GenAI activities found within the Data Science and Visualization literature.

GenAI Use	Illustrative Activities	#
Generate	Generate code for common algorithms; Generate visualisations *Instructor:* Generate teaching materials/exercises/datasets	7
Interpret	Clarify/revise data science concepts; Analyze datasets; Explore concepts	4
Reflect	Reflect on prompt-engineering impact; Document GenAI interactions/process *Instructor:* Evaluate utility in student/teacher support	4
Brainstorm	Recommend libraries/resources/career paths; Support design decisions; Stimulate design ideas	3
Evaluate	Evaluate raw/engineered prompts for wrangling/analysis/visualization; Evaluate performance in data analysis; Critique generated solutions	3
Refine	Debug code	1

6.8.1 *Understanding Concepts and Algorithms: Explanations and Hints.*

In multiple data science and visualization courses, ChatGPT served as a dynamic tool to support conceptual understanding of algorithms and data-centric tasks. Students relied on the model for real-time explanations of programming logic, statistical methods, and machine learning workflows, including tasks like data wrangling, feature extraction, and parameter tuning. For example, in a course designed around project-based data analysis, ChatGPT was used to scaffold student learning by helping them understand Python syntax and interpret model behaviors through prompt iterations [112]. Similarly, in a graduate-level data visualization course, students explored how ChatGPT could assist in interpreting error messages and generating ideas for visualization, enabling them to clarify unfamiliar libraries and tools independently [56]. These activities were enhanced when paired with prompt engineering instruction, which enabled learners to ask more effective questions and receive pedagogically sound responses [14, 18].

6.8.2 *Feedback on Student Solutions.*

ChatGPT was commonly employed as a formative feedback tool, especially in programming assignments where students submitted iterative solutions. The AI provided actionable suggestions, such as identifying logical errors, improving code structure, and offering alternative strategies. Instructors observed that students who used ChatGPT could more rapidly diagnose bugs and refine their submissions, reducing the turnaround time for problem resolution and increasing autonomy [18]. Moreover, in some redesigned courses, students were required to submit a "prompt engineering report" alongside their final data science project, critically reflecting on the model's feedback and its influence on their decision-making [14]. This dual-assessment structure promoted meta-cognitive awareness and encouraged learners to treat GenAI as a collaborator rather than an answer provider.

6.8.3 *Student Reflections on GenAI Usage.*

Students were asked to reflect on the usefulness of ChatGPT for a wide variety of activities related to the course and their broader career, including: programming, learning and understanding data science concepts, analyzing datasets, suggesting new learning materials, and finding career paths [129]. Of particular relevance to data science was a reflection question that asked "How much do you agree that ChatGPT can help analyze data sets with no or less human effort? (by considering whether it can automatically give correct preprocessing, or handle special issues, such as imbalance, automatically)". This draws attention to the subtleties of handling special cases in data and asking students to reflect on whether or not GenAI is capable of effectively assisting with these cases. Another course that asked students to reflect on their use of ChatGPT had post-assignment elicited feedback on how effective GenAI was in improving the quality of their assignment submissions, reducing their time spent, and enhancing their engagement and curiosity in the subject (among other questions about its usefulness) [56]. Another approach was to have students complete an assignment both with and without GenAI and then reflect on the advantages and disadvantages of its responses [14]. These approaches illustrate the importance instructors find in having students reflect on both the affordances and drawbacks of GenAI in their coursework.

6.8.4 ***Positive Sentiments: Increased Engagement, Independence, and Ability to Create Complex Code.*** Across studies, students reported highly positive experiences when using ChatGPT to support data science and visualization tasks. Key benefits included increased engagement, faster access to explanations, and a heightened sense of independence when working on complex problems [56, 112]. In particular, non-native English speakers [129], and students with less prior coding experience expressed that ChatGPT lowered the barrier to entry for understanding code syntax and documentation [14, 56]. Learners appreciated its availability and the non-judgmental nature of interactions, which allowed them to explore ideas and troubleshoot without hesitation [14]. Instructors also noted qualitative improvements in the complexity and polish of student outputs, especially when GenAI use was paired with critical reflection and responsible design.

6.8.5 ***Negative Sentiments: Difficulties with Visual Tasks and Learner Over-Reliance.*** Students sometimes received inaccurate, verbose, or contextually irrelevant responses from ChatGPT, especially when tackling open-ended visualization tasks or assignments requiring critical thinking and design justification [18, 56]. The model's inability to engage in nuanced visual reasoning or provide consistent support for tasks like aesthetic optimization or storytelling limited its effectiveness in higher-order design challenges. In some cases, learners became overly dependent on the model, leading to a diminished sense of agency or shallow learning if not accompanied by appropriate scaffolding [14]. These observations underline the importance of integrating GenAI literacy into the curriculum, particularly in teaching prompt engineering and critical evaluation of AI-generated content.

6.9 Object Oriented Programming

The result set contained five papers related to Object-Oriented Programming (OOP) courses. Three out of five papers described GenAI in OOP courses [4, 114, 115]. The other two papers primarily focus on exploring whether GenAI can solve OOP exercises rather than learning-focused applications in OOP courses [19, 20]. While this second set of papers are marginally relevant, they were included in this report due to the relatively small number of OOP papers found. Table 11 summarizes the GenAI activities and aspects of this cluster.

Table 11: GenAI activities in this set, grouped by base code with illustrative activities.

GenAI Use	Illustrative Activities	#
Generate	*Instructor:* Test generation to support grading; Creating student exercises	2
Evaluate	Evaluating generated feedback on project reports *Instructor:* Solving existing tasks	2
Interpret	Making sense of course concepts and finding resources	1
Get Feedback	*Instructor:* Grading student exercises	1

Several key applications of GenAI were identified in the context of OOP courses: a) using GenAI to evaluate existing exercises [19, 20], b) using GenAI to generate new exercises [115], c) providing feedback to students in the form of question-answering [114], and d) providing feedback to students on their assignments [4].

The following sections present a detailed discussion of these findings.

6.9.1 ***Using GenAI to Assist in Course Preparation.*** GenAI models can play a role in the course preparation process by both validating existing exercises [19, 20] and generating new ones [115]. In this context, validation of existing exercises involves assessing what GenAI models are capable of solving and helping instructors refine their materials to prevent students from relying solely on these models without proper understanding [19].

Validation of Existing Exercises. To evaluate GenAI models, the authors of both papers used the same set of six OOP exercises from their courses [19, 20]. Overall, the models tested (GPT-3.5, GPT-4, and Bard) were able to solve these programming exercises. However, the generated code often failed to pass all test cases and exhibited code quality issues. An exception was noted with the GPT-4 model, which successfully passed all test cases for one of the exercises [20].

GPT-3.5 and Bard performed worse than the GPT-4 model, often producing syntax-related issues such as minor compilation errors or missing imports. Additionally, all three models struggled with OOP concepts, such as failing to declare a class as abstract or incorrectly identifying and implementing business rules [19]. In the context of education, GenAI models can be utilized to validate assignments by determining if the assignments can be easily solved by the models [19]. Both studies also emphasized that simply checking the correctness of the code is not sufficient. With the growing prevalence of AI, extra effort is needed to ensure the generated code meets high-quality standards [19, 20].

Generating New Exercises. The ChatGPT system, based on the GPT-3 model, was used to evaluate its ability to generate OOP programming exercises [115]. It is important to note the authors focused solely on generating task descriptions and did not generate any code for them. Overall, LLMs performed well in this task, even for scenarios requiring the use of custom APIs in the exercises. However, the more complex the exercise, the more prompts were needed to achieve a good result [115]. This approach can significantly save instructor time and improve efficiency in creating programming exercises.

From the student perspective, the generated exercises were generally well-received, with most students finding them appropriate in terms of structure, difficulty level, and clarity [115]. This aligns with findings from the instructor's perspective, supporting the idea that LLMs are suitable for generating coding exercises, at least their descriptions (without code placeholders or test cases).

6.9.2 ***Using GenAI to Provide Feedback.*** Two papers report GenAI interventions within OOP courses [4, 114].

Question Answering. ChatGPT and a RAG-based system were compared to evaluate their ability to provide answers to student questions [114]. The goal of this feedback is to help students clarify their doubts about OOP theory. Overall, the findings showed that GenAI models perform well for this task, delivering precise and detailed answers [114]. However, employing advanced techniques such as RAG can significantly improve the clarity and quality of responses by incorporating additional information, such as links to resources, more context about course exercises, or tailored information presented in a more student-oriented way [114].

General Feedback. Another study [4] focused on providing general feedback on student assignments by comparing GenAI models with human-generated feedback. ChatGPT's textual feedback for all exercises was more detailed, in-depth, and direct compared to feedback provided by human experts. However, auto-generated feedback should still be reviewed and tailored by experts or instructors to ensure accuracy, build student trust, and confirm the feedback is reliable.

From the student perspective, auto-generated feedback received significantly higher ratings compared to feedback from experts; suggesting that GenAI models are already capable of providing rich and valuable feedback for OOP exercises.

6.9.3 ***Positive Sentiments: Improves Instructor Efficiency and Provides Helpful Feedback.*** GenAI was found to be helpful for generating complex exercises, though it is critical to maintain some ambiguity in problem descriptions such that students cannot simply copy them into a GenAI prompt [115]. For students, GenAI is helpful in providing understandable feedback and help, especially when RAG or other techniques ensure the output is from an instructor-curated resource [4, 114].

6.9.4 ***Negative Sentiments: Fails Requirements and Produces Code that is not Learning-Oriented.*** AI-generated code was often found to have incorrect or missing imports and failed to meet requirements, highlighting the need for instructors to double-check any GenAI-generated instructional materials provided to students [19]. It also was prone to creating code that would not necessarily help students learn, and was harder to work with; such shortcomings then required substantial revisions from instructors [115].

6.10 Security

Four papers report GenAI usage within computer security courses. From those, three were teacher-oriented, while one was student-oriented. The existing works explore a) how well LLMs can handle security exercises, b) what changes can be done to incorporate LLMs in security education, and c) how well can an LLM-based tool support students in doing security-related tasks. Table 12 summarizes the GenAI activities and aspects of this cluster.

Table 12: GenAI activities found in Security literature.

GenAI Use	Illustrative Activities	#
Evaluate	*Instructor:* Evaluate security issues introduced when using GenAI; Assist with security tasks; Detection of XSS attacks; Use of GenAI in social engineering; Detection of phishing attacks; Recommend penetration test cases; Recommend vulnerability fixes	8
Interpret	Help to understand/apply/analyze concepts (e.g., file permissions, memory leaks, reverse engineering); Personalized tutoring that analyzes terminal and file activity during challenge-based exercises *Instructor:* Identify security vulnerabilities	3

6.10.1 ***Evaluation of LLM's capacity to handle security exercises.*** Two of the found works attempted to measure how well ChatGPT can handle security-related tasks and questions [52, 67]. Both of these studies report that ChatGPT is capable of supplying students with information associated with computer security tasks. More specifically, although it has difficulty with vulnerabilities that are spread across multiple files [67], it can identify some vulnerabilities and recommend penetration testing cases [67], provide information about vulnerability identification and correction [67], as well as automate some parts of the job [52].

Furthermore, Jamieson et al. [52] reported the need to train students in prompt writing and in validating GenAI's output. Finally, Li et al. [67] noted that, if nothing changes in terms of grading, students might leverage ChatGPT to pass the course.

6.10.2 ***Propose new course design.*** Wang et al. [123] proposed a new design for a course focusing on security which includes ChatGPT related activities, such as a) using ChatGPT to assist in security tasks, b) identification of security issues caused by ChatGPT, c) automating web programming tasks, d) detecting attacks in code (e.g., cross-site scripting (XSS), e) using ChatGPT for social engineering, and f) employing ChatGPT to detect phishing attacks. However, it should be noted that this new course design was still conceptual and untested.

6.10.3 ***Student-support tools.*** Nelson et al. [83] implemented an LLM-based tool – called Sensai – that supports students execution of security tasks. This tool tries to give hints and asking probing questions that try to make students think instead of giving them direct answers. The tool works by transparently extracting and utilizing the learner's working context (e.g., contents of the code editor and terminal). The tool was used in a real educational scenario, having also been evaluated by the students. The results of the evaluation show that, in general, students do not prefer Sensai over more traditional support structures, such as the professor, the teaching assistants and external online resources. However, most students indicated that it is 'easier' to seek guidance from this tool than from those more traditional resources.

6.10.4 ***Positive Sentiments.*** One positive sentiment was the idea that ChatGPT can be used to accelerate learning of computer security learning objectives [52]. Another positive aspect is the possibility of raising the evaluation threshold [67].

6.10.5 ***Negative Sentiments.*** ChatGPT's capacity to solve many of the exercises was noted as a potential cause of student demotivation by [52].

6.11 Machine Learning

The result set contained three studies related to machine learning courses, highlighting its use in supporting student learning across coding, analysis, and assessment tasks. Table 13 summarizes the GenAI activities and aspects of this cluster.

Table 13: GenAI activities found in ML literature.

GenAI Use	Illustrative Activities	#
Evaluate	Critically evaluate the results of generated analysis; Evaluate quality of questions; Validate generated visual artifacts *Instructor:* Answer existing exam questions	4
Generate	Generate QR codes as a visual artifact *Instructor:* Generate new exam questions	2
Refine	Tune and refine prompt parameters; Tune the prompts to improve results	2
Interpret	Analyze and code relevant literature in a survey	1

6.11.1 ***Understanding Concepts and Algorithms: Explanations and Hints.*** In research-oriented and project-based learning environments, students used ChatGPT to explore complex topics such as model selection, overfitting, and hyperparameter tuning by posing natural language queries and receiving contextualized explanations [64]. This interaction supported their comprehension of machine learning pipelines and helped demystify common coding errors and analytical steps. In another intervention, the integration of ControlNet and Stable Diffusion allowed students to manipulate input parameters and visually observe the corresponding output transformations, thereby deepening their understanding of how algorithmic decisions influence model behavior [58]. Additionally, in assessment-focused scenarios, LLMs such as ChatGPT-3 and Codex provided structured, chain-of-thought responses that guided students through the logic of exam-style problems in regression, neural networks, and decision trees [31].

6.11.2 ***Feedback on Student Solutions.*** GenAI tools were widely used to provide feedback and iterative support for student-generated outputs in machine learning education. On the one hand, students leveraged ChatGPT to assist with technical problem-solving, code review, and refinement of research questions during a course-based undergraduate research experience [64]. On the other hand, ChatGPT helped students revise literature summaries and debug code, supporting comprehension and workflow efficiency [64]. Additionally, LLMs such as ChatGPT-3 and Codex were used to automatically answer and generate machine learning exam questions, where their performance was benchmarked against human-written content. The tools demonstrated comparable effectiveness in providing explanations and generating assessment-relevant content, thereby simulating instructor-level feedback in scalable settings [31].

6.11.3 ***Positive Sentiments:*** In general students across all interventions expressed strong satisfaction with GenAI's ability to enhance their learning experiences. Participants noted improved productivity, increased engagement, and clearer conceptual understanding when using GenAI for coding tasks, literature review, or interpreting machine learning pipelines [64]. Visual feedback mechanisms, such as those developed with ControlNet, were praised for fostering creativity and deepening learners' intuitive grasp of model behavior [58]. Instructors observed that LLMs provided scalable support for both formative assessment and exploratory learning, confirming their value in complex machine learning contexts [31].

6.11.4 ***Negative Sentiments:*** Students working with real-time visual generation tools expressed concerns about the system's complexity, the potential for misuse (e.g., plagiarism), and the steep learning curve associated with new ML-integrated GenAI tools [58]. In the research-focused setting, a few students became overly reliant on ChatGPT, which risked diminishing their critical thinking and originality [64]. Moreover, while LLMs demonstrated high performance on assessment tasks, they occasionally produced subtly incorrect or incomplete answers, emphasizing the need for human oversight and critical evaluation by students [31]. These findings suggest the importance of implementing GenAI literacy frameworks and maintaining a balanced pedagogical approach that reinforces student reflection and accountability.

6.12 Networks

Two papers from our result set focused on computer networks. Table 14 summarizes the GenAI activities and aspects of this cluster.

Table 14: GenAI activities found in Network literature.

GenAI Use	Illustrative Activities	#
Interpret	Explore the transition from IPv4 to IPv6 in a collaborative group setting; Use for asking general questions and exploring concepts; Apply Text2Net to explore real-world network scenarios; Use Text2Net to configure and practice with complex network setups	4
Evaluate	Critically evaluate outputs (for accuracy, relevance, quality)	1
Generate	*Instructor:*Design and development of assessments and rubrics	1

6.12.1 ***Exploring complex network scenarios and protocol transitions***. The first study examined the integration of GenAI within the assessment to create an authentic assessment for students in a network engineering course [17]. Students used GenAI in their assessments to explore the transition of a network from IPv4 to IPv6 in a collaborative group assessment setting. Students were required to use GenAI for both inquiry and critical evaluation of the output, with marks allocated to this in the rubric. The second study focused on the application of GenAI in designing a tool, Text2Net, to facilitate the setup of complex network scenarios more efficiently and to provide practical insights into real-world scenarios [76].

6.12.2 ***Positive Sentiments***. Both studies have positive perceptions from the participants. In the first study, students expressed appreciation for the assessment's structure and its alignment with course objectives; however, they noted the need for more comprehensive guidance on effectively utilizing GenAI tools [17]. In the second study, both instructors and students, gave high ratings regarding the ease of use, transparency, and educational value of the Text2Net tool [76].

6.12.3 ***Negative Sentiments:*** In the first study, students expressed concern over the limited guidance provided for using GenAI tools, which risked fostering over-reliance on AI and disadvantaging those with lower AI literacy [17]. In the second study, participants noted that Text2Net's functionality remains constrained—restricted to static routing and dependent on non-scalable RegEx and pattern-matching techniques—and that occasional inaccuracies in LLM-generated outputs required additional user clarification [76].

6.13 Theoretical Computer Science

Two papers from our result set focused on the theory cluster: one relating to a Theory of Computation (ToC) course and another one course in applied logic and formal methods. Table 15 summarizes the GenAI activities and aspects of this cluster.

Table 15: GenAI activities found in TCS literature.

GenAI Use	Illustrative Activities	#
Generate	Write a formal specification; Generate a program using specification	2
Evaluate	*Instructor:* Evaluate the performance of ChatGPT-4 on answering questions covering basic topics in ToC	1
Interpret	Ask to explain assignment-related concepts	1

6.13.1 ***GenAI Capabilities in the Context of Theoretical Computer Science***. The first paper was instructor-focused [41]. It offered insights from experiments evaluating the performance of ChatGPT-4 on 450 questions on basic topics in ToC. One of the aims was to inform instructors on how they can limit ChatGPT's impact on students' learning of relevant ToC concepts, utilizing the identified weaknesses of the tool. The latter includes issues of ChatGPT-4 interpreting state diagrams drawn in Word using its built-in shapes, the tool's incremental approach that does not explore faults in its initial reasoning, ChatGPT's struggle with "universality" questions that consider every possible case, and practical regex-related questions. These insights are important for instructors in theory-related subjects as it shows that ChatGPT-4 is largely unsuccessful at solving theory problems. However, in the light of the speed of LLM's development, further work needs to be done to validate that these weaknesses are still relevant.

6.13.2 ***Experiences of Offering a Monitored GenAI Environment***. The second paper was a student-focused paper, which revolved around the authors' experiences creating a monitored environment in which students could use AskGPT, a Visual Studio Code plugin based on OpenAI's `text-davinci-003` LLM without restriction [93]. This approach could theoretically also be applied in contexts outside of theory, however, it is worth noting that the authors' initial expectations as to scale and pattern of GenAI usage were not fulfilled – students used the tool much less than anticipated. Some reasons behind this included students' concerns of being monitored, concerns about GenAI interfering with their learning, lack of conversational context within the plugin, awkward UI, and others. This suggests that offering an environment with limited capabilities and a direct monitoring component may dissuade students from engaging with it. Moreover, it does not resolve the issue of students potentially relying on external solutions like ChatGPT. The authors also found that some students felt like they would benefit from learning about capabilities of the LLM tool in a structured way as they were not necessarily prepared to work with LLMs.

6.13.3 ***Positive Sentiments:*** No positive sentiments were found with regard to GenAI's incorporation with ToC topics and educating students.

6.13.4 ***Negative Sentiments:*** Theory of Computation (ToC) is a difficult topic for students (and genAI) because of its technical depth and the required precision. From these two papers it was evident that GenAI was not as capable as desired for ToC education. The first paper found that GenAI largely struggled with solving proof-type problems (the most common struggle with students taking ToC), which implies a weakness for pedagogical uses of GenAI for teaching proofs. While the first paper did not include a student assessment; the second paper did, and students largely did not use the GenAI tool because of how little it assisted them.

6.14 Miscellaneous

Three additional papers fell outside the major clusters but are included here due to their noteworthy integrations of GenAI.

6.14.1 ***Distributed Systems:*** In an advanced distributed systems unit, students extensively used LLMs for tasks such as code generation and debugging, with 98% relying on them for assignments [6]. While the positive sentiment was that the LLMs were generally seen as beneficial for productivity and understanding, there was a concerning trend of over-reliance, as many students submitted whole assignment prompts for complete solutions. This pattern highlights the need to update undergraduate curricula to include training on effective prompting strategies and address the ethical concerns surrounding academic integrity. The study recommends that future research should evaluate the long-term impacts of LLM usage on learning and knowledge retention.

6.14.2 ***Operating Systems:*** Zhang et al. [128] presents the setup of SortingHat, an innovative digital teaching assistant designed to enhance Operating Systems education by addressing the unique needs of each student. It uses a large language model enhanced with retrieval augmented generation (RAG) to provide personalized guidance and adaptive exercises based on individual learning histories. The system ensures fair assessment through an evaluation pipeline that gives consistent grading and constructive feedback. Additionally, extensive testing of the framework demonstrated its reliability, producing consistent evaluation scores while accurately reflecting differences in the quality of student work. This is a work-in-progress; however, the comprehensive setup of this digital assistant has the potential to be extended to other advanced computing courses, as well.

6.14.3 ***Web Development Course:*** MacNeil et al. [74] discuss an experiment where different types of code explanations were generated using LLMs and incorporated into an interactive e-book for web software development. The authors created three types of explanations: line-by-line, important concepts, and high-level summaries, allowing students to engage with them alongside code snippets. Findings revealed that students found these explanations generally helpful, with engagement varying based on the complexity of the code and the type of explanation. Students preferred summary explanations over line-by-line explanations, despite requesting more of the latter. The authors aim to explore how to enhance explanations through length, clarity, and completeness.

7 Instructor Survey Method

To understand the challenges faced by computer science faculty when incorporating GenAI into their courses, a survey-based study to elicit feedback on GenAI usage practices, concerns, preferences, and expectations was developed. Figure 3 provides a high-level schematic for the instructor survey methodology.

7.1 Instructor Survey Goals

The main objective of the survey was to complement the literature review with opinions and experiences of educators. Additionally, the responses are the evidence needed to formulate a set of reasons why instructors do not use GenAI yet.

7.2 Survey Design and Testing

This survey was intended to capture educational practices and other GenAI-related information not yet available in the literature, due to the fast evolution of the topic, and help complete the picture of GenAI use in post-introductory courses.

An initial set of survey items aligned with the WG's goals (i.e., identify what is being done in post-introductory computing courses) was drafted, then iteratively refined based on internal group feedback. A pilot run with a sub-group of critical contacts was used to refine wording of items and to produce a completion time estimation.

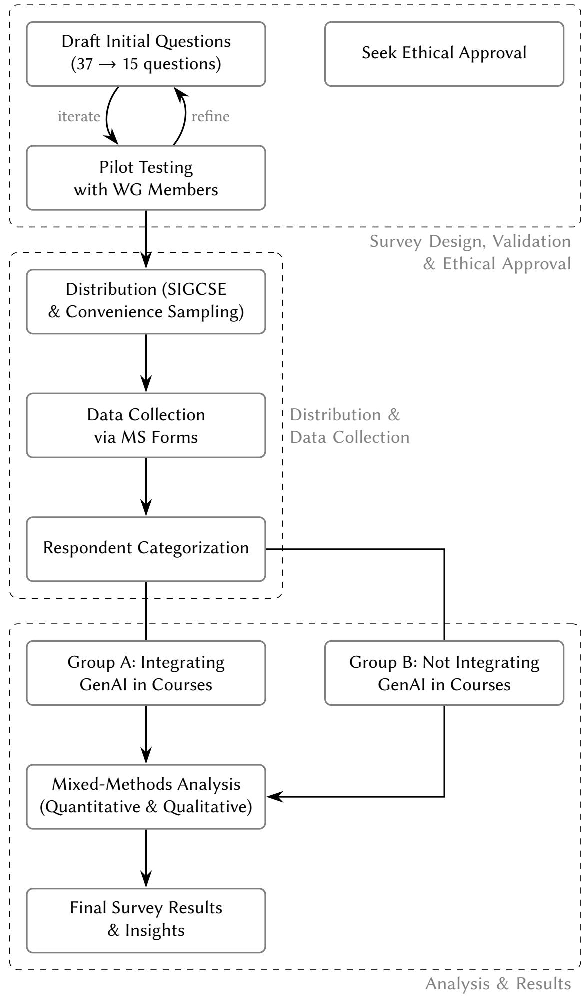

Figure 3: Workflow steps for the educator survey on GenAI integration in upper-level computing courses.

7.3 Survey Structure

The instructor survey included 15 items: six multiple-choice questions, three binary Yes/No questions, and eight open-ended questions. The survey appears in Appendix C.

7.4 Target Populations and Outline

The questionnaire targeted two groups of CS instructors:

A. Instructors who are already integrating GenAI into their teaching, and
B. Instructors who have not integrated GenAI in their courses.

7.4.1 ***Survey of GenAI-integrators:*** For group A, the questionnaire prompted respondents regarding:

- Specific activities in which GenAI was integrated into their courses (e.g., brainstorming for generating project or functionality ideas, requirements clarification, etc.)
- Examples of concrete activities where GenAI was used

- Effectiveness of those activities and the reasoning behind their effectiveness (e.g., which activities worked or didn't work well and why)
- Skills taught or referenced in their courses (e.g., general prompting skills, prompt engineering techniques)
- Revisions (accomplished or planned) to learning objectives, course objectives, or intended outcomes in response to student use of GenAI
- Expected level of transparency from students when using GenAI tools (e.g., ranging from a full log of interactions or transcript to no requirements)
- Plans to revise assessment methods in courses incorporating GenAI (e.g., awarding more points for the process or development journey, designing tasks that are harder for GenAI to solve)
- Current policy regarding student use of GenAI tools
- Availability of instructor-created resources for incorporating GenAI (e.g., sharing the URL to those resources)

7.4.2 ***Survey of GenAI-NON-integrators:*** For group B, the instructors who have not integrated GenAI, the questionnaire prompted respondents regarding:

- Reasons they have not adopted GenAI (e.g., lack of institutional support, lack of knowledge, etc.)
- Possible future plan (if any) to integrate GenAI tools into their courses

Including instructors who have not integrated GenAI into this survey intended to help shed light on the reasons behind non-adoption (e.g., institutional policies, instructor's personal preferences, etc.).

7.4.3 ***Items Common to Both:*** In addition to the above items, the questionnaire asked about the respondents' teaching background (e.g., taught courses). Appendix C presents all survey items.

7.5 Distribution Strategy

The survey was distributed using the following channels:

- SIGCSE-MEMBER mailing list: the ACM Special Interest Group on Computer Science Education includes a broad community of instructors and CSE researchers.
- Convenience sampling: people from the WG members' personal networks known to teach post-introductory courses. This approach helped to reach instructors not subscribed to research-oriented mailing lists. Since the WG team includes members from eight different countries and 15 institutions, this method leverages a geographically and institutionally diverse set of contacts, increasing the likelihood of capturing a broader range of perspectives and practices.

7.6 Ethics Approval and Data Handling

Ethics approval for this research was granted by the Science and Engineering Ethics Committee of Swansea University on July 1, 2025 (Ref. no. 1 2025 13825 13865). Amendments to the original application were approved on August 26, 2025.

The survey data was collected using Microsoft Forms. Participation in the survey was anonymous and no personal information was collected. The raw data was handled by Swansea University – the University that issued the ethics approval. It was pre-analyzed and de-identified as needed before being made available to other researchers in the WG. For example, on three occasions respondents shared links to their personal website or GitHub repository. Those links were decoupled from the rest of their responses and later reviewed separately to explore their relevance to the WG's aims. The analysis was then carried out by a sub-group WG members, from Swansea University (UK), Lusófona University (Portugal), KTH Royal Institute of Technology (Sweden), Northumbria University (UK), JetBrains (Serbia), Interdisciplinary Transformation University (Austria), and Worcester Polytechnic Institute (USA).

7.7 Analysis Approach

The survey responses were analyzed using a mixed-methods approach. For closed-ended questions, descriptive statistical analysis (e.g., frequencies and percentages) were used to identify general trends. To include responses from the "Other" option with free-text input, the responses were clustered into categories to capture recurring themes or insights not covered by predefined choices.

Open-ended questions were thematically coded to identify common perspectives, strategies, and concerns. During the coding process for each open-ended question, at least two WG members independently proposed potential themes. These themes were reviewed by another WG member. Themes were finalized upon consensus. Any disagreements were resolved by one of the group leaders.

8 Results from the Instructor Survey

The results of the instructor survey are presented in four parts:

- Overview of respondents and the subjects they teach
- Examination of the reasons reported by those who have *not integrated* GenAI and their future plans
- Summary of the responses from respondents *who have integrated* GenAI activities and analysis of their sentiments on that experience
- Summary of the skill development related to GenAI, changes to assessment practice, changes to learning objectives, adoption of AI usage policies, and open resources on GenAI, as reported by respondents

8.1 Overview of Respondents

Overall, 118 valid responses were collected. The responses were separated into groups based on their response:

- Group **IN** are respondents who have **in**tegrated GenAI into their courses
- Group **NI** are respondents who have **n**ot **i**ntegrated GenAI into their courses

According to the survey results, 56 out of 118 instructors (47%) reported already integrating (IN) GenAI in their courses. On the other hand, 62 instructors (53%) reported not integrating (NI) GenAI in their, as presented in Table 16.

Figure 4 shows the post-introductory subjects taught by the surveyed respondents (n.b. respondents typically taught more than one subject as can be expected). Table 17 shows in more detail the individual counts for each subject split by integration of GenAI. In

Table 16: Overview of respondents and GenAI integration.

Category	#	%
Respondents teaching post-introductory courses	118	—
— Not integrating GenAI (NI)	62	53%
— Integrating GenAI (IN)	56	47%

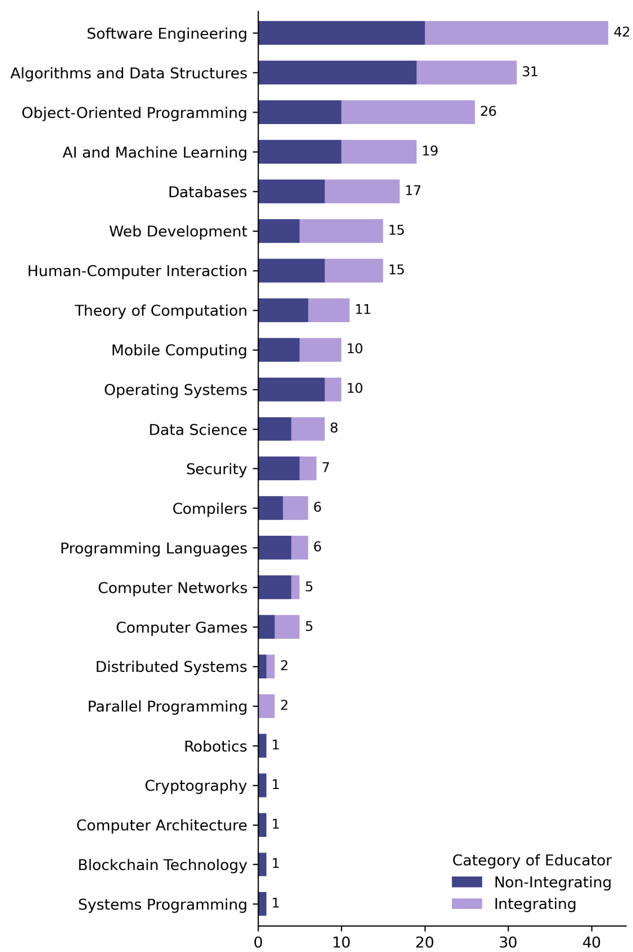

Figure 4: Count of subjects by integrating and non-integrating instructors.

terms of subjects, Software Engineering, Algorithms and Data Stuctures, Object-Oriented Programming and AI and Machine Learning accounted for 50% of the total.

Analysis of subject counts revealed a strong positive association between NI and IN categories (Pearson's $r = 0.87$, $p < .001$; Spearman's $\rho = 0.85$, $p < .001$), indicating that subjects frequently represented in one category were also frequent in the other. A Wilcoxon signed-rank test found no significant difference between NI and IN values across subjects ($W = 85$, $p = .28$), suggesting no systematic imbalance between the two categories.

Table 17: Counts of subjects split by NI (not-integrating) and IN (integrating) instructors.

Subject	NI	IN	Total	%
Software Engineering	20	22	42	18%
Algorithms and Data Structures	19	12	31	13%
Object-Oriented Programming	10	16	26	11%
AI and Machine Learning	10	9	19	8%
Databases	8	9	17	7%
Human-Computer Interaction	8	7	15	6%
Web Development	5	10	15	6%
Theory of Computation	6	5	11	5%
Operating Systems	8	2	10	4%
Mobile Computing	5	5	10	4%
Data Science	4	4	8	3%
Security	5	2	7	3%
Compilers	3	3	6	3%
Programming Languages	4	2	6	3%
Computer Games	2	3	5	2%
Computer Networks	4	1	5	2%
Distributed Systems	1	1	2	0.8%
Parallel Programming	0	2	2	0.8%
Robotics	1	0	1	0.4%
Cryptography	1	0	1	0.4%
Computer Architecture	1	0	1	0.4%
Blockchain Technology	1	0	1	0.4%
Systems Programming	1	0	1	0.4%
Total	126	114	240	100%

8.2 Instructors Not Integrating GenAI (NI)

This subsection focuses on the 62 respondents who reported not integrating GenAI in their practice. The first subsection is focused on the reasons why they have not integrated GenAI. The second subsection presents the plans these respondents are considering for possible future GenAI integration. The third subsection presents the general personas of non-integrators that may be useful when discussing attitudes towards GenAI integration.

8.2.1 ***Reasons for not using GenAI:***. Table 18 summarizes responses to why GenAI had not been integrated. These responses were inductively coded to identify common themes and their relative frequency. The coded responses clustered into eight higher-level themes. The most common was **concerns about student learning** ($n = 22$). Respondents frequently mentioned risks of eroding foundational skills, fostering over-reliance, or undermining confidence and interpersonal interaction. Some respondents emphasized the pedagogical risks of students over-relying on GenAI at the expense of foundational skills:

> *R51 — "It's unclear to me how to use generative AI tools to support the learning goals of my courses, without enabling or encouraging students to let GenAI tools take over the work, both because that would rob students of problem-solving experience and because GenAI confidently generates plausible but incorrect solutions."*

Others highlighted the long-term professional risks, warning about diminished competence in the workforce:

Table 18: Instructor reasons for not integrating GenAI.

Theme	Indicative Codes	#	Distribution
Concerns about student learning	Concern students not building foundations; concern about negative impact on students; concerns about impact on learning; concerns about correctness of GenAI output	22	
Knowledge and guidance gaps	Lack of knowledge/ideas on effective use; lack of knowledge or mastery; lack of clear guidelines; lack of institutional policy/support	34	
Integration attitudes and plans	Future integration not discounted; future integration considered/planned; developing integration approach; focus on long-term impact; waiting to see	17	
Pedagogical value and fit	Unclear value/usefulness; not useful/relevant; bad subject fit; no pedagogical benefit	12	
Resource and effort constraints	Time investment needed; lack of financial resources; institutional barriers; assessments need more thought	10	
Permitted or limited usage	Usage allowed for specific purposes (debugging, explaining concepts, polishing writing); allowing basic use with instructor explanation	8	
Negative experiences and resistance	Negative sentiment towards GenAI; negative experience with students' overreliance; won't integrate unless mandatory	5	

> *R69 — "Using GenAI in the education process will significantly diminish what students learn, thereby reducing the actual knowledge and skills they will bring into their future careers. Ultimately everyone will suffer if we produce a generation of crappy computer science students!"*

Still others voiced the concern in direct terms (which made us laugh), underscoring a belief that GenAI runs counter to the very mission of education:

> *R77 — "It makes you stupid, but the role of education is to make people less stupid. If we were doing skills training, that's one thing, but this is education."*

The next most frequently occurring theme was **knowledge and guidance gaps** ($n = 17$). Respondents pointed to both institutional uncertainties and their own lack of preparedness as reasons for postponing integration. Some emphasized the challenge with unclear or absent policies:

> *R83 — "The current institutional policy is a bit unclear on what's allowed and what not."*

> *R24 — "Lack of clear guidelines on what aspects of GenAI are permitted for students to use in their assignments."*

Others highlighted their own limited expertise, noting the challenge of learning to use GenAI tools effectively while simultaneously considering how to teach with them:

> *R27 — "Hard to work out how to do it effectively while we are still learning to use the tools ourselves. Now that there is some maturity in the tools we can begin to think about how to teach students to use them."*

> *R109 — "Lack of knowledge of how to effectively integrate. I know students are using it on their end though."*

Finally, some framed the issue as one of instability, stressing the difficulty of building sustainable practices around tools that are evolving too rapidly:

> *R70 — "The GenAI landscape is rapidly shifting, so I still feel unfamiliar with the tools to rely on them in a course setting."*

A similar amount of weight was attributed to **integration attitudes and plans** ($n = 17$). Unlike respondents who had firmly decided for or against GenAI, these comments reflected a softer position; open to exploring integration but cautious about how best to proceed. Some framed their stance as supportive in principle, but emphasized the need for careful alignment with learning outcomes:

> *R15 — "I am not against the idea, but it requires a good amount of time to think through how it can be of benefit and fit the learning outcomes."*

Others described being in the process of developing workable approaches, balancing existing policies with the challenge of designing new activities that integrate GenAI meaningfully:

> *R19 — "I'm still working on developing a good approach for integrating it into the course. While students are allowed to use them in some specifically-outlined ways (consistent with historical policy), designing new work that integrates GenAI well takes time."*

Still others pointed to the need for additional research and inspiration before committing to full integration:

> *R87 — "I do have some ideas in how GenAI would be useful in HCI dommain however, I would like to do more research into how GenAI could be used within the curriculum and some model activities so I could adapt those or take inspiration to build some activities."*

Less frequent, though still notable, were discussions of **pedagogical value and fit** ($n = 12$), with some instructors questioning the relevance of GenAI to their subjects or doubting its usefulness.

> *R22 — "I do not see the relevance in integrating GenAI into an Advanced Algorithms & Discrete Systems course."*

> *R98 — "I have not found an area where it would be useful in security."*

R114 — "The courses I teach cover logic, math and fundamentals. Students need to build this skill set themselves. I've not seen a Gen AI that does well with logic."

Resource and effort constraints ($n = 10$) highlighted the practical barriers preventing more active integration, including time pressures, financial costs, and institutional limitations. Some respondents described the challenge in terms of workload, noting that even if policies or intentions were in place, they lacked the capacity to design new activities:

R66 — "I have guidelines for students on how they are allowed to use GenAI tools, but I have not yet developed exercises that require them to use these tools. Reason: haven't had the time."

Others emphasized the financial costs involved, as well as concerns about fairness and equal access for students:

R36 — "Lack of time and money to experiment with use of GenerativeAI tools. To use them you need to integrate a paid account with tools like Cursor [...] I don't have the time to investigate at the moment. I don't have the resources. Also if I wanted to integrate use of GenerativeAI the same conditions would apply to the students."

A smaller number highlighted **permitted or limited usage** ($n = 8$), often restricted to narrowly defined contexts such as debugging or polishing writing. Some respondents described allowing students to draw on GenAI for partial tasks while ensuring that core learning outcomes were preserved:

R47 — "I didn't feel the need to. The students use it to generate code for the team works, but then again I am not that concerned with this, as aforementioned project is elaborate and not fully solved using GenAI."

Others took a more pragmatic stance, permitting use so long as it was transparent and framed as one resource among many:

R51 — "I do allow students to use GenAI tools to help with homework, just as they do any other source—cite your sources and write in your own words—because any more restrictive policy would be unenforceable."

Finally, a minority expressed strong **negative sentiment or resistance** ($n = 5$). In this particular case, there is a sense of impending pressure to integrate against their instincts (we truly wish them the best!):

R103 — "i hate generative ai. i think the benefits are not worth the costs. i am only going to "adopt" it if i am at risk of being fired if i don't."

8.2.2 ***Future plans for integration:*** Figure 5 shows the distribution of responses regarding future plans for integrating GenAI. The most frequently selected options were allowing responsible use ($n = 33$) and encouraging critical evaluation of outputs ($n = 30$) (note: respondents could select multiple options). Other common plans included using GenAI in course projects ($n = 21$), for brainstorming and ideation ($n = 20$), and for clarifying requirements ($n = 10$). Respondents also reported intentions to demonstrate tools in class ($n = 17$) and to involve students in analyzing GenAI as a source of ethical questions and systematic bias ($n = 12$). Several respondents chose Other ($n = 9$) and suggested diverse uses, such as:

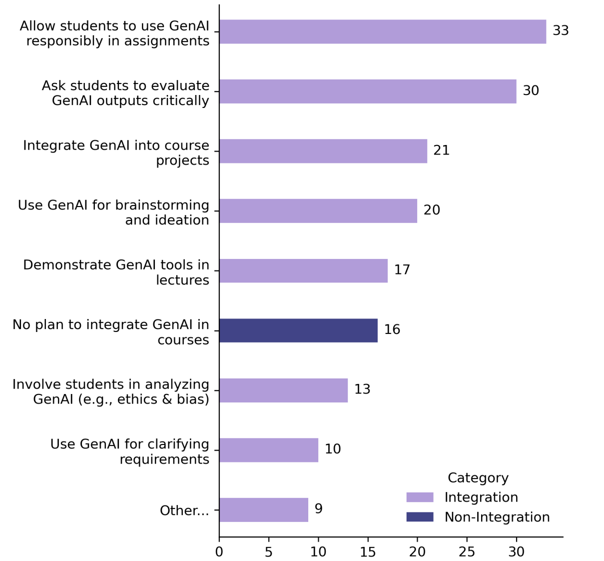

Figure 5: Future plans for integration of GenAI

R36 — "I would like to teach a course on vibe coding."

R96 — "Maybe create a workshop to do qualitative data analysis with LLMs."

R103 — "if students use it sensibly and declare as such I won't penalise it, but I'm always going to discourage it."

Finally, 16 respondents indicated they had no plans to integrate GenAI into their courses. Overall, while 53% of respondents reported not currently integrating GenAI, the majority nevertheless expressed intentions to explore some form of future integration.

It is worth noting that a very strong positive correlation was found between those respondents who have integrated GenAI and those who have not (Pearson $r = 0.95$, $p < 0.001$; Spearman $r_s = 0.95$, $p < 0.001$), indicating the relative ordering of integration approaches was highly consistent between groups. The consistency of the orders can be observed by juxtaposing Figure 5 and Figure 6.

8.2.3 ***Personas of Non-integrators:*** The analysis of the respondent data led to the identification of personas which may be useful in understanding this uncertain time of transition, resistance, and new realities in academic life. The personas are: **Pedagogical Guardian**, **Aloof Observer**, **Skeptical Resistor**, **Curious Overloaded**, and **Tentative Experimenter**. While this set of personas certainly do not capture all positions on GenAI, they represent the majority of *GenAI-skeptical* survey respondents. Table 19 provides the definitions of the personas.

8.3 Instructors Integrating GenAI (IN)

This subsection analyzes the 56 responses from instructors who reported integrating GenAI in their post-introductory courses. Section 8.3.1 provides an overview of the general ways in which respondents integrated GenAI. Section 8.3.2 summarizes the specific activities used in their courses. Finally, Section 8.3.3 reports the

Table 19: Personas of non-integrating instructors.

Persona	Description
Pedagogical Guardian	Concerned about learning, fundamentals, inter-personal collaboration, assessment validity, fairness, and outcome alignment
Aloof Observer	Doesn't see relevance for their subject/module (too theoretical, too advanced, or too basic)
Skeptical Resistor	Strongly negative; believes AI erodes education or ethics
Curious Overloaded	Would like to adopt but lacks time, money, or institutional clarity
Tentative Experimenter	Dabbling or planning to adopt soon; not fully integrated yet

positive and negative sentiments the respondents experienced from these different forms of integration.

8.3.1 ***Ways of GenAI integration.*** Figure 6 shows the ways in which integrating respondents have integrated GenAI into their courses. Three of the "other" responses are instructors using GenAI to create quizzes or generating distractors. The other three were included using GenAI as a tutor or as a support tool to help students with content generation or debugging.

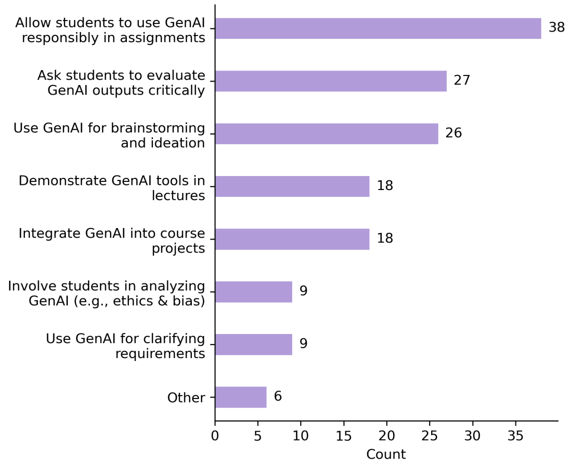

Figure 6: The ways in which instructors have reported using GenAI in practice.

The most common instructional use of GenAI was to allow students to use it responsibly in assignments ($n = 38$), followed by activities encouraging students to critically evaluate GenAI outputs ($n = 27$) and to use GenAI for brainstorming and ideation ($n = 26$). Less frequent, but still notable, were efforts to integrate GenAI into course projects ($n = 18$) and to demonstrate GenAI tools in lectures ($n = 18$). More specialized uses, such as employing GenAI to clarify requirements ($n = 9$) or involving students in analyzing GenAI with respect to ethics and bias ($n = 9$), appeared only occasionally.

8.3.2 ***Overview GenAI aspects from instructors' activities:*** A taxonomy of GenAI aspects was created in Section 6 and applied to the activities found in the research literature. This taxonomy was then applied to the activities that instructors described in Question#8. Table 20 shows the distribution of GenAI aspects found. Generate (43%) and evaluate (26%) overwhelmingly dominate the table, accounting for 69% of the activities. The remaining aspects are all below 10%.

Table 20: Percent of GenAI aspects by instructor activities.

Aspect of GenAI Use	%	Distribution
Generate	43	
Evaluate	26	
Reflect	8	
Refine	8	
Brainstorm	6	
Interpret	4	
Simulate	2	
Get Feedback	2	
Design	2	

Whilst it is difficult to make a meaningful comparison of a research study to an instructor's response to a survey question, some similarities worth reflecting upon appear. A chi-squared test of independence was used to compare the distribution of GenAI aspects across research studies and instructor activities. The overall test approached significance, $\chi^2(8, N = 214) = 14.35, p = .073$, with a moderate effect size (Cramér's $V = .27$). Standardized residuals indicated the largest contributions to the divergence were from *Interpret* (overrepresented in research studies, underrepresented in instructor activity) and *Get Feedback* (similarly more common in research than in practice). In contrast, *Generate* and *Evaluate* occurred more frequently in instructor activities than in research, though with smaller contributions. These findings suggest that while both groups prioritize generative use of GenAI, research places greater emphasis on interpretive and feedback-oriented applications, whereas instructors lean more heavily on direct generation and evaluation. One final note: it is entirely possible some respondents also authored papers included in the literature review.

8.3.3 ***Instructors' GenAI Activities by Subject:*** Figure 7 provides a per subject breakdown of the GenAI aspects of use from the instructors' activities. The data shows that *Generate* is the most common GenAI activity across domains, especially in Algorithms and Data Structures (58%) and Human–Computer Interaction (57%), with Web Development and Theoretical Computer Science also showing notable use (40% each). *Evaluate* is broadly present in all areas, peaking in Theoretical Computer Science (60%). Other activities are more domain-specific: *Interpret* and *Simulate* appear only in HCI (14% each), *Refine* is found in Software Engineering, Web Development, and Computer Games (20–22%), and *Reflect* appears only in Algorithms and Data Structures (8%) and Web Development (20%). *Brainstorming* is exclusive to Computer Games (60%), while get feedback is rare, limited to Software Engineering (11%). No activity was coded as *Design*.

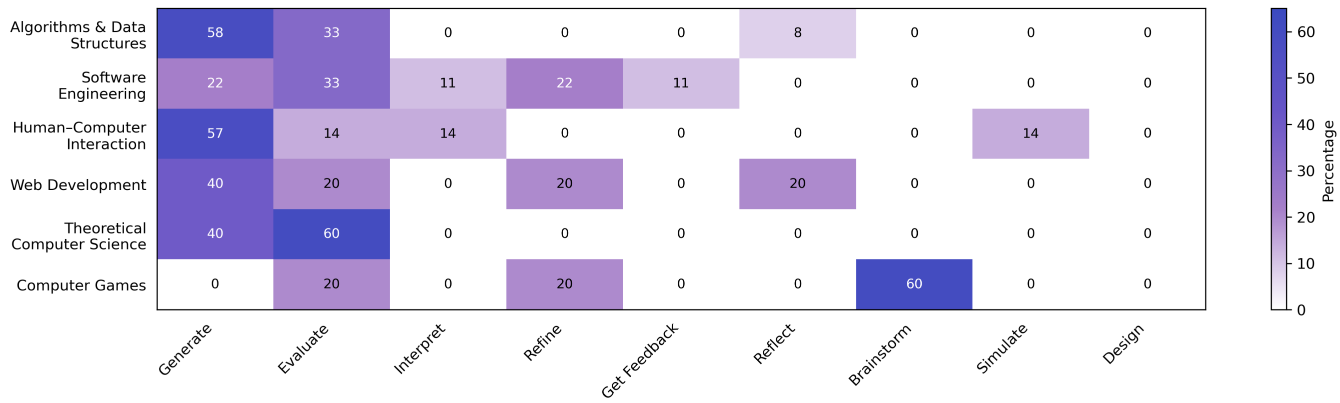

Figure 7: Heatmap of GenAI Aspects from instructor activities across different subjects, normalized by percentage.

Overall, the distribution suggests Algorithms and Data Structures and HCI emphasize generative and interpretive use, Theoretical Computer Science stresses evaluation, Web Development balances several activities, and Computer Games is dominated by brainstorming. In order to get a sense of the types of activities, the following sections give a per-subject review.

Table 21: Activities for Algorithms and Data Structures.

Actor	GenAI Activity
Student	Generate code in advanced data structures course
Student	Solve optimization problem; and Evaluate generated solution for optimization problems
Instructor Student	Generate code live in lecture in a peer instruction format; and Reflect upon live demo of GenAI code generation in classroom
Student	Use GenAI to create analogies of course concepts (e.g. recursion) then explain it to a coursemate
Student	Generate red-black coloring for binary trees; and Evaluate the red-black coloring for binary trees output
Student	Generate code; and Evaluate code output
Instructor Student	Generate incorrect code; and Students review and debug incorrect code

Algorithms and Data Structures: This subject has the greatest variety ($n = 12$) of responses. Table 21 summarizes the activities instructors reported experimenting with; for clarity, responses were edited into a common format that identifies the actor and states the activity in the imperative. As shown in Figure 7, most activities center on generation and evaluation, with only limited reflection. Notably, some instructors described scenarios where the instructor uses GenAI to produce incorrect code and students are then tasked with evaluating it as an artifact of GenAI. Several activities follow this call-and-response pattern of generate and evaluate, underscoring the importance of critically assessing GenAI output across different contexts.

Table 22: Activities for Software Engineering.

Actor	GenAI Activity
Student	Refactor programs to make use of design patterns; and Evaluate each stage of a refactoring process using GenAI
Student	Comprehend legacy code base
Student	Evaluate software design using GenAI
Student	Verify generated software requirements
Student	Generate testing scenarios; and Evaluate the testing scenarios; and Refine testing scenarios using GenAI
Student	Generate code for group assignment in systems engineering

Software Engineering: This subject has the second-greatest variety of responses ($n = 9$). Table 22 summarizes the activities instructors reported experimenting with. As shown in Figure 7, the distribution was more balanced than in Algorithms and Data Structures, spanning generation, evaluation, interpretation, refinement, and feedback, with evaluation the most frequent. Notable examples include compound activities, where students progress through a sequence of related tasks to achieve a larger goal (e.g., in testing), as well as cases where GenAI is used to help interpret artifacts considered especially challenging for junior students (e.g., legacy code bases).

Human-Computer Interaction: This subject had seven individual activities, as shown in Table 23. Similar to Algorithms and Data Structures, generation was the most common activity type; less common were evaluation, interpretation, and simulation, as seen in Figure 7. As in Software Engineering, several activities were compound in nature, highlighting the versatility of GenAI in supporting

Table 23: Activities for Human–Computer Interaction.

Actor	GenAI Activity
Student	Create personas; and Create prototypes; and Generate code from prototypes
Student	Generate cover letter; and Prepare for interview; and Evaluate use of GenAI
Student	Extract research design from given set of academic literature

existing practices. What stands out in HCI is the inclusion of an activity simulating an interview, indicating a more sophisticated use case of GenAI instructors are beginning to explore.

Table 24: Activities for Web Development.

Actor	GenAI Activity
Student	Optimize code and improve accessibility; and Reflect on GenAI in the context of web app development
Instructor Student	Generate web application with deliberate flaws; and Evaluate generated web application with flaws
Student	Generate model data by accessing the OpenAI API for a web service to serve band information

Web Development: This subject had five reported activities, as shown in Table 24. As illustrated in Figure 7, there was a modest emphasis on generation, with evaluation, refinement, and reflection comprising the remainder. Similar to Algorithms and Data Structures, instructors described compound activities in which GenAI was used to deliberately introduce flaws into an artifact, a web application, after which students were tasked with evaluating the artifact and identifying the flaws. This cross-subject pattern suggests a shared concern with developing students' ability to critically assess GenAI outputs. At a broader level, students were also encouraged to reflect on the role of GenAI within Web Development courses.

Table 25: Activities for Theoretical Computer Science.

Actor	GenAI Activity
Instructor Student	Generate a proof; and Evaluate a proof generated by GenAI
Instructor Student	Generate answers to quiz questions; and Evaluate correctness of generated quiz answers
Student	Evaluate a proof generated by GenAI

Theoretical Computer Science: This subject had five reported activities, as shown in Table 25. There was a strong emphasis on evaluating generated artifacts, with a smaller focus on generation, as seen in Figure 7. As might be expected in a theory-oriented subject, the generation and evaluation of proofs was a recurring theme. While the generation of learning materials has generally been considered less interesting in this report because of its generic nature, one instructor activity involving quizzes re-framed this use by having students evaluate the quality of GenAI answers to questions they themselves would eventually need to answer.

Table 26: Activities for Computer Games.

Actor	GenAI Activity
Student	Brainstorm sequels to previously developed game projects
Student	Determine the cause of Unity game scene defects
Student	Critical analysis of AI output
Instructor	Brainstorm project requirements
Student	Try out different prototype ideas for visual design

Computer Games: The final subject with a reasonable number of activities ($n = 5$) is Computer Games, as shown in Table 26. What distinguishes GenAI use in Computer Game courses from other courses is the strong emphasis on using GenAI for brainstorming, followed by evaluation and refinement, as seen in Figure 7. This represents another instance of a higher-order use of GenAI, which would otherwise require human participation. Although small in number, it is notable that no activities involved generation. Finally, there were no compound activities found.

Table 27: Activities for Remaining Topics.

Actor	GenAI Activity
Student	Free use of GenAI for database design; and Reflect on GenAI use for database design (*Databases*)
Instructor	Generate scenarios for ERD and schemas (*Databases*)
Student	Generate code for OOP assignment; and Generate test code for OOP assignment (*Object-oriented Programming*)
Student	Generate code; and Reflect on generated code (*Functional Programming*)
Student	Generate practice questions for data science practice (*Data Science*)
Student	Generate boilerplate code for communication system (*Distributed Systems*)
Student	Modify GenAI parameters and evaluate responses (*Machine Learning*)

Remaining Topics: Table 27 rounds up the activities instructors shared from courses such as Databases, OOP, Functional Programming, Data Science, Distributed Systems and Machine Learning. As these subjects where underrepresented, they were excluded from Figure 7. Most are based on generation activities for subject specific artifacts.

8.3.4 ***Instructors' experience with integrating GenAI tools***. To understand instructors' experiences with integrating GenAI into post-introductory courses, a thematic analysis was conducted, as described earlier, on questions 6 and 7 from Table 31, identifying a subset ($n = 26$) of respondents who shared their positive or negative experiences.

In several cases, respondents mentioned negative experiences while answering the question about positive experiences and vice versa. Additionally, relevant experiences were also mentioned in question 8 from Table 31. This also motivated the analysis of these three questions collectively in order to obtain a more comprehensive view and collect details that could have been missed had these questions been analyzed in isolation.

8.3.5 ***Positive experience with integrating GenAI tools:*** Instructors, regardless of the specific courses they are teaching, emphasize the importance of students developing solid evaluation skills to work with GenAI output. As shown in Figure 6, the second most commonly used activity is to *'ask students to evaluate GenAI outputs critically'*. These include tasks requiring students to evaluate AI-generated artifacts provided by instructors (*R41*) or tasks that require students to interact with GenAI directly (*R108, R116, R119*).

A fair share of respondents in this subset reported positive experiences when students engaged in **comparative and critical evaluation** of GenAI output ($n = 8$):

> *R88 — "Asking students to critically evaluate GenAI's replies has worked well, as well as developing skills in prompt engineering and the iterative and incremental use of GenAI rather than single prompt-result."*

> *R94 — "General review of material, drill or review questions, summarizing material, and evaluating quality of AI results [worked well]."*

> *R105 — "[...] students must ask for two different solutions for the same problem and then decide which one is the best and explaining their decision. So, I would say "Ask students to critically evaluate GenAI's replies" works reasonably well."*

Among other positive experiences were students' **enhanced abilities** ($n = 4$) when using GenAI, as well as increased **productivity and efficiency** ($n = 4$), and **improved academic outcomes** ($n = 3$):

> *R73 — "To allow/encourage students to use GenAI worked well in some respects - the failure rate on the module was lower than previous years, and the quality of work by the best perfo[r]ming students is much higher."*

GenAI tools enabled students to *"get a head start on their assignments"* (*R17*), to get clear explanations of error messages in a Database course (*R111*), to save time when debugging code in a Data Visualisation course (*R101*), or to learn how GenAI can enhance the development process in a Mobile Applications course (*R134*). One respondent pointed out the GenAI's paraphrasing ability to be particularly helpful for students with limited English proficiency.

> *R107 — "When specifying/clarifying requirements, GenAI can improve the phrasing to have clear requirements with reduced ambiguities. This usually requires several iterations, but the results are very useful, especially for non-native English speakers teaching a course in English."*

While GenAI tools have the potential to improve one's abilities and productivity, enabling them accomplish tasks they would not be able to accomplish otherwise, a respondent expressed uncertainty as to whether such help is ultimately beneficial or not:

> *R115 — "Some students reported that the LLM enabled them to complete parts of the project that would otherwise have been too difficult. I'm not sure if this is good or bad!"*

Another respondent observed that improvement in one area does not preclude the existence of issues elsewhere. This raises concern about the extent to which GenAI-supported outputs reflect the real depth of student engagement with the tasks.

> *R63 — "Requirements clarification is a little better although they have trouble thinking critically about holes in requirements."*

Several respondents reported positive experiences of using certain activities, without specifying the details ($n = 5$).

> *R33 — "I think for all of them* [critical evaluation; responsible use] *the effect was positive but limited."*

> *R54 — "Everything* [activities: critical evaluation, demonstration of GenAI tools, responsible use, generating quizzes for self check] *worked fine, so i plan to continue."*

> *R79 — "It* [brainstorming activity] *worked well and I plan to use it for other tasks as well"*

> *R99 — "Requirements clarification worked pretty well because it plays to the strengths of GenAI, while also requiring that the students do work upfront themselves."*

> *R132 — "Both* [activities: demonstration of GenAI tools, responsible use] *worked well, so planning to continue using them."*

Engagement and motivation of students was also among the positive experiences reported by the instructors ($n = 2$). One respondent observed that students valued the opportunity to use GenAI in their studies:

> *R18 — "[...] many students really liked the freedom that being able to use AI gives them - if they can use google [sic] and StackOverflow, why not AI as well?"*

An instructor in a Systems Programming course (*R42*) allowed their students to use GenAI to complete assignments, assessing them not on the code itself but on a group-based walkthrough video, covering code development, testing and execution, details about teamwork and workload distribution, and other aspects. According to this instructor, the students enjoyed making these videos and produced high-quality video content.

8.3.6 ***Negative experience with integrating GenAI tools:*** Despite several instructors observing students' ability to critically assess GenAI output, a comparable number of respondents reported that students displayed **poor evaluation abilities** ($n = 8$).

R39 — "students who used AI tools to generate the code often had a fatal design flaw in their code and didn't notice it."

R43 — "Students (second year) are not able to evaluate GenAI's replies. Most of them used the generated code without understanding what it is supposed to do. For example, when faced with the task of extracting some particular elements from an HTML webpage, they relied on generated element IDs, and they didn't match the real ones."

R48 — "Even when I have students criticize LLM outputs, they are usually insufficiently critical, and explore few ways to make the prompts more robust."

One possible explanation for students' poor evaluation abilities may be the lack of skills, which – according to one of the respondents – is rooted in 'shortcutting' and over-reliance tendencies at introductory and mid-level CS courses:

R63 — "Critical evaluation does not work well because the students, even after 2 semesters (CS1 and CS2) are unable to read or evaluate generated code. They simply do not have the skill because they blindly used GenAI in CS1 and CS2."

In this respondent's Advanced Software Design course, students were asked to refactor code by converting it to factory and strategy patterns using GenAI as a support tool. Although some students performed well, others were confused, leading the instructor to the following conclusion:

R63 — "Lesson learned: you cannot do this kind of activity unless students have learned to read and write basic programs without AI."

The sentiments expressed by *R63* regarding the lack of skills in the context of evaluating GenAI output were echoed by other respondents:

R64 — "The evaluation of replies didn't work well because students didn't have the tools to evaluate well."

R99 — ""Ask students to critically evaluate GenAI's replies" didn't work so great because students didn't have the background to know what a "good" solution looked like."

Despite having negative experience with students' critical evaluation skills, one of the respondents stated they wanted to encourage students to work on their prompt engineering techniques, and even considered potential integration of GenAI into the final-year project, albeit having concerns about the impact of such practice:

R48 — "I don't consider that best practice and is more of a "when you have a hammer, everything looks like a nail" approach."

This leads us back the issue of students having a **skill gap** ($n = 4$), which appears to go hand-in-hand with **shortcutting** ($n = 5$) – the practice of using GenAI as an easy way to get work done without considering the implications:

R52 — "They would also memorize code and just regurgitate it it back – this was obvious because they would often not apply it to newly posed problem or would not be able to make very minor syntax changes in the newly posed problem [...] But since they spent the semester not writing any code, they had a very surface level understanding of the language. Often having no idea that they did not know."

R126 — "The usage of GenAI tools made the students to jump into the solution, without thinking about the problem."

These responses suggest some students tend to use GenAI irresponsibly, even when instructors explicitly stress the importance of **responsible use** ($n = 7$).

R26 — "The responsible use is never very responsible. And proper attribution is hard for them for some reason."

R105 — "I've encouraged students to use GenAI responsibly [...] most students seem to ignore these advices and just try to blindly solve the problems with minimal effort."

One of the respondents pointed out that the mere practice of using GenAI in demonstrations may lead some students to believe it is acceptable to use GenAI output directly:

R134 — "[...] often it is students thinking that if the lecturer uses AI to show code examples, they can also use it without thinking beyond copy and pasting code."

Over-reliance on GenAI was among most common negative experiences mentioned respondents ($n = 8$):

R35 — "They [students] do sometimes copy and paste with minimal study of the content ..."

R41 — "Students just kept using gen AI as if it was an oracle."

After observing students' poor understanding and over-reliance on GenAI, the instructor has made a plan to tackle these issues by asking fundamental questions throughout the learning process:

R54 — "Students heavily used GenAI to solve the assignment. Unfortunately, there were cases when authors didn't understand solutions. So now i plan to ask fundamental questions and their own experience on 1-on-1 meetings during study."

Being concerned about potential over-reliance, an instructor in a graduate Software Engineering course proactively introduced oral examinations as a preventative measure:

R115 — "To mitigate this problem in a grad SE course, I added oral exams in which students had to walk a TA through some of the code for their project."

Poor activity design ($n = 2$) also led to negative experiences. For instance, in a Databases course, the instructor allowed students to use GenAI for clarifying requirements. However, this left some students confused about the scope of a project (*R*35). This highlights the importance of providing students with clear specifications on both GenAI usage and the scope of an activity.

Among other negative experiences highlighted by the respondents was students' **low engagement** ($n = 2$) with either activities involving direct interaction with GenAI as part of assignments or during sessions specifically designed to teach effective ways of interacting with GenAI:

R73 — "a practical session which the students didn't seem to want to en[g]age with very deeply (the session involved prompting code generation and then debugging and optimising the outputs."

Despite concerns about irresponsible use, some respondents mentioned students' **reluctance or resistance** (n = 4) to use GenAI. In some cases, the students shared explicit negative attitudes towards being required to use it. This suggests that some of them are conscientious about the potential negative impact of over-relying on these tools.

R18 — "I found that at times, students did not like being able to use AI in coursework freely. At a metacognitive level, many students reported that it made them feel as if they were learning less."

R73 — "I required students to use GenAI as part of their programming assignment, and to write a reflective essay on their experience doing so. However, some students told me they were against using GenAI in their studies. Some students also expressed concern that allowing the use of GenAI would make the coursework "too easy" (which I didn't agree with)."

R101 — "One common complaint is that the course "forces you to rely on GenAI too much" and thus students didn't feel like they left [...] with a strong grasp of programming in JS / D3."

8.4 Teaching GenAI-related skills

A total of 56 instructors provided feedback on the GenAI-related skills they teach or reference (question 9 of Table 31). Of the instructors teaching at least one such skill, the most common option, by a large margin, was prompt engineering (n = 26). Afterwards, three skills related with artifacts were reported in similar numbers: artifact generation (n = 17), artifact refinement (n = 15) and artifact evaluation (n = 14). A smaller number of instructors indicated teaching students how to use GenAI to decompose problems (n = 8). One educator selected 'other' and reported teaching critical thinking. Finally, multiple instructors reported teaching no such skills (n = 25)[5]. Figure 8 presents a visualization of these results.

The relatively high proportion of instructors who are not explicitly teaching any GenAI-based skill (25 instructors, or 46.43%) may indicate soft integration, where students are allowed to use GenAI independently without much structure or guidance. The low frequency of instructors teaching GenAI-based problem decomposition is also noteworthy, since this is a higher-order, transferable skill, that could facilitate students' application of GenAI in complex and open problems. A similar observation can be made for critical thinking, which was reported by only a single instructor; however, this skill was not one of the pre-specified options, and it is possible many instructors do not consider it a GenAI-specific skill.

8.5 Changes to Assessment Practices

Table 28 shows the various changes to assessment practice instructors are considering based on their responses to question 13 of the survey. questions 6 to 8 provided further insights into changes within specific courses.

In a Data Visualization course where students were allowed to use GenAI without restrictions, the instructor shifted their focus from marking code to other aspects:

[5]The response 'None of the above' is interpreted as 'No skills' being taught, since an 'other' option which allowed skill specification was also available but not used by those respondents.

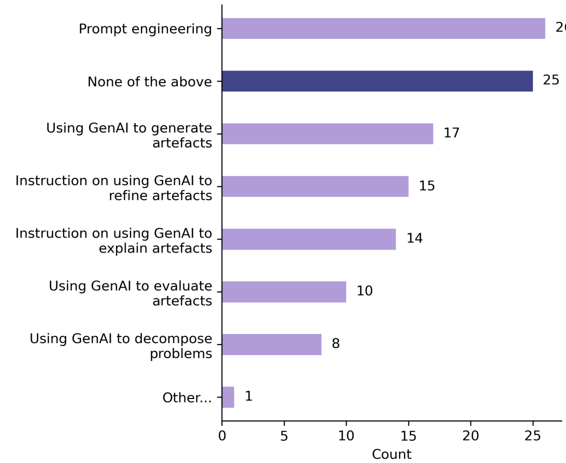

Figure 8: GenAI-related skills instructors teach in their courses.

R101 — "I don't grade student code at all; I only grade the creativity and quality of the visualizations that students make."

The answers of other respondents suggest considering other aspects such as students' understanding of their own code / project and their ability to think critically. For instance, an instructor teaching Algorithms, Object-Oriented Programming and Mobile Computing rewards transparency of GenAI use and critical thinking in all of their modules:

R105 — "I give bonus points in the project if the students include an interaction log with an LLM that is (1) related with the project and (2) demonstrates critical thinking over the LLM replies. In particular, students must ask for two different solutions for the same problem and then decide which one is the best and explaining their decision."

Another instructor in Advanced Data Structures also relies on a proctored assessment as a means to test students' understanding. This approach allows students to make their own choices about the extent of GenAI usage:

R52 — "I stopped grading for correctness or giving much credit for them [projects]. Instead, students complete on their own (no autograder) and turn in only for completeness. Second, I created "Project Evaluations" which is a paper, in person, proctored assessment to test their understanding of the project. This allows students to use Gen AI tools and each other as much or as little as they'd like."

8.6 Changes to Learning Objectives

The survey responses to question 10 regarding revision of learning objectives in response to students using GenAI indicate that majority of the integrators (26 of 56; 84%) have not changed their objectives, as seen in Table 29. However, 22 instructors provided qualitative comments for question 11, which asked how they had changed, or planned to change, the learning objectives of their course in relation to GenAI usage. This section summarizes the findings from thematic analysis of their comments.

Category	Change of Assessment Strategy	#	Distribution
Assessment Tasks	Add an oral explanation of the deliverables	28	
	Design tasks that are harder for GenAI to solve	20	
	Ask for in-person feature changes (e.g., add a new feature to the delivered code)	13	
	Give more attention to validation of results (e.g., evidence of testing)	12	
	Create videos of their work	1	
Focus on Reflection	Emphasize responsible GenAI usage over GenAI-proofing	23	
	Give more attention to student reflection	20	
Change in Grading	Give more points to the process or development journey	19	
	Less weight in the grade for GenAI things	1	
Assessment Type	Use version control to monitor progress	4	
	Paper-based exams, labs and tests	3	
	More open-ended projects	1	
	No-Internet exams (either blue book or in an air-gapped testing room)	1	
	Use proctored exams	1	
None	No change	5	
Uncertain	Unsure about changes	2	

Table 28: Changes to assessment practice in response to GenAI.

Table 29: Revision of Learning Objectives

Revision Status	#	Distribution
No	26	
Not yet, but I'm planning	21	
Yes	8	
Not applicable	1	

Seven of the 22 responses to question 11 were not directly applicable to the topic of changes to learning objectives. Three respondents had plans to change learning objectives in future but had not fully decided how this would be done. The remaining (n = 12) responses focused on new learning objectives; the following themes were identified from these.

Evaluating and understanding code was added to learning objectives by several instructors (n = 10). This includes students 'evaluating GenAI outputs' and also evaluating their own code and code written by others, demonstrating their understanding of its workings. *"Critical thinking"* and *"problem solving"* codes were also included in this theme. Instructors described their plans:

> *R17 — "I would like to change the learning objectives to .. ensure that critical evaluation and analysis of existing or AI generated code is included as part of the learning outcomes."*

> *R57 — "increase critical thinking and code comprehension skills."*

Responsible and ethical use of GenAI was emphasized in learning objectives by several respondents (n = 5). This theme also included student reflection on their own practice. An instructor who leads a course in Computer Ethics stated:

> *R94 — "I am planning on adding something regarding ethical implications of AI as part of the learning objectives."*

AI competency was a theme representing a couple of responses. This involved adding learning objectives relating to appropriate usage of GenAI tools in courses. One instructor noted their intent to:

> *R94 — "Explicitly include something like 'appropriate use of AI tools' as part of the learning objective, as this is becoming an expected professional skill."*

Project and teamwork was another theme noted, emphasizing project management and software development processes rather than focusing on the final product in learning objectives. The theme includes codes such as *"Design/organise large projects"* (n = 1), *"More project oriented tasks"* (n = 1) *"Include teamwork"* (n = 1), *"Prototyping with GenAI"* (n = 1). One instructor summarized their learning objectives as follows:

> *R42 — "Objectives now involve teamwork, understanding/walking through code, making large organizational/design decisions in code projects."*

A shift in course focus was reported by four respondents. One instructor making such a shift encapsulated many of the above themes in their response:

> *R85 — "GenAI influenced these revisions by making it important for students to learn how to use AI responsibly and effectively, not just to build projects, but also to improve their problem-solving and code review skills. The updated learning objectives now aim to prepare students for a modern development environment where AI-assisted programming is becoming common."*

8.7 AI Policies

A total of 52 instructors provided qualitative feedback on their institution's or course-level GenAI policies. Their responses revealed a spectrum of approaches, ranging from outright encouragement to strict prohibition, as well as nuanced conditional allowances. The following six themes have been identified.

Policy presence describes whether a formal policy on GenAI use exists at all. While most of the respondents reported having clearly defined rules, a small number indicated that no formal guidelines are in place. This theme consists of two codes: *"Has policy"* ($n = 50$) and *"No policy"* ($n = 2$). These numbers indicate that, in most cases, instructors operate under some documented framework, at the course or institutional level. The presence of a policy can provide clarity, consistency, and a basis for addressing potential disputes, whereas the absence of one may leave unclear expectations. Notably, some "Has policy" responses described minimal restrictions, or blanket restrictions, suggesting these policies vary widely. Responses include:

> *R133 — "The GenAI usage policies vary by each course, and all clearly define dos and don'ts."*
>
> *R116 — "GenAI is not officially allowed."*

Mandatory, permissive or encouraging policies were among the most common. The dominant code, *"AI is permitted"* ($n = 21$), reflects a broad openness to GenAI integration. A smaller group explicitly encouraged use (*"Use of AI is encouraged"* ($n = 3$)), while one policy even mandated it (*"AI must be used for assignments"*). As one instructor described:

> *R35 — "Students may, and in fact, must, use Gen AI at least once on each assignment."*

Restrictive or prohibitive policies were also frequently reported, representing a substantial number of all responses ($n = 30$). These policies aim to limit GenAI use to protect assessment integrity and ensure original student work.

The theme includes codes such as *"AI must not be used for exams/tests"* ($n = 11$), *"AI use forbidden for entire tasks/assignments"* ($n = 3$), *"AI is discouraged other than as a supplementary learning tool"* ($n = 2$), *"AI can't be used for essay or presentation"* ($n = 1$), *"AI forbidden for certain assignments"* ($n = 1$), *"AI forbidden in class"* ($n = 1$), *"AI must not be used for graded work"* ($n = 2$), *"Can only use syntax/concepts taught in curriculum"* ($n = 2$), *"Forbidden"* ($n = 4$), *"Forbidden with exceptions"* ($n = 3$).

The diversity of restrictions, from complete bans to targeted limitations, indicates that many instructors are still balancing between acknowledging GenAI's availability and protecting critical learning moments from it. As one instructor explained:

> *R107 — "Students are allowed to use GenAI as a kind of supplementary "learning tool" and are told that GenAI-generated solutions for tasks such as assignments, homeworks, etc. lead to attempted deception/plagiarism, even though we are not able to check this."*

Conditional use based on task type was another recurring pattern, reflecting a balanced approach between permission and restriction. These policies allow GenAI in situations where it can support learning, such as brainstorming, debugging, or exploring alternative solutions. However, in these situations, instructors draw clear boundaries around assessments or contexts where independent problem-solving is essential.

This theme includes *"AI permitted for assignments"* ($n = 7$), *"AI is permitted for certain assignments"* ($n = 3$), *"AI is permitted for debugging"* ($n = 1$), *"AI is permitted for idea evaluation and finding sources only"* ($n = 1$), *"AI is permitted, unless explicitly forbidden"* ($n = 1$), *"AI permitted for small code snippets"* ($n = 1$), and *"AI permitted outside of class"* ($n = 1$). The range of conditional allowances suggests that many instructors see value in integrating AI as a scaffold rather than a substitute for skill development. As one instructor summarized their approach:

> *R85 — "Students are allowed to use GenAI tools (like ChatGPT or GitHub Copilot) for homework, studying, and project brainstorming, as long as they understand and can explain their work [...] However, GenAI tools are not allowed during exams or for direct code copying in major projects. All use of GenAI must be properly acknowledged."*

Policy communication emerged as a clear theme, showing that simply having a policy is not enough. A policy must be explicitly conveyed to students in a way that sets expectations and outlines consequences. Instructors in this category focus on making the rules transparent, often linking them to academic integrity standards and institutional guidelines.

This theme includes *"Document/attribute use"* ($n = 20$), *"Inform about negative consequences (plagiarism)"* ($n = 1$), and *"Inform students"* ($n = 1$). Such policies help ensure that students understand not only what is allowed and what is prohibited, but also why those boundaries exist. The emphasis on communication also suggests a preventive approach, aiming to reduce unintentional misuse of GenAI by clarifying its appropriate role in coursework.

> *R71 — "The extent to which Gen AI can be used will be clearly stated in the assignment. Gen AI tools used to generate coursework must be clearly marked, cited, and acknowledged; failure to do so could constitute plagiarism."*
> *R53 — "I plan to incorporate instructions for sensible use, if they really want to use it, and explain how it might hinder their learning if they use it heavily."*

Responsible use expectations focus on fostering ethical, transparent, and critical engagement with GenAI tools. Rather than banning GenAI outright or allowing unrestricted use, these policies establish behavioral expectations designed to ensure students remain active agents in their own learning.

This theme includes *"Students should be able to explain their work"* ($n = 5$), *"Inform students about GenAI"* ($n = 5$), *"Responsible prompts"* ($n = 2$), *"Responsible use"*($n = 1$), and *"Students should double-check AI responses"* ($n = 1$). Such guidelines encourage learners to critically evaluate AI-generated content, attribute its use appropriately, and maintain ownership over the problem-solving process. Instructors adopting this approach often see GenAI as a tool to be mastered responsibly, not a shortcut to bypass skill development. One respondent noted:

> *R119 — "GenAI is allowed but should be acknowledged and used responsibly."*

8.8 Open GenAI Resources

This section summarizes the findings from question 15 about publicly available resources for using GenAI in post-introductory courses. The results revealed there are almost no resources available on this topic:

Table 30: Transparency requirements for use of GenAI.

Transparency in GenAI use	#	Distribution
Declaration of any use	36	
Attribution of GenAI-generated content	28	
Summary of usage	19	
Reflection on influence on process or results	19	
No requirements	8	
Full log of interactions or transcript	7	
They are not permitted.	1	

> *R41 — "I've looked, they are all really bad. They are about 18 months behind the state of the art and about 18 months behind the students."*

However, a total of eight resources were mentioned in the survey. Out of these, five are still under development and have not been published yet. Therefore, they were not included in this paper. The remaining three resources are complete and publicly available:

- **Machine Learning course**[6] provides an overview of a self-paced course designed to teach essential machine learning concepts, algorithms, and real-world applications.
- **AlphaZero Project**[7] is an assignment designed to guide students in building an AI system inspired by AlphaZero, developed by DeepMind.
- **LensQL**[8] is AI-powered tool designed to support learning and debugging in PostgreSQL.

Machine Learning course[6]. The first available resource is a complete course used for teaching machine learning at a university. The course includes 24 modules, covering topics from basic to advanced levels. Exercises are marked with difficulty levels, and learning materials include videos, interactive slides, and quizzes for theory. For practice, students can access example slides, Jupyter notebooks, in-IDE exercises, or participate in Kaggle competitions. The course requires basic Python programming skills and foundational knowledge of linear algebra and calculus. The materials are regularly updated, with the author revising them each semester.

AlphaZero Project[7]. The second resource is a machine learning assignment that can be used as part of a machine learning course. The assignment introduces the core principles of AlphaZero's architecture by applying them to simpler board games like TicTacToe and Connect Four. Students will learn key concepts such as Monte Carlo Tree Search (MCTS), deep reinforcement learning, and neural network design. The project involves implementing a simplified AlphaZero algorithm, creating a Residual Neural Network (ResNet) for game evaluation, using MCTS for move exploration, and training the system through self-play to iteratively improve its strategies. The assignment is provided as an IDE course, which students need to open and work on in the PyCharm IDE.

LensQL[8]. The last available resource is a tool designed to help students learn SQL. The tool emphasizes error-based pedagogy, helping students understand and learn from syntax, logic, and semantic mistakes in their queries. It assists students in refining their query logic and provides instructors with insights into student learning processes and challenges. The tool can be used as part of a database systems course.

[6]Machine Learning course https://avalur.github.io/mlcourse/index.html
[7]AlphaZero Project: https://github.com/avalur/avalur.github.io/blob/master/advanced_ml/hw_alpha_zero.md
[8]LensQL: https://lensql.ponzidav.com

9 Discussion

This working group was motivated by the goal to discover how computer science education is changing and adapting to the emergence of GenAI in the post-introductory courses. Specifically, we set out to answer the following research questions:

RQ1. What research literature exists on the integration of GenAI into post-introductory computing courses in terms of subjects and activities?
RQ2. How have instructors integrated GenAI into their post-introductory computing courses?
RQ3. What are the higher-level trends on the use of GenAI within post-introductory computing courses in terms of novel activities, changes in assessment, skills development, usage policies and learning objectives?

9.1 Literature Review Findings

In response to RQ1, the findings of the systematic literature review map the current landscape of GenAI use in post-introductory computing courses, identifying both the subjects in which it has been reported and the forms of integration highlighted in the literature. Although GenAI is still in the early stages of integration into the CS curriculum, a clear trend emerges: Software Engineering dominates the literature with 30 papers (37% of our dataset), followed by Databases (10 papers), Human–Computer Interaction (10 papers), and Algorithms & Data Structures (8 papers).

This distribution likely reflects both the immediate practical relevance of GenAI to software development workflows and the strength of existing research communities in these areas. Notably, this trend echoes previous research on where the focus of computing education has been applied [90, 113]. There is a clear and wide gap here as many subjects core to the CS curriculum (programming languages, compilers, computer architecture, and so on) are not represented.

While this imbalance is evident, the analysis also revealed strong similarities in how GenAI has been employed across different subjects. These similarities informed the development of our taxonomy of use, which was then applied to the identified activities. A clear trend found is the predominance of *generation*, which accounts for 33% of activities. Given the current exploratory phase of GenAI adoption, this is unsurprising: each subject area is still probing what can be generated and how such outputs might support learning.

Evaluation and interpretation together account for the next 34% of activities. A cross-subject pattern involves the deliberate use of flawed or incomplete GenAI outputs that students are tasked with critiquing, debugging, or improving. This pairing aligns well with professional practice, where the ability to critically assess and refine generated artifacts is increasingly important. Interpretation also featured prominently: when GenAI is used appropriately, it can enable students to engage with significantly more complex problems, such as navigating large codebases or understanding intricate systems, well beyond the scope of typical instructional examples.

Another noteworthy finding is the emergence of higher-order activities, such as simulation, role-playing, and the creation of interactive experiences that would otherwise be difficult to design. While these uses currently appear only sporadically across subjects, they point to promising opportunities for developing rich learning experiences that engage students at a deeper level.

Although reflection was less frequently observed, it was present across multiple subjects. Here, GenAI was not typically used directly; rather, students were encouraged to reflect critically on how their discipline is being reshaped by the challenges and opportunities presented by GenAI.

Finally, across subjects, several common positive sentiments emerged. GenAI was frequently described as providing effective scaffolding, enabling students to approach tasks that might otherwise have been beyond their current abilities. In many cases, this also translated into higher levels of engagement, as students were able to experiment more freely and receive immediate feedback. A further recurring theme was the role of GenAI in extending instructor support: by automating routine explanations or generating practice material, it was seen as a way to scale assistance in contexts where staff–student ratios are often a limiting factor.

In contrast, a set of shared concerns was also evident. Students using GenAI often struggled with more advanced or nuanced tasks, raising doubts about its reliability for higher-level learning objectives. Its tendency to produce repetitive or formulaic outputs was also highlighted, potentially limiting the variety and creativity of student work. Perhaps most prominently, there were concerns around fostering over-reliance, with both students and instructors expressing unease that habitual dependence on GenAI might undermine the development of independent problem-solving skills.

9.2 Instructor Survey Findings

In response to RQ2, the findings from the instructor survey provides a compliment to the review and captures a more intimate account of how instructors are integrating GenAI and adapting their established practices in light of its impact.

The instructor survey yielded 118 responses from computing educators, with nearly half (47%) already integrating GenAI into their post-introductory courses and the remaining 53% not integrating. This balance suggests that while adoption is still emerging, GenAI is no longer a niche phenomenon. Instead, it is becoming a live issue for most instructors, regardless of whether they have already integrated such tools.

The most frequently taught subjects among respondents were Software Engineering, Algorithms & Data Structures, and Object-Oriented Programming, echoing the pattern of results from the literature review. Notably, there was no systematic subject bias between integrators and non-integrators: instructors in both groups were distributed across the same core subjects.

Non-integrators cited several overlapping reasons for hesitation. Concerns about undermining foundational skills and fostering over-reliance (n = 22) point to enduring anxieties around automation and the preservation of core disciplinary knowledge. Equally, gaps in knowledge and guidance (n = 17) highlight the unevenness of institutional support: many instructors appear willing but uncertain about how to proceed responsibly. The fact that most non-integrators (84%) nevertheless plan to adopt GenAI in the near future suggests that hesitation is not rejection. Instead, many are adopting a "wait-and-see" approach, looking for clearer evidence, policies, or pedagogical frameworks before proceeding. This finding aligns with broader observations in computing education, where adoption curves often lag until early exemplars and resources are available.

Among integrators, current practices emphasize responsible use in assignments (n = 38), critical evaluation (n = 27), and brainstorming (n = 26). The predominance of generation (43%) and evaluation (26%) reinforces the exploratory stage of adoption: instructors are first testing what can be produced, and then immediately engaging students in critique. Higher-order uses, such as simulation or role-playing, remain rare but may signal directions for future innovation. Taken together, these practices suggest that GenAI is being used less as a replacement for traditional problem-solving and more as a tool for provoking critical engagement and supporting ideation.

The teaching of explicit GenAI-related skills is uneven. While prompt engineering (n = 26) was the most commonly mentioned, nearly half of instructors (46%) reported teaching no explicit skills at all. This divergence points to a tension: some instructors see GenAI literacy as a skill in its own right, while others view GenAI as an external tool not requiring dedicated instruction in a subject-specific context. Assessment adaptations, such as adding oral components, designing GenAI-resistant tasks, or shifting emphasis from products to processes, reflect an awareness that traditional evaluation models may be insufficient in an AI-mediated learning environment. These changes are small but significant nudges toward more authentic assessment practices that could benefit computing education more broadly.

Nearly 90% of respondents reported the existence of formal policies, ranging from outright bans to mandates for use. The most common requirement was transparency through student declarations of GenAI use (n = 36). This diversity of policy approaches reflects an unsettled landscape: while institutions recognize the urgency of responding to GenAI, consensus on best practice has not yet emerged. The emphasis on transparency suggests a pragmatic recognition that GenAI use cannot be eradicated, only managed.

Instructors reported both encouraging outcomes and pressing challenges. On the positive side, GenAI appeared to foster critical evaluation when scaffolded, increased productivity and engagement, and in some cases improved academic outcomes. Yet challenges persist: students often accepted GenAI outputs uncritically (n = 8), relied on the technology excessively (n = 8), and struggled to judge quality. These findings echo concerns from the non-integrators, reinforcing the view that effective use of GenAI depends not on access alone but on carefully designed pedagogy that cultivates independent judgment.

Perhaps the most unfortunate finding is the lack of quality educational resources that instructors are either aware of or actively using. Only three publicly available materials were identified from our respondents. Most of the resources known to respondents were deemed outdated or inadequate. This gap represents a possible barrier to adoption and an opportunity: without shared, field-tested resources, individual instructors are left to improvise, often duplicating effort. At the same time, the need for resources highlights a clear area for collective investment by the computing education community.

Taken together, these survey findings portray a field in transition. Instructors are neither dismissing GenAI nor embracing it uncritically. Instead, they are experimenting at the margins, cautiously adapting policies, and seeking to reconcile the opportunities for engagement and productivity with the risks of dependency and superficial learning.

9.3 Integrative Findings

In response to RQ3, we integrate the findings from both review and survey to synthesize higher-level trends that are emerging with regard to the integration of GenAI into post-introductory courses.

9.3.1 ***Transferrable Activity Patterns:*** From the activities extracted from the review and survey, we can begin to identify portable designs that adapt across subjects while addressing subject-specific pedagogical challenges:

Compound vs simple: A very general pattern that emerged is that GenAI activities can become sophisticated in their orchestration by combining multiple simple activities into a more complex compound activity with GenAI being used in several different ways at several different activity stages.

Generate then evaluate: Perhaps the most prevalent activity paring across subjects were students must use GenAI to generate an artifact and then either: evaluate it themselves or use GenAI to assist them in evaluating it.

Critique then create: Students first evaluate and repair GenAI-produced artifacts, then produce their own with tighter constraints and evidence requirements.

Test-first with GenAI support: Students write tests/specifications, use GenAI to propose implementations or traces, then reconcile differences. This maintains focus on understanding while leveraging AI capabilities.

Dual-path activities: Complete a task manually and with GenAI; compare outputs, effort, and error profiles; reflect on when GenAI helps or harms. This provides concrete experience with the scaffolding vs. short-cutting tension.

Experience-based agents: Use structured prompts or role cards (e.g., reviewer, debugger, planner, interviewer) to guide interactions and simulate subject specific experiences previously impossible without human participation.

Prompt reproducibility: Require students to submit minimal reproducible prompts and seed context; evaluate clarity, constraints, and safety.

9.3.2 ***Paradoxes and Tensions:*** At the same time, there are concerning paradoxes and tensions that can emerge, revealing the double-edged nature of using GenAI in pedagogical contexts:

The Critical Evaluation Paradox: A curious finding from our survey is what we term the "critical evaluation paradox": while instructors report success in teaching students to critically evaluate GenAI outputs ($n = 8$), an equal number report that students fail to develop these crucial skills ($n = 8$). This suggests that successful GenAI integration requires more than simply asking students to "think critically" about AI-generated content.

Successful implementations appear to share several characteristics: explicit instruction in evaluation criteria and techniques, structured activities that compare GenAI outputs with human-generated alternatives, clear rubrics and guidelines for assessment, and iterative practice with immediate feedback.

The Scaffolding versus Shortcutting Tension: Our findings reveal a fundamental tension in GenAI integration: the same tools that can provide valuable scaffolding for learning can also enable academic shortcutting. The determining factor appears to be instructional design rather than the technology itself. Effective scaffolding implementations provide instructor-designed prompts, require reflection and documentation of GenAI usage, focus on process rather than product in assessment, and maintain clear boundaries around appropriate use.

9.3.3 ***Skills Development Framework.*** Both our literature review and survey data highlight the emergence of new skill requirements for students in GenAI-enhanced computing education:

In terms of **Technical Skills**:

- Prompt engineering and iterative refinement techniques
- Critical evaluation of AI-generated artifacts using domain-specific criteria
- Understanding of GenAI capabilities and limitations across different computing contexts
- Integration of GenAI tools into professional development workflows

In terms of **Metacognitive Skills**:

- Reflection on GenAI usage and its impact on learning processes
- Recognition of when GenAI assistance is appropriate versus counterproductive
- Development of GenAI literacy and ethical reasoning in professional contexts
- Calibration of confidence in GenAI-assisted vs. independent work

9.3.4 ***Impacts of Assessment and Usage Policies:*** The survey revealed that GenAI integration is already reshaping both assessment practices and course-level usage policies. Instructors are experimenting with new forms of evaluation that emphasize process over product and introduce safeguards against over-reliance on GenAI. At the same time, policies governing GenAI use remain highly varied, yet certain common patterns are emerging across disciplines. The following lists summarize the key trends in assessment adaptations and usage regulations.

In terms of **Assessment:**

- **Shift from product to process** – oral explanations, development journey documentation, reflection requirements
- **GenAI-resistant task design** – harder problems, in-person feature modifications, proctored components
- **Transparency mandates** – require some form of GenAI usage declaration
- **Multi-modal assessment** – combining traditional exams with GenAI-assisted projects

In terms of **Usage Policies:**

- **Conditional permissions prevail** – task-specific allowances rather than blanket policies

- **Responsible use emphasis** – transparency and attribution requirements standard
- **Subject-agnostic approaches** – similar policy patterns across disciplines
- **Institutional vs. course-level variation** – mixed governance approaches

9.3.5 ***Implications for Learning Objectives.*** Few papers explicitly raise or revise Learning Objectives (LOs) despite the impacts of GenAI, yet our survey found that instructors who did modify their courses focused on specific competencies, which might include some that align with our taxonomy of GenAI use.

Critical evaluation and verification: Detecting, localizing, and correcting GenAI errors; triangulating with tests, specifications, or multiple tools - including evaluating and understanding both AI-generated and human-written code.

Responsible and effective GenAI use: Crafting reproducible prompts, documenting provenance, and articulating when and why to trust outputs - encompassing responsible and ethical use of GenAI tools as emerging professional skills.

Iterative improvement: Using GenAI for feedback and refinement while maintaining human oversight and standards of evidence - developing enhanced critical thinking and problem-solving capabilities.

Design and planning with GenAI: Leveraging GenAI to expand option sets, surface constraints, and reason about trade-offs - particularly valuable in complex system design tasks.

Communication about GenAI-assisted work: Explaining decisions, limitations, and ethical considerations to technical and non-technical audiences - essential for professional practice.

Aligning activities to explicit LOs clarifies permissible GenAI support, aids fair assessment, and helps students internalize where GenAI can accelerate learning versus where independent mastery is required.

9.4 Implications of Findings

In brief, we can highlight several major implications:

(1) Post-introductory subjects are developing and evaluating subject-specific activities that integrate GenAI in novel ways.
(2) These activities range from simple to complex, with emerging experience-based activities representing an especially promising development for learning.
(3) Common aspects of these activities can be observed across subjects, suggesting opportunities for cross-disciplinary inspiration and adaptation.
(4) Several foundational subjects were not captured by the instruments of this report, leaving notable gaps and opportunities to design additional subject-specific activities.
(5) Most instructors have either already integrated GenAI or are actively moving toward integration. While some remain hesitant, they represent a minority.
(6) Skills, assessment practices, usage policies, and learning outcomes remain in a fluid state. The literature offers limited guidance, and our findings suggest that instructors are only beginning to navigate these changes within their own personal and professional contexts. This situation likely mirrors introductory courses, where both students and instructors are adapting in real time to the evolving presence of GenAI.
(7) A third actor is emerging in educational settings: alongside students and instructors, GenAI functions as a flexible and adaptive agent, elevating its role from tool to mediator of learning interactions.

9.5 Working Group Recommendations

The following recommendations emerge from our synthesis of the literature and survey findings. While some remain exploratory, we hope they can stimulate further debate and concrete action within the community.

(1) In line with broader CSEd research, introductory courses dominate the research literature (we identified over 200 relevant papers), while post-introductory subjects remain underrepresented. The same imbalance is visible in current GenAI work. We encourage the community to build on the promising activities documented in this report and expand contributions in post-introductory courses.
(2) We recommend developing shared repositories for GenAI-based activities, inspired by initiatives such as *Nifty Assignments* and the *Canterbury Question Set*. Such platforms would enable instructors to share, adapt, and validate activities in a more open and accessible manner.
(3) We encourage instructors to adopt, adapt, and report on the activities highlighted in this report, including providing replication and validation data. This evidence base will help identify which approaches are most effective for supporting student learning across diverse contexts.
(4) Our analysis of literature and instructor practices informed the proposal of a taxonomy of GenAI use aspects. We suggest that applying such a taxonomy can help structure the design of activities in a way that makes them more interpretable and transferable across subjects. The taxonomy itself is a first step towards establishing a more theoretical lens for studying use of GenAI in computing education.
(5) We recommend prioritizing the development of high-quality educational resources to address the severe resource gap identified in the survey. Collaborative, community-driven materials will reduce duplication of effort and support more equitable access to effective practices.
(6) Finally, we encourage dialogue around assessment and policy design. Instructors are already experimenting with GenAI-resistant tasks, process-focused evaluation, and varied policy approaches, but there is little consensus. Community-driven guidelines could help balance responsible use, academic integrity, and innovation in pedagogy.

9.6 Limitations

9.6.1 ***Limitations of the Systematic Literature Review.*** The WG team acknowledges that the results presented here are necessarily constrained by the methodological decisions and resources available during this work. Key limitations include:

- The challenge of constructing search strings that identified all relevant publications without generating an unmanageable number of false positives.

- Restricting the search to two databases, which may have omitted relevant publications found in other sources.
- The increasing tendency for work to be published first on preprint servers such as arXiv, which were not systematically included.
- Alternative search strategies (e.g., snowballing) may have provided broader coverage; this was raised in our ITiCSE discussions (cf. Tony Clear, personal communication).
- Inconsistencies in paper classification and judgment across a team of 15 reviewers, despite calibration efforts.
- Variability in the methodology used for clustering subjects and analyzing subject categories.
- The treatment of borderline cases, where inclusion or exclusion decisions were not always straightforward.
- The taxonomy of aspects emerged inductively during analysis rather than being predefined. While inductive coding is appropriate in a new and underexplored area, a deductive approach might have yielded greater consistency.

9.6.2 ***Limitations of the Instructor Survey.*** The instructor survey also carries several limitations that should temper interpretation of the results:

- As with all voluntary surveys, there is the possibility of self-selection bias, where those with stronger views or experiences regarding GenAI were more likely to respond.
- The survey relied on self-reported practices, which may not always reflect actual classroom implementation.
- The findings related with public resources reported on section 8.8 (Open GenAI Resources) and discussed on section 9.2 (Instructor Survey Findings) should be interpreted cautiously, as the survey workflow, which required the instructor to locate and paste URLs, might have contributed to some degree of underreporting.
- The dataset, while international, may not be fully representative of all computing education contexts, particularly in regions or institutions where GenAI integration is less advanced.
- Several questions invited open-text responses, which can introduce variability in interpretation and coding despite efforts to normalize categories. Furthermore, we accept there may have been a researcher bias in our selection, however multiple authors reviewed the responses for balance.
- Because responses came from a convenience (non-probability) sample and we counted themes from coded open-ended answers, the numbers are indicative instead of population estimates. The results might have differed if we asked participants to rate each theme directly.

9.7 Future Work

A clear outcome of this working group is the recognition that the computing education community is still in the early stages of transitioning toward a future where GenAI is an integrated component in the teaching/learning process preparing students for careers of the future. This work provides a comprehensive overview of the state of the art in 2025, but it would be foolish to expect that similar efforts will are not needed in the coming years to capture the progress and maturation of GenAI integration across all subjects and levels.

This team's crystal ball sees potential benefit to the educational community from:

- Longitudinal studies of students who used / didn't use GenAI in courses
- Capabilities of GenAI in both solving problems and supporting teaching and learning in specific courses
- Studies on the evolution of desirable career skills and employer expectations

Additionally, the team sees a need for a clear set of GenAI-specific learning outcomes for courses and computing programs that could guide the evolution of degree programs.

As the field develops, it is likely that individual subdisciplines will need to engage in their own introspection, examining both how GenAI is being used and how it is shaping disciplinary practice. Looking further ahead, perhaps a decade from now, our collective efforts may be complemented by reviews that weave together these independent yet interconnected threads of evolution.

10 Conclusion

GenAI has already had a profound impact on computer science education in its short history, and its influence will continue to grow. In this work, we focused specifically on how that impact is being addressed in post-introductory courses. Our review revealed an emerging landscape of creative and engaging activities that leverage GenAI to open new frontiers for teaching and learning. Students are now asked not only to generate and interpret, but also to evaluate, brainstorm, design, refine, simulate and reflect with the support of GenAI as they engage with disciplinary content.

At the same time, both researchers and instructors express mixed views on how best to integrate GenAI into the curriculum in ways that are responsible, rational, effective, and ethical. The challenge is compounded by the rapid pace of change: today's students already require skills that were difficult to anticipate even five years ago.

Taken together, our findings suggest that the field of computer science education is moving forward both confidently and cautiously toward integrating GenAI across the curriculum. Yet it is equally clear that much work remains to be done.

Use of AI Declaration

The authors of this report have followed the ACM Policy on Authorship GenAI was used in the following ways in this report:

- Smoothing flow and academic tone of text throughout the major report sections. Note we used an internal policy: we write-first, GenAI suggests edit second, we write/edit last. All grammar bugs belong to the humans.
- Providing guesses of a paper's focus based on title/abstract. Source code is publicly available for scrutiny on GitHub.
- Providing review of a paper's content using a common structured prompt – note all papers were reviewed by authors and both the GenAI summary and human summary were used in various places.
- Detecting themes in qualitative data derived from the original data – note no respondent data was sent to any GenAI.

- Producing the code to visualize statistics in charts and tables in section 6 and section 8.

"The Rest of the Robots"

Some readers may find the first part of title of this report familiar as "The Rest of the Robots" is the title of a collection of short stories by Isaac Asimov [7]. The back cover of the 2018 paperback edition reads, in part: "*Asimov's collection of short stories uncovers the practical and ethical issues humanity will encounter in a robotic future*". While this WG leaves the topic of ethics for others to address, it is hoped this report uncovers practical issues humanity faces in this robotic present. Additional inspiration for the title of this paper was derived from a previous ITiCSE WG paper, "The Robots are Here ..." [94], which itself was inspired by an ACE paper, "The Robots are Coming ..." [36]. It is our pleasure to acknowledge their work, implicitly and explicitly.

Acknowledgments

Partially funded by the European Union's DIGITAL-2021-SKILLS-01 Programme under grant agreement no. 101083594.

References

[1] Willem Aerts, George Fletcher, and Daphne Miedema. 2024. A Feasibility Study on Automated SQL Exercise Generation with ChatGPT-3.5. In *Proceedings of the 3rd International Workshop on Data Systems Education: Bridging Education Practice with Education Research* (Santiago, AA, Chile) *(DataEd '24)*. Association for Computing Machinery, New York, NY, USA, 13–19. doi:10.1145/3663649.3664368

[2] Friday Joseph Agbo, Chris Olivia, Godsalvation Oguibe, Ismaila Temitayo Sanusi, and Godwin Sani. 2025. Computing education using generative artificial intelligence tools: A systematic literature review. *Comput. Educ. Open* 100266 (June 2025), 100266.

[3] Ildar Akhmetov and Mirjana Prpa. 2025. Simulating Requirement Elicitation: Development and Evaluation of a Persona-Based Tool. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 2* (Pittsburgh, PA, USA) *(SIGCSETS 2025)*. Association for Computing Machinery, New York, NY, USA, 1357–1358. doi:10.1145/3641555.3705250

[4] Mohammad T. Alshammari. 2025. An Investigation into ChatGPT-Based Instructional Feedback on Programming Projects. In *2025 4th International Conference on Computing and Information Technology (ICCIT)*. 624–628. doi:10.1109/ICCIT63348.2025.10989326

[5] Pasquale Ardimento, Mario Luca Bernardi, Marta Cimitile, and Michele Scalera. 2024. Enhancing Software Modeling Learning with AI-Powered Scaffolding. In *Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems* (Linz, Austria) *(MODELS Companion '24)*. Association for Computing Machinery, New York, NY, USA, 103–106. doi:10.1145/3652620.3687776

[6] Utkarsh Arora, Anupam Garg, Aryan Gupta, Samyak Jain, Ronit Mehta, Rupin Oberoi, Prachi, Aryaman Raina, Manav Saini, Sachin Sharma, Jaskaran Singh, Sarthak Tyagi, and Dhruv Kumar. 2025. Analyzing LLM Usage in an Advanced Computing Class in India. In *Proceedings of the 27th Australasian Computing Education Conference (ACE '25)*. Association for Computing Machinery, New York, NY, USA, 154–163. doi:10.1145/3716640.3716657

[7] Isaac Asimov. 2018. *The Rest of the Robots*. HarperCollins, New York. Originally published in 1964.

[8] Jose Barambones, Cristian Moral, Angélica de Antonio, Ricardo Imbert, Loïc Martínez-Normand, and Elena Villalba-Mora. 2024. ChatGPT for Learning HCI Techniques: A Case Study on Interviews for Personas. *IEEE Transactions on Learning Technologies* 17 (2024), 1460–1475. doi:10.1109/TLT.2024.3386095

[9] Brett A. Becker and Keith Quille. 2019. 50 Years of CS1 at SIGCSE: A Review of the Evolution of Introductory Programming Education Research. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (Minneapolis, MN, USA) *(SIGCSE '19)*. Association for Computing Machinery, New York, NY, USA, 338–344. doi:10.1145/3287324.3287432

[10] Alba Bisante, Venkata Srikanth Varma Datla, Emanuele Panizzi, Gabriella Trasciatti, and Stefano Zeppieri. 2024. Enhancing Interface Design with AI: An Exploratory Study on a ChatGPT-4-Based Tool for Cognitive Walkthrough Inspired Evaluations. In *Proceedings of the 2024 International Conference on Advanced Visual Interfaces* (Arenzano, Genoa, Italy) *(AVI '24)*. Association for Computing Machinery, New York, NY, USA, Article 41, 5 pages. doi:10.1145/3656650.3656676

[11] Uwe M. Borghoff, Mark Minas, and Jannis Schopp. 2025. Generative AI in Student Software Development Projects: A User Study on Experiences and Self-Assessment. In *Proceedings of the 6th European Conference on Software Engineering Education (ECSEE '25)*. Association for Computing Machinery, New York, NY, USA, 161–170. doi:10.1145/3723010.3723012

[12] Meryem Boubakri and Khalid Nafil. 2024. Enhancing Student Learning in Scrum Projects with Generative AI Assistance. In *2024 21st International Conference on Information Technology Based Higher Education and Training (ITHET)*. 1–8. doi:10.1109/ITHET61869.2024.10837605

[13] Allan Brockenbrough and Dominic Salinas. 2024. Using generative AI to create user stories in the software engineering classroom. In *2024 36th International Conference on Software Engineering Education and Training (CSEE&T)* (Würzburg, Germany). IEEE, 1–5.

[14] Jonathan W. Browning, John Bustard, Neil Anderson, and Leo Galway. 2024. A Data Science Course Utilizing GenAI. In *2024 IEEE Frontiers in Education Conference (FIE)*. 1–7. doi:10.1109/FIE61694.2024.10893452

[15] Doga Cambaz and Xiaoling Zhang. 2024. Use of AI-driven code generation models in teaching and learning programming: A systematic literature review. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1* (Portland OR USA). ACM, New York, NY, USA, 172–178.

[16] Cecilia Ka Yuk Chan and Wenjie Hu. 2023. Students' voices on generative AI: Perceptions, benefits, and challenges in higher education. *International Journal of Educational Technology in Higher Education* 20, 1 (2023), 43 pages.

[17] Yue Chen, Kok Keong Chai, Jonathan Loo, Reza Moosaei, and Joel Obstfeld. 2025. GenAI-Empowered Group-Based Authentic Assessment for Network Engineering Courses. In *2025 IEEE Global Engineering Education Conference (EDUCON)*. 1–7. doi:10.1109/EDUCON62633.2025.11016382

[18] Zixin Chen, Jiachen Wang, Meng Xia, Kento Shigyo, Dingdong Liu, Rong Zhang, and Huamin Qu. 2025. StuGPTViz: A Visual Analytics Approach to Understand Student-ChatGPT Interactions. *IEEE Transactions on Visualization and Computer Graphics* 31, 1 (Jan 2025), 908–918. doi:10.1109/TVCG.2024.3456363

[19] Bruno Pereira Cipriano and Pedro Alves. 2023. GPT-3 vs Object Oriented Programming Assignments: An Experience Report. In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1* (Turku, Finland) *(ITiCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 61–67. doi:10.1145/3587102.3588814

[20] Bruno Pereira Cipriano and Pedro Alves. 2024. LLMs Still Can't Avoid Instanceof: An Investigation Into GPT-3.5, GPT-4 and Bard's Capacity to Handle Object-Oriented Programming Assignments. In *Proceedings of the 46th International Conference on Software Engineering: Software Engineering Education and Training* (Lisbon, Portugal) *(ICSE-SEET '24)*. Association for Computing Machinery, New York, NY, USA, 162–169. doi:10.1145/3639474.3640052

[21] Tony Clear, Åsa Cajander, Alison Clear, Roger McDermott, Mats Daniels, Monica Divitini, Matthew Forshaw, Niklas Humble, Maria Kasinidou, Styliani Kleanthous, Can Kultur, Ghazaleh Parvini, Mohammad Polash, and Tingting Zhu. 2025. AI Integration in the IT Professional Workplace: A Scoping Review and Interview Study with Implications for Education and Professional Competencies. In *2024 Working Group Reports on Innovation and Technology in Computer Science Education* (Milan, Italy) *(ITiCSE 2024)*. Association for Computing Machinery, New York, NY, USA, 34–67. doi:10.1145/3689187.3709607

[22] Lee Clift and Olga Petrovska. 2025. Learning without Limits: Analysing the Usage of Generative AI in a Summative Assessment. In *Proceedings of the 9th Conference on Computing Education Practice (CEP '25)*. Association for Computing Machinery, New York, NY, USA, 5–8. doi:10.1145/3702212.3702214

[23] Juan Ortiz Couder, William C Pate, Daniel A Machado, and Omar Ochoa. 2024. Incorporating AI in the teaching of requirements tracing within software engineering. In *2024 IEEE Frontiers in Education Conference (FIE)* (Washington, DC, USA). IEEE, 1–8.

[24] Aaron S Crandall, Bryan J Fischer, and Johannah L Crandall. 2024. WIP: ARTful insights from a pilot study on GPT-based automatic code reviews in undergraduate computer science programs. In *2024 IEEE Frontiers in Education Conference (FIE)* (Washington, DC, USA). IEEE, 1–5.

[25] Claudio Cubillos, Rafael Mellado, Daniel Cabrera-Paniagua, and Enrique Urra. 2025. Generative Artificial Intelligence in Computer Programming: Does It Enhance Learning, Motivation, and the Learning Environment? *IEEE Access* 13 (2025), 40438–40455. doi:10.1109/ACCESS.2025.3532883

[26] Javier Cámara, Javier Troya, Julio Montes-Torres, and Francisco J. Jaime. 2024. Generative AI in the Software Modeling Classroom: An Experience Report With ChatGPT and Unified Modeling Language. *IEEE Software* 41, 6 (2024), 73–81. doi:10.1109/MS.2024.3385309

[27] Júlia Veiga da Silva, Simone André da Costa Cavalheiro, and Luciana Foss. 2024. Automata theory in computing education: A systematic review. *Simpósio Brasileiro de Informática na Educação (SBIE)* (2024), 301–313.

[28] Annisa Dwi Damayanti, Hamdan Gani, Feng Zhipeng, Helmy Gani, Sitti Zuhriyah, Nurani, Nurhayati Djabir, and Nur Ilmiyanti Wardani. 2025. Bibliometric and Content Analysis of Large Language Models Research in Software

Engineering: The Potential and Limitation in Software Engineering. *International Journal of Advanced Computer Science and Applications* 16, 4 (2025). doi:10.14569/IJACSA.2025.0160436

[29] Daniele De Bari, Giacomo Garaccione, Riccardo Coppola, Marco Torchiano, and Luca Ardito. 2024. Evaluating Large Language Models in Exercises of UML Class Diagram Modeling. In *Proceedings of the 18th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement* (Barcelona, Spain) *(ESEM '24)*. Association for Computing Machinery, New York, NY, USA, 393–399. doi:10.1145/3674805.3690741

[30] Ozgur Dedehayir and Martin Steinert. 2016. The hype cycle model: A review and future directions. *Technological Forecasting and Social Change* 108 (2016), 28–41. doi:10.1016/j.techfore.2016.04.005

[31] Iddo Drori, Sarah J. Zhang, Reece Shuttleworth, Sarah Zhang, Keith Tyser, Zad Chin, Pedro Lantigua, Saisamrit Surbehera, Gregory Hunter, Derek Austin, Leonard Tang, Yann Hicke, Sage Simhon, Sathwik Karnik, Darnell Granberry, and Madeleine Udell. 2023. From Human Days to Machine Seconds: Automatically Answering and Generating Machine Learning Final Exams. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Long Beach, CA, USA) *(KDD '23)*. Association for Computing Machinery, New York, NY, USA, 3947–3955. doi:10.1145/3580305.3599827

[32] Maria R. Ebling. 2024. Resources for Teaching Operating Systems: A Survey of Instructors and a Literature Review. *ACM Trans. Comput. Educ.* 24, 4, Article 46 (Oct. 2024), 28 pages. doi:10.1145/3688853

[33] Alessandro Fantechi, Stefania Gnesi, Lucia Passaro, and Laura Semini. 2023. Inconsistency detection in natural language requirements using ChatGPT: A preliminary evaluation. In *2023 IEEE 31st International Requirements Engineering Conference (RE)* (Hannover, Germany). IEEE.

[34] Laura Farinetti and Luca Cagliero. 2025. A Critical Approach to ChatGPT: An Experience in SQL Learning. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1* (Pittsburgh, PA, USA) *(SIGCSETS 2025)*. Association for Computing Machinery, New York, NY, USA, 318–324. doi:10.1145/3641554.3701932

[35] Tony Haoran Feng, Paul Denny, Burkhard Wuensche, Andrew Luxton-Reilly, and Steffan Hooper. 2024. More Than Meets the AI: Evaluating the performance of GPT-4 on Computer Graphics assessment questions. In *Proceedings of the 26th Australasian Computing Education Conference*. 182–191.

[36] James Finnie-Ansley, Paul Denny, Brett A. Becker, Andrew Luxton-Reilly, and James Prather. 2022. The Robots Are Coming: Exploring the Implications of OpenAI Codex on Introductory Programming. In *Proceedings of the 24th Australasian Computing Education Conference* (Virtual Event, Australia) *(ACE '22)*. Association for Computing Machinery, New York, NY, USA, 10–19. doi:10.1145/3511861.3511863

[37] Rosella Gennari and Soufiane Krik. 2024. Responsible Design of Socio-Technical Solutions with Social Design Students: a Case Study. In *Proceedings of the 2024 International Conference on Advanced Visual Interfaces* (Arenzano, Genoa, Italy) *(AVI '24)*. Association for Computing Machinery, New York, NY, USA, Article 50, 9 pages. doi:10.1145/3656650.3656663

[38] Çağdaş Evren Gerede. 2024. Are We Asking the Right Questions to ChatGPT for Learning Software Design Patterns?. In *2024 9th International Conference on Computer Science and Engineering (UBMK)*. 1092–1097. doi:10.1109/UBMK63289.2024.10773596

[39] Nasser Giacaman and Valerio Terragni. 2025. *Empowering Computing Students with Large Language Models by Developing an Escape Room Game.*

[40] Toshali Goel, Orit Shaer, Catherine Delcourt, Quan Gu, and Angel Cooper. 2023. Preparing Future Designers for Human-AI Collaboration in Persona Creation. In *Proceedings of the 2nd Annual Meeting of the Symposium on Human-Computer Interaction for Work* (Oldenburg, Germany) *(CHIWORK '23)*. Association for Computing Machinery, New York, NY, USA, Article 4, 14 pages. doi:10.1145/3596671.3598574

[41] Matei A. Golesteanu, Garrett B. Vowinkel, and Ryan E. Dougherty. 2024. Can ChatGPT pass a Theory of Computing Course?. In *Proceedings of the 2024 on ACM Virtual Global Computing Education Conference V. 1* (Virtual Event, NC, USA) *(SIGCSE Virtual 2024)*. Association for Computing Machinery, New York, NY, USA, 33–38. doi:10.1145/3649165.3690116

[42] Jamie Gorson Benario, Jenn Marroquin, Monica M. Chan, Ernest D.V. Holmes, and Daniel Mejia. 2025. Unlocking Potential with Generative AI Instruction: Investigating Mid-level Software Development Student Perceptions, Behavior, and Adoption. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1* (Pittsburgh, PA, USA) *(SIGCSETS 2025)*. Association for Computing Machinery, New York, NY, USA, 395–401. doi:10.1145/3641554.3701859

[43] Susmita Haldar, Mary Pierce, and Luiz Fernando Capretz. 2024. WIP: Assessing the effectiveness of ChatGPT in preparatory testing activities. In *2024 IEEE Frontiers in Education Conference (FIE)* (Washington, DC, USA). IEEE, 1–5.

[44] Khadija Hanifi, Orcun Cetin, and Cemal Yilmaz. 2023. On ChatGPT: Perspectives from Software Engineering Students. In *2023 IEEE 23rd International Conference on Software Quality, Reliability, and Security (QRS)* (Chiang Mai, Thailand). IEEE, 196–205.

[45] Sara Hashemi. 2025. *A New Conference Presented Scientific Papers Written and Reviewed by A.I. as an Experiment. Here's What Happened.* https://www.smithsonianmag.com/smart-news/a-new-conference-presented-scientific-papers-written-and-reviewed-by-ai-as-an-experiment-heres-what-happened-180987577/

[46] Matthew Hertz. 2010. What do "CS1" and "CS2" mean? investigating differences in the early courses. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education* (Milwaukee, Wisconsin, USA) *(SIGCSE '10)*. Association for Computing Machinery, New York, NY, USA, 199–203. doi:10.1145/1734263.1734335

[47] Chia-Ling Ho, Xin-Ying Liu, Yu-Wei Qiu, and Shih-Yang Yang. 2024. Research on Innovative Applications and Impacts of Using Generative AI for User Interface Design in Programming Courses. In *Proceedings of the 2024 International Conference on Information Technology, Data Science, and Optimization* (Taipei, Taiwan) *(I-DO '24)*. Association for Computing Machinery, New York, NY, USA, 68–72. doi:10.1145/3658549.3658566

[48] Muntasir Hoq, Yang Shi, Juho Leinonen, Damilola Babalola, Collin Lynch, Thomas Price, and Bita Akram. 2024. Detecting chatgpt-generated code submissions in a cs1 course using machine learning models. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*. 526–532.

[49] Irene Hou, Hannah Vy Nguyen, Owen Man, and Stephen MacNeil. 2025. The Evolving Usage of GenAI by Computing Students. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 2*. 1481–1482.

[50] Jelena Ilić, Mirjana Ivanović, and Aleksanda Klašnja-Milićević. 2024. The impact of ChatGPT on student learning experience in higher STEM education: A systematic literature review. In *2024 21st International Conference on Information Technology Based Higher Education and Training (ITHET)* (Paris, France). IEEE, 1–9.

[51] Pooriya Jamie, Reyhaneh HajiHashemi, and Sharareh Alipour. 2025. Utilizing ChatGPT in a Data Structures and Algorithms Course: A Teaching Assistant's Perspective. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '25)*. Association for Computing Machinery, New York, NY, USA, Article 567, 7 pages. doi:10.1145/3706599.3720291

[52] Peter Jamieson, Suman Bhunia, and Dhananjai M. Rao. 2023. With ChatGPT, Do We have to Rewrite Our Learning Objectives - CASE Study in Cybersecurity. In *2023 IEEE Frontiers in Education Conference (FIE)*. 1–5. doi:10.1109/FIE58773.2023.10343052

[53] Ravi Janumpally, Suparna Nanua, Andy Ngo, and Kenneth Youens. 2025. Generative artificial intelligence in graduate medical education. *Frontiers in Medicine* 11 (2025), 1525604.

[54] Ishika Joshi, Ritvik Budhiraja, Harshal Dev, Jahnvi Kadia, Mohammad Osama Ataullah, Sayan Mitra, Harshal D Akolekar, and Dhruv Kumar. 2024. ChatGPT in the Classroom: An Analysis of Its Strengths and Weaknesses for Solving Undergraduate Computer Science Questions. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*. 625–631.

[55] Jasem Khelifi, Moataz Chouchen, Ali Ouni, Dong Wang, Raula Gaikovina Kula, Salma Hamza, and Mohamed Wiem Mkaouer. 2024. GitRev: An LLM-based gamification framework for modern code review activities. In *2024 IEEE International Conference on Source Code Analysis and Manipulation (SCAM)* (Flagstaff, AZ, USA). IEEE, 235–241.

[56] Nam Wook Kim, Hyung-Kwon Ko, Grace Myers, and Benjamin Bach. 2024. ChatGPT in Data Visualization Education: A Student Perspective. In *2024 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. 109–120. doi:10.1109/VL/HCC60511.2024.00022

[57] Barbara Kitchenham. 2004. Procedures for performing systematic reviews. *Keele, UK, Keele University* 33, 2004 (2004), 1–26.

[58] Tomáš Kormaník, Michal Gabonai Gabonai, and Jaroslav Porubän. 2024. Using Machine Learning Concepts with ControlNet for Educational Advancements. In *2024 International Conference on Emerging eLearning Technologies and Applications (ICETA)*. 350–355. doi:10.1109/ICETA63795.2024.10850830

[59] Kai Korpimies, Antti Laaksonen, and Matti Luukkainen. 2024. Unrestricted Use of LLMs in a Software Project Course: Student Perceptions on Learning and Impact on Course Performance. In *Proceedings of the 24th Koli Calling International Conference on Computing Education Research (Koli Calling '24)*. Association for Computing Machinery, New York, NY, USA, Article 23, 7 pages. doi:10.1145/3699538.3699541

[60] Amruth N. Kumar, Rajendra K. Raj, Sherif G. Aly, Monica D. Anderson, Brett A. Becker, Richard L. Blumenthal, Eric Eaton, Susan L. Epstein, Michael Goldweber, Pankaj Jalote, Douglas Lea, Michael Oudshoorn, Marcelo Pias, Susan Reiser, Christian Servin, Rahul Simha, Titus Winters, and Qiao Xiang. 2024. *Computer Science Curricula 2023*. Association for Computing Machinery, New York, NY, USA.

[61] Harsh Kumar, Ilya Musabirov, Mohi Reza, Jiakai Shi, Xinyuan Wang, Joseph Jay Williams, Anastasia Kuzminykh, and Michael Liut. 2024. Guiding Students in Using LLMs in Supported Learning Environments: Effects on Interaction Dynamics, Learner Performance, Confidence, and Trust. *Proceedings of the ACM on Human-Computer Interaction* 8, CSCW2 (2024), 1–30. doi:10.1145/3687038

[62] Harsh Kumar, Ruiwei Xiao, Benjamin Lawson, Ilya Musabirov, Jiakai Shi, Xinyuan Wang, Huayin Luo, Joseph Jay Williams, Anna N. Rafferty, John Stamper, and Michael Liut. 2024. Supporting Self-Reflection at Scale with Large Language Models: Insights from Randomized Field Experiments in Classrooms. In *Proceedings of the Eleventh ACM Conference on Learning @ Scale* (Atlanta, GA, USA) *(L@S '24)*. Association for Computing Machinery, New York, NY, USA, 86–97. doi:10.1145/3657604.3662042

[63] Malgorzata Lagisz, Yefeng Yang, Sarah Young, and Shinichi Nakagawa. 2025. A practical guide to evaluating sensitivity of literature search strings for systematic reviews using relative recall. *Research Synthesis Methods* 16, 1 (2025), 1–14. doi:10.1017/rsm.2024.6

[64] Paula Lauren. 2024. Work-in-Progress: Course-based Undergraduate Research Experience (CURE) with Generative AI in a Computer Science Course. In *2024 IEEE Frontiers in Education Conference (FIE)*. 1–5. doi:10.1109/FIE61694.2024.10893139

[65] Paula Lauren and Paul Watta. 2023. Work-in-Progress: Integrating Generative AI with Evidence-based Learning Strategies in Computer Science and Engineering Education. In *2023 IEEE Frontiers in Education Conference (FIE)*. 1–5. doi:10.1109/FIE58773.2023.10342970

[66] H. Li and X. Wu. 2025. The use of generative AI tools in academic writing: a systematic review of research trends and thematic insights. *AI Ethics* (2025). doi:10.1007/s43681-025-00827-0

[67] Jingyue Li, Per Håkon Meland, Jakob Svennevik Notland, André Storhaug, and Jostein Hjortland Tysse. 2023. Evaluating the Impact of ChatGPT on Exercises of a Software Security Course. In *2023 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. 1–6. doi:10.1109/ESEM56168.2023.10304857

[68] Jian Liao, Linrong Zhong, Longting Zhe, Handan Xu, Ming Liu, and Tao Xie. 2024. Scaffolding Computational Thinking With ChatGPT. *IEEE Transactions on Learning Technologies* 17 (2024), 1628–1642. doi:10.1109/TLT.2024.3392896

[69] Jonathan Liu, Seth Poulsen, Erica Goodwin, Hongxuan Chen, Grace Williams, Yael Gertner, and Diana Franklin. 2025. Teaching Algorithm Design: A Literature Review. *ACM Trans. Comput. Educ.* 25, 2, Article 17 (May 2025), 20 pages. doi:10.1145/3727987

[70] Yunkai Liu. 2023. Leveraging the Power of AI in Undergraduate Computer Science Education: Opportunities and Challenges. In *2023 IEEE Frontiers in Education Conference (FIE)*. 1–5. doi:10.1109/FIE58773.2023.10343474

[71] Yi Liu. 2024. Integrating conversational large language models into student learning: A case study of ChatGPT in software engineering education. In *2024 IEEE Frontiers in Education Conference (FIE)* (Washington, DC, USA). IEEE, 1–8.

[72] Michael Luu, Matthew Ferland, Varun Nagaraj Rao, Arushi Arora, Randy Huynh, Frederick Reiber, Jennifer Wong-Ma, and Michael Shindler. 2023. What is an Algorithms Course? Survey Results of Introductory Undergraduate Algorithms Courses in the U.S.. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1* (Toronto ON, Canada) *(SIGCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 284–290. doi:10.1145/3545945.3569820

[73] Andrew Luxton-Reilly, Simon, Ibrahim Albluwi, Brett A. Becker, Michail Giannakos, Amruth N. Kumar, Linda Ott, James Paterson, Michael James Scott, Judy Sheard, and Claudia Szabo. 2018. Introductory programming: a systematic literature review. In *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education* (Larnaca, Cyprus) *(ITiCSE 2018 Companion)*. Association for Computing Machinery, New York, NY, USA, 55–106. doi:10.1145/3293881.3295779

[74] Stephen MacNeil, Andrew Tran, Arto Hellas, Joanne Kim, Sami Sarsa, Paul Denny, Seth Bernstein, and Juho Leinonen. 2023. Experiences from Using Code Explanations Generated by Large Language Models in a Web Software Development E-Book. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1* (Toronto ON, Canada) *(SIGCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 931–937. doi:10.1145/3545945.3569785

[75] Eric D. Manley, Timothy Urness, Andrei Migunov, and Md. Alimoor Reza. 2024. Examining Student Use of AI in CS1 and CS2. *J. Comput. Sci. Coll.* 39, 6 (April 2024), 41–51.

[76] Alireza Marefat, Abbaas Alif Mohamed Nishar, and Ashwin Ashok. 2025. Text2Net: Transforming Plain-text to a Dynamic Interactive Network Simulation Environment. In *SoutheastCon 2025*. 625–630. doi:10.1109/SoutheastCon56624.2025.10971486

[77] John McCarthy, Marvin L Minsky, Nathaniel Rochester, and Claude E Shannon. 2006. A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. *AI magazine* 27, 4 (2006), 12–12.

[78] Nabor C Mendonça. 2024. Evaluating ChatGPT-4 Vision on Brazil's National Undergraduate Computer Science Exam. *ACM Transactions on Computing Education* (2024).

[79] André Menolli, Bruno Strik, and Luiz Rodrigues. 2024. Teaching Refactoring to Improve Code Quality with ChatGPT: An Experience Report in Undergraduate Lessons. In *Proceedings of the XXIII Brazilian Symposium on Software Quality (SBQS '24)*. Association for Computing Machinery, New York, NY, USA, 563–574. doi:10.1145/3701625.3701681

[80] Simone Mezzaro, Alessio Gambi, and Gordon Fraser. 2024. An Empirical Study on How Large Language Models Impact Software Testing Learning. In *Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering* (Salerno, Italy) *(EASE '24)*. Association for Computing Machinery, New York, NY, USA, 555–564. doi:10.1145/3661167.3661273

[81] Diego Miranda, Dayana Palma, Adrian Fernández, René Noel, Cristian Cechinel, and Roberto Munoz. 2024. Enhancing Agile Project Management Education with AI: ChatGPT-4's Role in Evaluating Student Contributions. In *2024 43rd International Conference of the Chilean Computer Science Society (SCCC)*. 1–4. doi:10.1109/SCCC63879.2024.10767612

[82] Khalid Nafil and Youssef Lefdaoui. 2024. Innovative Approach to Agile Education: Generative AI-Supported Planning Poker Simulation. In *2024 21st International Conference on Information Technology Based Higher Education and Training (ITHET)*. 1–6. doi:10.1109/ITHET61869.2024.10837671

[83] Connor Nelson, Adam Doupé, and Yan Shoshitaishvili. 2025. SENSAI: Large Language Models as Applied Cybersecurity Tutors. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1* (Pittsburgh, PA, USA) *(SIGCSETS 2025)*. Association for Computing Machinery, New York, NY, USA, 833–839. doi:10.1145/3641554.3701801

[84] Alexander Tobias Neumann, Yue Yin, Sulayman Sowe, Stefan Decker, and Matthias Jarke. 2025. An LLM-Driven Chatbot in Higher Education for Databases and Information Systems. *IEEE Transactions on Education* 68, 1 (Feb 2025), 103–116. doi:10.1109/TE.2024.3467912

[85] Andrés Neyem, Luis A. González, Marcelo Mendoza, Juan Pablo Sandoval Alcocer, Leonardo Centellas, and Carlos Paredes. 2024. Toward an AI Knowledge Assistant for Context-Aware Learning Experiences in Software Capstone Project Development. *IEEE Transactions on Learning Technologies* 17 (2024), 1599–1614. doi:10.1109/TLT.2024.3396735

[86] Andres Neyem, Juan Pablo Sandoval Alcocer, Marcelo Mendoza, Leonardo Centellas-Claros, Luis A. Gonzalez, and Carlos Paredes-Robles. 2024. Exploring the Impact of Generative AI for StandUp Report Recommendations in Software Capstone Project Development. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1* (Portland, OR, USA) *(SIGCSE 2024)*. Association for Computing Machinery, New York, NY, USA, 951–957. doi:10.1145/3626252.3630854

[87] Ha Nguyen, Nate Stott, and Vicki Allan. 2024. Comparing Feedback from Large Language Models and Instructors: Teaching Computer Science at Scale. In *Proceedings of the Eleventh ACM Conference on Learning @ Scale* (Atlanta, GA, USA) *(L@S '24)*. Association for Computing Machinery, New York, NY, USA, 335–339. doi:10.1145/3657604.3664660

[88] Ana Carolina Oran, Letícia Braga Montenegro, Hellmut Alencar Schuster, José Carlos Duarte, Williamson Silva, and Rayfran Rocha Lima. 2024. Integrating ChatGPT in Project Management Education: Benefits and Challenges in the Academic Environment. In *Proceedings of the XXIII Brazilian Symposium on Software Quality (SBQS '24)*. Association for Computing Machinery, New York, NY, USA, 596–604. doi:10.1145/3701625.3701684

[89] Sonia Pamplona, Nelson Medinilla, and Pamela Flores. 2018. A Systematic Map for Improving Teaching and Learning in Undergraduate Operating Systems Courses. *IEEE Access* 6 (2018), 60974–60992. doi:10.1109/ACCESS.2018.2871768

[90] Zacharoula Papamitsiou, Michail Giannakos, Simon, and Andrew Luxton-Reilly. 2020. Computing Education Research Landscape through an Analysis of Keywords. In *Proceedings of the 2020 ACM Conference on International Computing Education Research* (Virtual Event, New Zealand) *(ICER '20)*. Association for Computing Machinery, New York, NY, USA, 102–112. doi:10.1145/3372782.3406276

[91] Seth Poulsen, Sami Sarsa, James Prather, Juho Leinonen, Brett A. Becker, Arto Hellas, Paul Denny, and Brent N. Reeves. 2024. Solving Proof Block Problems Using Large Language Models. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1* (Portland, OR, USA) *(SIGCSE 2024)*. Association for Computing Machinery, New York, NY, USA, 1063–1069. doi:10.1145/3626252.3630928

[92] Kishore Prakash, Shashwat Rao, Rayan Hamza, Jack Lukich, Vatsal Chaudhari, and Arnab Nandi. 2024. Integrating LLMs into Database Systems Education. In *Proceedings of the 3rd International Workshop on Data Systems Education: Bridging Education Practice with Education Research* (Santiago, AA, Chile) *(DataEd '24)*. Association for Computing Machinery, New York, NY, USA, 33–39. doi:10.1145/3663649.3664371

[93] Siddhartha Prasad, Ben Greenman, Tim Nelson, and Shriram Krishnamurthi. 2023. Generating Programs Trivially: Student Use of Large Language Models. In *Proceedings of the ACM Conference on Global Computing Education Vol 1* (Hyderabad, India) *(CompEd 2023)*. Association for Computing Machinery, New York, NY, USA, 126–132. doi:10.1145/3576882.3617921

[94] James Prather, Paul Denny, Juho Leinonen, Brett A. Becker, Ibrahim Albluwi, Michelle Craig, Hieke Keuning, Natalie Kiesler, Tobias Kohn, Andrew Luxton-Reilly, Stephen MacNeil, Andrew Petersen, Raymond Pettit, Brent N. Reeves, and Jaromir Savelka. 2023. The Robots Are Here: Navigating the Generative AI Revolution in Computing Education. In *Proceedings of the 2023 Working Group Reports on Innovation and Technology in Computer Science Education* (Turku, Finland) *(ITiCSE-WGR '23)*. Association for Computing Machinery, New York,

NY, USA, 108–159. doi:10.1145/3623762.3633499

[95] James Prather, Juho Leinonen, Natalie Kiesler, Jamie Gorson Benario, Sam Lau, Stephen MacNeil, Narges Norouzi, Simone Opel, Vee Pettit, Leo Porter, Brent N. Reeves, Jaromir Savelka, David H. Smith, Sven Strickroth, and Daniel Zingaro. 2025. Beyond the Hype: A Comprehensive Review of Current Trends in Generative AI Research, Teaching Practices, and Tools. In *2024 Working Group Reports on Innovation and Technology in Computer Science Education* (Milan, Italy) *(ITiCSE 2024)*. Association for Computing Machinery, New York, NY, USA, 300–338. doi:10.1145/3689187.3709614

[96] Cong Pu. 2024. Integrating Generative AI with Data Structures and Algorithm Analysis Course Homework. In *2024 IEEE Frontiers in Education Conference (FIE)*. 1–9. doi:10.1109/FIE61694.2024.10892822

[97] Ashwin Rachha and Mohammed Seyam. 2024. LLM-Enhanced Learning Environments for CS: Exploring Data Structures and Algorithms with Gurukul. In *2024 IEEE Frontiers in Education Conference (FIE)*. 1–9. doi:10.1109/FIE61694.2024.10893211

[98] Nishat Raihan, Mohammed Latif Siddiq, Joanna C.S. Santos, and Marcos Zampieri. 2025. Large Language Models in Computer Science Education: A Systematic Literature Review. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1* (Pittsburgh, PA, USA) *(SIGCSETS 2025)*. Association for Computing Machinery, New York, NY, USA, 938–944. doi:10.1145/3641554.3701863

[99] Vijayalakshmi Ramasamy, Eli Kulpinski, Thomas Beaupre, Aaron Antreassian, Yunhwan Jeong, Peter J. Clarke, Anthony Aiello, and Charles Ray. 2024. Enhancing CS Education with LAs Using AI-Empowered AIELA Program. In *2024 IEEE Frontiers in Education Conference (FIE)*. 1–9. doi:10.1109/FIE61694.2024.10893294

[100] Vijayalakshmi Ramasamy, Suganya Ramamoorthy, Gursimran Singh Walia, Eli Kulpinski, and Aaron Antreassian. 2024. Enhancing User Story Generation in Agile Software Development Through Open AI and Prompt Engineering. In *2024 IEEE Frontiers in Education Conference (FIE)*. 1–8. doi:10.1109/FIE61694.2024.10893343

[101] Valeria Ramirez Osorio, Angela Zavaleta Bernuy, Bogdan Simion, and Michael Liut. 2025. Understanding the Impact of Using Generative AI Tools in a Database Course. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1* (Pittsburgh, PA, USA) *(SIGCSETS 2025)*. Association for Computing Machinery, New York, NY, USA, 959–965. doi:10.1145/3641554.3701785

[102] Sanka Rasnayaka, Guanlin Wang, Ridwan Shariffdeen, and Ganesh Neelakanta Iyer. 2024. An Empirical Study on Usage and Perceptions of LLMs in a Software Engineering Project. In *Proceedings of the 1st International Workshop on Large Language Models for Code* (Lisbon, Portugal) *(LLM4Code '24)*. Association for Computing Machinery, New York, NY, USA, 111–118. doi:10.1145/3643795.3648379

[103] Sara Riazi and Pedram Rooshenas. 2025. LLM-Driven Feedback for Enhancing Conceptual Design Learning in Database Systems Courses. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1* (Pittsburgh, PA, USA) *(SIGCSETS 2025)*. Association for Computing Machinery, New York, NY, USA, 1001–1007. doi:10.1145/3641554.3701940

[104] Roberto Rodriguez-Echeverría, Juan D Gutiérrez, José M Conejero, and Álvaro E Prieto. 2024. Analysis of ChatGPT Performance in Computer Engineering Exams. *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje* (2024).

[105] Michael P Rogers, Hannah Miller Hillberg, and Christopher L Groves. 2024. Attitudes towards the use (and misuse) of ChatGPT: A preliminary study. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*. 1147–1153.

[106] Niklas Rönnberg and Ahmet Börütecene. 2024. Use of Generative AI for Fictional Field Studies in Design Courses. In *Adjunct Proceedings of the 2024 Nordic Conference on Human-Computer Interaction* (Uppsala, Sweden) *(NordiCHI '24 Adjunct)*. Association for Computing Machinery, New York, NY, USA, Article 23, 5 pages. doi:10.1145/3677045.3685439

[107] Chandan Kumar Sah, Lian Xiaoli, Muhammad Mirajul Islam, and Md Kamrul Islam. 2024. Navigating the AI Frontier: A Critical Literature Review on Integrating Artificial Intelligence into Software Engineering Education. In *2024 36th International Conference on Software Engineering Education and Training (CSEE&T)*. 1–5. doi:10.1109/CSEET62301.2024.10663054

[108] Mario Sánchez and Andrea Herrera. 2023. Assessing ChatGPT's Proficiency in CS1-Level Problem Solving. In *Colombian Conference on Computing*. Springer, 71–81.

[109] Patricia Santos, Keysha Urgel, and Verónica Moreno. 2024. Generative artificial intelligence in teaching and learning of ICT engineering education: A literature review and illustrative scenarios. In *2024 47th MIPRO ICT and Electronics Convention (MIPRO)* (Opatija, Croatia). IEEE.

[110] Orit Shaer and Angelora Cooper. 2024. Integrating Generative Artificial Intelligence to a Project-Based Tangible Interaction Course. *IEEE Pervasive Computing* 23, 1 (Jan 2024), 63–69. doi:10.1109/MPRV.2023.3346548

[111] Anshul Shah, Anya Chernova, Elena Tomson, Leo Porter, William G. Griswold, and Adalbert Gerald Soosai Raj. 2025. Students' Use of GitHub Copilot for Working with Large Code Bases. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1* (Pittsburgh, PA, USA) *(SIGCSETS 2025)*. Association for Computing Machinery, New York, NY, USA, 1050–1056. doi:10.1145/3641554.3701800

[112] Yiyin Shen, Xinyi Ai, Adalbert Gerald Soosai Raj, Rogers Jeffrey Leo John, and Meenakshi Syamkumar. 2024. Implications of ChatGPT for Data Science Education. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1* (Portland, OR, USA) *(SIGCSE 2024)*. Association for Computing Machinery, New York, NY, USA, 1230–1236. doi:10.1145/3626252.3630874

[113] Simon and Judy Sheard. 2020. Twenty-Four Years of ITiCSE Papers. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education* (Trondheim, Norway) *(ITiCSE '20)*. Association for Computing Machinery, New York, NY, USA, 5–11. doi:10.1145/3341525.3387407

[114] Wilson Soto. 2025. Tool-Based Retrieval-Augmented Generative as an Automated Assistant in Object-Oriented Programming Course. In *2025 IEEE Engineering Education World Conference (EDUNINE)*. 1–6. doi:10.1109/EDUNINE62377.2025.10981323

[115] Sandro Speth, Niklas Meißner, and Steffen Becker. 2023. Investigating the Use of AI-Generated Exercises for Beginner and Intermediate Programming Courses: A ChatGPT Case Study. In *2023 IEEE 35th International Conference on Software Engineering Education and Training (CSEE&T)*. 142–146. doi:10.1109/CSEET58097.2023.00030

[116] Jason Tham. 2024. Teaching UX: Amid the Hype of Generative AI. *Journal of User Experience* 20, 1 (2024), 1–8.

[117] Ayca Tuzmen. 2024. Use of generative artificial intelligence in the education of software verification and validation. In *2024 IEEE Frontiers in Education Conference (FIE)* (Washington, DC, USA). IEEE, 1–6.

[118] David W. Valentine. 2004. CS educational research: a meta-analysis of SIGCSE technical symposium proceedings. *SIGCSE Bull.* 36, 1 (March 2004), 255–259. doi:10.1145/1028174.971391

[119] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[120] Michael Vierhauser, Iris Groher, Tobias Antensteiner, and Clemens Sauerwein. 2024. Towards integrating emerging AI applications in SE education. In *2024 36th International Conference on Software Engineering Education and Training (CSEE&T)* (Würzburg, Germany). IEEE, 1–5.

[121] Beian Wang, Chong Wang, Peng Liang, Bing Li, and Cheng Zeng. 2024. How LLMs Aid in UML Modeling: An Exploratory Study with Novice Analysts. In *2024 IEEE International Conference on Software Services Engineering (SSE)*. 249–257. doi:10.1109/SSE62657.2024.00046

[122] Tianjia Wang, Ramaraja Ramanujan, Yi Lu, Chenyu Mao, Yan Chen, and Chris Brown. 2024. DevCoach: Supporting Students in Learning the Software Development Life Cycle at Scale with Generative Agents. In *Proceedings of the Eleventh ACM Conference on Learning @ Scale* (Atlanta, GA, USA) *(L@S '24)*. Association for Computing Machinery, New York, NY, USA, 351–355. doi:10.1145/3657604.3664663

[123] Yang Wang, Margaret McCoey, Qian Hu, and Maryam Jalalitabar. 2024. Teaching Security in the Era of Generative AI: A Course Design of Security + ChatGPT. In *2024 IEEE Integrated STEM Education Conference (ISEC)*. 01–06. doi:10.1109/ISEC61299.2024.10664905

[124] Yanan Wu, Xiaoping Zeng, and Qibei Lin. 2025. Generative AI Integrated Educational Model for User-Centered Design. In *Proceedings of the 2024 3rd International Conference on Artificial Intelligence and Education (ICAIE '24)*. Association for Computing Machinery, New York, NY, USA, 129–135. doi:10.1145/3722237.3722260

[125] Yuankai Xue, Hanlin Chen, Gina R Bai, Robert Tairas, and Yu Huang. 2024. Does ChatGPT Help With Introductory Programming? An Experiment of Students Using ChatGPT in CS1. In *Proceedings of the 46th International Conference on Software Engineering: Software Engineering Education and Training*. 331–341.

[126] He Zhang, Muhammad Ali Babar, and Paolo Tell. 2011. Identifying relevant studies in software engineering. *Information and Software Technology* 53, 6 (2011), 625–637. doi:10.1016/j.infsof.2010.12.010 Special Section: Best papers from the APSEC.

[127] Xishuo Zhang, Lin Liu, Yi Wang, Xiao Liu, Hailong Wang, Chetan Arora, Haichao Liu, Weijia Wang, and Thuong Hoang. 2024. Auto-Generated Personas: Enhancing User-centered Design Practices among University Students. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI EA '24)*. Association for Computing Machinery, New York, NY, USA, Article 52, 7 pages. doi:10.1145/3613905.3651043

[128] Yifan Zhang, Xinkui Zhao, Zuxin Wang, Zhengyi Zhou, Guanjie Cheng, Shuiguang Deng, and Jianwei Yin. 2025. SortingHat: Redefining Operating Systems Education with a Tailored Digital Teaching Assistant. In *Companion Proceedings of the ACM on Web Conference 2025* (Sydney NSW, Australia) *(WWW '25)*. Association for Computing Machinery, New York, NY, USA, 2951–2954. doi:10.1145/3701716.3715199

[129] Yong Zheng. 2023. ChatGPT for Teaching and Learning: An Experience from Data Science Education. In *Proceedings of the 24th Annual Conference on Information Technology Education* (Marietta, GA, USA) *(SIGITE '23)*. Association for Computing Machinery, New York, NY, USA, 66–72. doi:10.1145/3585059.3611431

A Validation Set of Papers

The following two sets of papers were used to help develop the search string and validate its performance in the systematic literature review (see Section 5).

WG Members Suggestions

The following is the set of papers used to validate the search string for the systematic literature review. It consists of papers suggested by WG members that should be included as upper-computing relevant.

10.1145/3641554.3701785 — Understanding the Impact of Using Generative AI Tools in a Database Course
10.1145/3649165.3690116 — Can ChatGPT pass a Theory of Computing Course?
10.1145/3641555.3705243 — Scaffolding Mock Conference Projects in Theory of Computing Courses
10.1145/3641554.3701946 — Investigating the Capabilities of Generative AI in Solving Data Structures, Algorithms, and Computability Problems
10.1145/3641554.3701823 — Analysis of Generative AI Policies in Computing Course Syllabi
10.1145/3633053.3633057 — Incorporating Generative AI into Software Development Education
10.1145/3639474.3640052 — LLMs Still Can't Avoid Instanceof: An Investigation Into GPT-3.5, GPT-4 and Bard's Capacity to Handle Object-Oriented Programming Assignments
10.1145/3641554.3701800 — Students' Use of GitHub Copilot for Working with Large Code Bases
10.1145/3663649.3664371 — Integrating LLMs into Database Systems Education
10.1145/3626252.3630927 — Software Engineering Education Must Adapt and Evolve for an LLM Environment
10.1145/3643795.3648379 — An Empirical Study on Usage and Perceptions of LLMs in a Software Engineering Project
10.1145/3626252.3630874 — Implications of ChatGPT for Data Science Education
10.1145/3576123.3576134 — My AI Wants to Know if This Will Be onthe Exam: Testing OpenAI's Codex on CS2 Programming Exercises
10.1145/3706599.3720291 — Utilizing ChatGPT in a Data Structures and Algorithms Course: A Teaching Assistant's Perspective
10.1109/CSEET62301.2024.10663055 — Leveraging Open Source LLMs for Software Engineering Education and Training
10.1109/TE.2024.3467912 — An LLM-Driven Chatbot in Higher Education for Databases and Information Systems
10.1109/FIE61694.2024.10893211 — LLM-Enhanced Learning Environments for CS: Exploring Data Structures and Algorithms with Gurukul

Raihan Papers from SIGCSE TS 2025

To extend the validation set further, we extracted the upper-computing relevant papers from [98].

10.1145/3716640.3716657 — Analyzing llm usage in an advanced computing class in india
10.1145/3587102.3588814 — Gpt-3 vs object oriented programming assignments: An experience report
10.1145/3580305.3599827 — From human days to machine seconds: Automatically answering and generating machine learning final exams.
10.1145/3636243.3636263 — More than meets the ai: Evaluating the performance of gpt-4 on computer graphics assessment questions
10.1145/3638067.3638100 — May we consult chatgpt in our human-computer interaction written exam? an experience report after a professor answered yes
10.1145/3585059.3611409 — Teaching it software fundamentals: Strategies and techniques for inclusion of large language models: Strategies and techniques for inclusion of large language models
10.1145/3613904.3642349 — Teach ai how to code: Using large language models as teachable agents for programming education
10.1145/3622780.3623648 — A seamless integration of chatgpt into jupyter environments for programming education
10.1145/3631802.3631830 — Codehelp: Using large language models with guardrails for scalable support in programming classes
10.1145/3545945.3569785 — Experiences from using code explanations generated by large language models in a web software development e-book
10.1145/3674149 — Evaluating chatgpt-4 vision on brazil's national undergraduate computer science exam
10.1145/3616961.3616974 — "call me kiran" – chatgpt as a tutoring chatbot in a computer science course
10.1145/3605507.3610629 — Dual-submission homework in parallel computer architecture: An exploratory study in the age of llms
10.1145/3585059.3611431 — Chatgpt for teaching and learning: An experience from data science education
10.1109/qrs60937.2023.00028 — On chatgpt: Perspectives from software engineering students
10.1109/vl/hcc60511.2024.00022 — Chatgpt in data visualization education: A student perspective
10.1109/esem56168.2023.10304857 — Evaluating the impact of chatgpt on exercises of a software security course
10.1109/csce60160.2023.00171 — Analysis of chatgpt performance in computer engineering exams
10.1109/icstw58534.2023.00078 — Chatgpt and software testing education: Promises & perils

B Lower Computing Literature Set

Someone, somewhere might find this set of papers ($n = 218$) useful to kickstart a future literature review of papers we judged as belonging to lower computing…

(1) 10.1109/FIE58773.2023.10343457
(2) 10.1145/3657604.3662032
(3) 10.1145/3636243.3636249
(4) 10.5555/3715622.3715630
(5) 10.1109/LA-CCI62337.2024.10814750
(6) 10.1145/3639474.3640058
(7) 10.1109/JICV59748.2023.10565716
(8) 10.1109/EDUCON60312.2024.10578746
(9) 10.1145/3649409.3691090
(10) 10.1145/3581754.3584111
(11) 10.1145/3641555.3705066
(12) 10.1145/3649217.3653554
(13) 10.1145/3643795.3648389
(14) 10.1145/3657604.3662036
(15) 10.1145/3627217.3627235
(16) 10.1145/3641555.3705282
(17) 10.1145/3617367
(18) 10.1145/3631802.3631848
(19) 10.1145/3626253.3633433
(20) 10.1145/3663384.3663393
(21) 10.1145/3657604.3662039
(22) 10.5555/3737313.3737334
(23) 10.1145/3641554.3701806
(24) 10.1145/3649405.3659504
(25) 10.1145/3501385.3543957
(26) 10.1145/3636243.3636259
(27) 10.1145/3699538.3699567
(28) 10.1145/3716640.3716656
(29) 10.1145/3649409.3691093
(30) 10.1109/TE.2024.3394060
(31) 10.1145/3626252.3630863
(32) 10.1145/3490100.3516473
(33) 10.1145/3587102.3588794
(34) 10.1109/EDUCON62633.2025.11016417
(35) 10.1109/ACCESS.2024.3443621
(36) 10.1145/3649165.3690123
(37) 10.1145/3605468.3609775
(38) 10.1145/3641554.3701906
(39) 10.1109/WSC63780.2024.10838799
(40) 10.1145/3626252.3630761
(41) 10.1109/ACCESS.2024.3380909
(42) 10.1109/ISAS60782.2023.10391549
(43) 10.1145/3626253.3635483
(44) 10.1145/3716640.3716651
(45) 10.1145/3701268.3701274
(46) 10.1145/3626252.3630789
(47) 10.1145/3613904.3642773
(48) 10.1145/3649165.3690113
(49) 10.5555/3715602.3715619
(50) 10.1109/MIPRO60963.2024.10569736
(51) 10.1145/3641554.3701910
(52) 10.1145/3716640.3716649
(53) 10.1109/EDUCON62633.2025.11016313
(54) 10.1145/3649217.3653568
(55) 10.5555/3665464.3665476
(56) 10.1145/3627217.3627233
(57) 10.1145/3657604.3664673
(58) 10.1145/3649850
(59) 10.1145/3649217.3653600
(60) 10.1145/3617650.3624946
(61) 10.1145/3545945.3569823
(62) 10.1145/3723178.3723268
(63) 10.1145/3649217.3653558
(64) 10.1109/EDUCON62633.2025.11016505
(65) 10.1145/3649409.3691076
(66) 10.1145/3631802.3631807
(67) 10.1109/FIE58773.2023.10343037
(68) 10.1145/3632621.3671429
(69) 10.1145/3641554.3701974
(70) 10.1145/3641555.3705051
(71) 10.1145/3626253.3635592
(72) 10.1109/EDUCON60312.2024.10578718
(73) 10.1145/3641554.3701945
(74) 10.1145/3626252.3630773
(75) 10.1145/3632620.3671116
(76) 10.1109/FIE61694.2024.10893099
(77) 10.1145/3716640.3716652
(78) 10.1109/ISEC61299.2024.10665141
(79) 10.1145/3641554.3701883
(80) 10.1109/UEMCON62879.2024.10754661
(81) 10.1145/3636555.3636882
(82) 10.1109/URTC65039.2024.10937513
(83) 10.1109/REEPE60449.2024.10479881
(84) 10.1145/3649217.3653621
(85) 10.1145/3737885
(86) 10.1145/3641555.3705195
(87) 10.1109/FIE58773.2023.10343247
(88) 10.1145/3641554.3701934
(89) 10.1109/ACCESS.2024.3445432
(90) 10.1145/3657604.3664701
(91) 10.5555/3715638.3715656
(92) 10.1145/3706599.3720203
(93) 10.1145/3631802.3631830
(94) 10.1145/3626252.3630817
(95) 10.5555/3722479.3722526
(96) 10.1145/3511861.3511863
(97) 10.1145/3626253.3635519
(98) 10.1145/3649217.3653543
(99) 10.1145/3568813.3600139
(100) 10.1145/3626253.3635542
(101) 10.1145/3649217.3653574
(102) 10.1109/FIE61694.2024.10892816
(103) 10.1109/ICEED62316.2024.10923780
(104) 10.1145/3649217.3653587
(105) 10.1145/3639474.3640061
(106) 10.1145/3639474.3640076
(107) 10.1145/3616961.3616974
(108) 10.5555/3665609.3665618
(109) 10.1145/3649409.3691073
(110) 10.1145/3641554.3701864
(111) 10.1145/3699538.3699556
(112) 10.1145/3641555.3705227
(113) 10.1145/3641554.3701791
(114) 10.1145/3626252.3630764
(115) 10.1145/3641554.3701853
(116) 10.1109/ICEED62316.2024.10923833
(117) 10.1145/3544548.3580919
(118) 10.1145/3613905.3637148
(119) 10.1145/3622780.3623648
(120) 10.1145/3657604.3664665
(121) 10.1145/3649217.3653596
(122) 10.1145/3649217.3653608
(123) 10.1145/3626253.3635427
(124) 10.1145/3613904.3642706
(125) 10.1109/FIE61694.2024.10893137
(126) 10.1145/3706468.3706529
(127) 10.1145/3641554.3701810
(128) 10.1145/3701716.3717810
(129) 10.1145/3636243.3636252
(130) 10.1145/3641554.3701959
(131) 10.1145/3641554.3701827
(132) 10.1109/EDUCON60312.2024.10578921
(133) 10.1145/3640544.3645234
(134) 10.1145/3576882.3617916
(135) 10.1145/3631700.3665227
(136) 10.1145/3624032.3624035
(137) 10.1109/ITHET61869.2024.10837624
(138) 10.5555/3715638.3715650
(139) 10.1145/3641555.3704765
(140) 10.1109/HNICEM60674.2023.10589162
(141) 10.1109/QRS62785.2024.00027
(142) 10.1145/3641554.3701913
(143) 10.1145/3641555.3705253
(144) 10.1145/3649217.3653594
(145) 10.1145/3649405.3659517
(146) 10.1109/FLLM63129.2024.10852455
(147) 10.1109/ISEC61299.2024.10665172
(148) 10.1145/3587102.3588852
(149) 10.1145/3699538.3699580
(150) 10.1145/3701268.3701273
(151) 10.1109/DEMOcon63027.2024.10748200
(152) 10.1109/TALE62452.2024.10834365
(153) 10.1145/3706598.3713832
(154) 10.1145/3649165.3690125
(155) 10.1145/3634814.3634816
(156) 10.1145/3649217.3653584
(157) 10.1145/3558489.3559072
(158) 10.1109/SEEDA-CECNSM63478.2024.00031
(159) 10.1145/3613905.3650937
(160) 10.1145/3649217.3653615
(161) 10.1145/3643795.3648380
(162) 10.1145/3649217.3653626
(163) 10.1145/3593663.3593692
(164) 10.1109/CSCE60160.2023.00172
(165) 10.1145/3716640.3716647
(166) 10.1145/3587102.3588792
(167) 10.1145/3686852.3687073
(168) 10.1145/3613904.3642229
(169) 10.1109/FIE61694.2024.10893442
(170) 10.1145/3641554.3701972
(171) 10.1109/FIE61694.2024.10892956

(172) 10.1109/TALE56641.2023.10398297
(173) 10.1145/3626252.3630826
(174) 10.1145/3545945.3569830
(175) 10.1145/3657604.3662040
(176) 10.1145/3632620.3671092
(177) 10.1145/3613905.3647967
(178) 10.1145/3649217.3653533
(179) 10.1145/3626252.3630897
(180) 10.1145/3626252.3630799
(181) 10.1145/3626252.3630960
(182) 10.1145/3626253.3635600
(183) 10.1145/3641555.3705166
(184) 10.1145/3639474.3640076
(185) 10.5555/3575618.3575622
(186) 10.1145/3706598.3713644
(187) 10.1145/3636243.3636245
(188) 10.1145/3626252.3630803
(189) 10.1145/3632620.3671103
(190) 10.1145/3626252.3630937
(191) 10.1109/SCSE61872.2024.10550837
(192) 10.1145/3641555.3705061
(193) 10.1145/3641555.3705201
(194) 10.1145/3626253.3635543
(195) 10.1109/DSAA61799.2024.10722841
(196) 10.1145/3631802.3631806
(197) 10.1109/FIE61694.2024.10892934
(198) 10.1145/3699538.3699591
(199) 10.1145/3632620.3671098
(200) 10.1145/3649165.3690094
(201) 10.1145/3649217.3653595
(202) 10.1145/3587102.3588785
(203) 10.1145/3643795.3648380
(204) 10.1145/3640794.3665542
(205) 10.1145/3568812.3603476
(206) 10.1145/3723010.3723034
(207) 10.1145/3626253.3635511
(208) 10.1145/3626252.3630938
(209) 10.1145/3636243.3636247
(210) 10.1145/3706468.3706500
(211) 10.1145/3626252.3630822
(212) 10.1109/ICONAT61936.2024.10774681
(213) 10.1109/EDUCON62633.2025.11016298
(214) 10.1109/TLT.2024.3499751
(215) 10.1145/3704289.3704301
(216) 10.1145/3689535.3689554
(217) 10.1145/3626253.3635404
(218) 10.1145/3626252.3630909

C Educator Survey

Table 31: Survey Questions for Instructors Using Generative AI in Upper-Level Computing Courses

Topic	ID	Question	Options
Background Information on upper-level courses	1	What upper-level computing courses (i.e., beyond CS1 / Introduction to Programming / similar) do you currently teach?	• AI / ML • Algorithms • Compilers • Computer Networks • Databases • OOP • Operating Systems • Software Engineering • Theory of Computation • HCI • Web Development • Mobile Computing • *Other*
	2	Have you integrated Generative AI tools in at least one advanced or upper-level computing course?	• Yes • No
Questions for Instructors that have not integrated GenAI into upper-level courses			
Reasons not using GenAI and possible ways to integrate it	3	Please explain the reasons why you have not yet started to adopt Generative AI (e.g. lack of institutional support, lack of knowledge, etc.)	N/A, an open-ended question
	4	Which of the following ways do you plan to integrate Generative AI tools (e.g., ChatGPT, GitHub Copilot) into your upper-level computing courses? (Select all that apply)	• Brainstorming (e.g., generate project or functionality ideas, discuss topics, etc.) • Requirements clarification (e.g., use the GenAI to reduce ambiguity in requirements, or further break requirements down, etc.) • Ask students to critically evaluate GenAI's replies (e.g., evaluate generated code) • Integrate GenAI into course projects (e.g., pair programming with GenAI, GenAI-specific requirements, etc.) • Demonstrate GenAI tools during lectures (e.g., live walkthroughs or examples) • Allow/encourage students to use responsibly GenAI during assignments (without dedicated tasks around it) • Involve students in studying or analyzing GenAI itself (e.g., model behavior, ethics, biases) • *I do not plan to integrate GenAI in my courses* • *Other*

Continued on next page

Table 31 – *Continued from previous page*

Topic	ID	Question	Options
Questions for Instructors that have integrated GenAI into upper-level courses			
General strategies to use GenAI	5	Which of the following ways have you integrated Generative AI tools (e.g., ChatGPT, GitHub Copilot) into your upper-level computing courses? (Select all that apply)	• Brainstorming (e.g., generate project or functionality ideas, discuss topics, etc.) • Requirements clarification (e.g., use the GenAI to reduce ambiguity in requirements, or further break requirements down, etc.) • Ask students to critically evaluate GenAI's replies (e.g., evaluate generated code) • Integrated GenAI into course projects (e.g., pair programming with GenAI, GenAI-specific requirements, etc.) • Demonstrated GenAI tools during lectures (e.g., live walkthroughs or examples) • Allowed/encouraged students to use GenAI during assignments (without dedicated tasks around it) • Involved students in studying or analyzing GenAI itself (e.g., model behavior, ethics, biases) • *Other*
	6	For the activities you selected in the previous question, which ones worked well? Are you planning to continue using them, and are there any changes you're planning to make?	N/A, an open-ended question
	7	For the activities you selected in the previous question, which ones did not work well? Are you planning to continue using them or discontinue them, and are there any changes you're planning to make?	N/A, an open-ended question
	8	Can you provide us with some concrete examples of GenAI-based activities that you have experimented with? Describe: the course(s), the task(s), the assessment mechanism(s), and any lessons learned. If you provide more than one example, please separate them using "****".	N/A, an open-ended question
	9	Which of the following skills do you explicitly teach or reference in your course? (Select all that apply.)	(1) General prompting skills (e.g., rephrasing, refining, reducing ambiguity) (2) Prompt engineering techniques (e.g., few-shot, chain-of-thought prompting, etc.) (3) Instruction on using GenAI tools to decompose problems (4) Instruction on using GenAI tools to generate artifacts (e.g., code, SQL, user-interface prototypes, …) (5) Instruction on using GenAI tools to test solutions (e.g., code, SQL, etc.) (6) Instruction on using GenAI tools to analyse and improve solutions (e.g., debug, etc) (7) Instruction on using GenAI tools to understand/explain artifacts (e.g., code, SQL, etc.) (8) *Other* (9) *None of the above*

Continued on next page

Table 31 – *Continued from previous page*

Topic	ID	Question	Options
Learning Objectives Revision	10	Have you revised the learning objectives (e.g., Intended Learning Outcomes, Course Objectives, Course Outcomes) of any upper-level computing course in response to students using GenAI tools in the course?	(1) Yes (2) Not yet, but I'm planning to (3) No (4) Not applicable
	11	In which course(s), and how were the learning objectives revised (or how do you plan to revise them)? Please include the course name(s) and describe the changes made or planned to the learning objectives. You may also reflect on how GenAI influenced these revisions.	N/A, an open-ended question
Assessment and Evaluation Practices	12	What level of transparency do you expect from students when using GenAI tools? (Select all that apply.)	(1) Full log of interactions or transcript (2) Summary of usage (e.g., prompts that have been used) (3) Declaration of any use (4) Attribution of GenAI-generated content in submissions (5) Reflection on how GenAI influenced their process or results (6) No requirements (7) *Other*
	13	How have you changed (or plan to change) your approach to assessments in higher-level courses that incorporate GenAI?	(1) No Change (2) Give more points to the process or development journey (3) Give more attention to validation of results (e.g., evidence of testing) (4) Give more attention to student reflection (5) Add an oral explanation of the deliverables (6) Ask for in-person feature changes (e.g., add a new feature to the delivered code) (7) Design tasks that are harder for GenAI to solve (8) Emphasize responsible GenAI usage over GenAI-proofing (9) Use version control to monitor progress (10) *Other*
GenAI Policy and Resources	14	What is your current policy regarding the use of GenAI tools by students in your course(s)? (Please describe where GenAI use is allowed, restricted, or explicitly forbidden. You may include examples for homework, studying, projects, exams, or other relevant situations. If your policy varies by course or is not formally defined, feel free to elaborate.)	N/A, an open-ended question
	15	Have you used or created any publicly available resources that can be helpful when incorporating GenAI into higher-level courses? If yes, please specify the URLs for these resources and explain their impact.	N/A, an open-ended question