Table 1 Grouping of author keywords

Keyword group	Author keywords
AI, ML, LLM	AI; gen AI; explainable AI; AI techniques; machine learning; feature extraction; NLP; neural networks; deep learning; LLM; LLM-assisted; language models; transformers; zero-shot learning; few-shot learning; fine-tuning; code-fluent large language models; foundation models; generative models; generative pre-training' pre-trained model; predictive model; recommender systems
AI code assistance	AI for code; clean code; code documentation; code efficiency optimization; code linting; code snippets; code translation; creative code; program generation; program intention inference; program synthesis; readability; refactoring; programming; programming concepts; programming misconceptions; code comprehension; code explanation; automated code review; code review; (automated) software engineering; automated test generation; automated code generation; code completion; code generation; code recommender systems; code writing; coding generation; completion engine; software engineering tooling
GPT*	GPT, GPT-3, GPT-4, ChatGPT
Education and pedagogy	AI in education; CS education; K12 CS education; computer network education; curriculum; education; educational design; educational programs; educational technology; higher education; pedagogy; post-secondary; programming education; software education; teaching; engineering education; active learning; intelligent tutoring; interactive learning; learning assistance system; instructor perspectives; student engagement; assessment; academic integrity; open-ended questions; oral examinations; programming knowledge assessment
Introductory & intermediate programming	Java; JavaScript; Python; XML; programming language; intermediate programming; object-oriented programming; data structures; algorithms; CS1; CS2; introductory programming; novice programming
HCI studies	HCI; surveys; usability; user; user-centered design; user study; wizard of Oz; classroom experience; conversational interaction; design; digital interaction; gui design; practitioners' survey; scenario-based design;software developer survey; task-oriented interactions
Software testing	API testing; automated API testing; software testing; test automation; test generation; test migration; testing; testing tools; coverage testing; failure-inducing test cases; mobile app testing; bug detection and repair; computer bugs; program repair; repairing; automated program repair; compiler error messages; syntax errors; programming error messages
GitHub CoPilot	GitHub CoPilot
AI-aided instruction	AI- generated exercises; exercise generation; automatic question generation; automated grading; automated feedback; feedback; question evalution; question generation; resource generation; programming exercises; programming assingments
Prompt engineering	prompt augmentation; prompt engineering; chain-of-thought prompting
Software requirements	software requirements; software requirements classification; software requirements elicitation; requirements engineering (RE); RE education; user stories; wide audience; creative requirements elicitation; elicitation interview script generation; functional requirements

Software & its engineering	software; software documentation; software architecture; software context; software evolution; software projects; software performance; software development; software development management; software development process; software development productivity; software process improvement; pair programming; Scrum; agile; retrospective; empirical evaluation; empirical/experimental software engineering
BERT & others	BERT; Bard; alphacode; OpenAI Codex
Human aspects and ethics	Ethical AI; privacy; trustworthy AI; diversity; human aspects; human-centered AI; human values
Modeling	domain modeling; UML; model-driven engineering; modelling; software models
Formal methods	Formal methods; validation
Socio-technical aspects	developer experience; community smells; socio technical analysis; social implications of technology; programming profession; cohort building; collaboration; collaborative learning; computer supported collaborative work; onboarding; newcomer

Table 2 Distribution of papers from SE_I category across the six phases of SDLC

Requirements Engineering	
Ethics	Ronanki (2023)
Goal-Oriented Modelling	Chen et al. (2023a)
Requirements Analysis And Classification	Rajender Kumar Surana et al. (2019); Jain et al. (2023); Arulmohan et al. (2023); Das et al. (2023)
Requirements Elicitation	Rietz (2019); Rietz and Maedche (2019); Harntanto et al. (2022); Ronanki et al. (2023)
User Stories	Marczak-Czajka and Cleland-Huang (2023)
Software Development	
Api Recommendations	Vazquez et al. (2023)
Code Completion	Sun et al. (2023); Li et al. (2023c)
Code Transpilation	Zhang et al. (2023)
Code Understanding	Sharma et al. (2017); Bradley et al. (2018); Lin et al. (2020); Frazier et al. (2022)
Development Tools	Li and Wang (2023); Treude (2023); Acher et al. (2023); Phokela et al. (2023); Cabra-Acela et al. (2023); Vaithilingam et al. (2023)
Digital Coworker	Sharma et al. (2019); Kuttal et al. (2021); De Vito et al. (2023a); Robe and Kuttal (2022)
Digital Mentor	Dominic et al. (2020a,b)
Human Aspects	Ghorbani et al. (2023); Zamfirescu-Pereira et al. (2023); **?**
Other	Storey and Zagalsky (2016); Melo et al. (2023)
Pair Programming	Imai (2022)
Program Repair	Wood et al. (2018)
Program Synthesis	Barke et al. (2023); Moroz et al. (2022); Nguyen and Nadi (2022); Al Madi (2022); Vaithilingam et al. (2022); Bird et al. (2022); Yan et al. (2023); Purwoko et al. (2023); Moradi Dakhel et al. (2023); Feng et al. (2023); Jesse et al. (2023); Wang et al. (2023); Siddiq et al. (2023); Tang et al. (2023); Scoccia (2023); Ciniselli et al. (2023); Melo (2023); Bucaioni et al. (2024); Ziegler et al. (2024); Mastropaolo et al. (2023); Delile et al. (2023); Liu et al. (2024); Rodriguez-Cardenas et al. (2023); Ross et al. (2023); Yetistiren et al. (2022)
Software Process	Melo et al. (2020)
Explainability	Sun et al. (2022)
Dataset	Robe et al. (2022)
Software Quality Assurance	
Code Efficiency	Pan and Lyu (2023)
Code Reviews And Documentation	Tufano et al. (2024)
Load Testing	Okanović et al. (2020)
Program Repair	Prenner et al. (2022); Xia et al. (2023); Weng and Andrzejak (2023); Wei et al. (2023); Ribeiro et al. (2023); Jin et al. (2023a); Wu et al. (2023); Zhang et al. (2024)

Security And Reliability	Tony et al. (2022); Jin et al. (2023b); Shi et al. (2023); Happe and Cito (2023)
Testing Automation	Dantas et al. (2023); Nguyen et al. (2023); Guilherme and Vincenzi (2023); Kang et al. (2023); Yu et al. (2023); Li et al. (2023b); Feldt et al. (2023)
Vulnerabilities Detection	Ozturk et al. (2023); Li et al. (2023a); Asare et al. (2023)
Dataset	Tihanyi et al. (2023)
Software Design	
Gui	Brie et al. (2023)
Model-Driven Engineering	Qasse et al. (2023)
Modelling	Chen et al. (2023b); Cámara et al. (2023)
System Architecture	Ahmad et al. (2023)
Use Cases	De Vito et al. (2023b)
Software Maintenance	
Code Reviews And Documentation	Su et al. (2023)
Code Summarization	Khan and Uddin (2022)
Log Parsing	Le and Zhang (2023)
Q And A Bots	Pinto et al. (2023)
Traceability	Rodriguez et al. (2023)
Software Management	
Community Smell	Voria et al. (2022)
Effort Estimation	Hefny et al. (2021)
Agile Project Management	Dam et al. (2019)
Expert Recommentation	Cerezo et al. (2019)
Software Process Improvement	Matthies et al. (2019)
Survey	
Mapping Study	Santhanam et al. (2022)
Other	Liang et al. (2024); Belzner et al. (2024); Fan et al. (2023)
Systematic Review	Del Carpio and Angarita (2023)

Table 3 Themes, Codes, and Example Sentences from CS_HE abstracts

Themes	Subthemes / Codes	Examples from Abstracts
Challenges in Conventional Education	Communicating about code	”However, communicating about code can be difficult — particularly in asynchronous settings where an instructor authors an explanation meant to be read and understood by a student later on.” (Oney et al., 2018) ”However, developing the expertise to comprehend and explain code accurately and succinctly is a challenge for many students.” (Leinonen et al., 2023)
	Addressing diverse needs	”In traditional programming education, addressing diverse student needs and providing effective and scalable learning experiences is challenging.” (Shaka et al., 2023) ”Providing personalized assistance at scale is a long-standing challenge for computing educators” (Sheese et al., 2024)
	Scalability of creating practice opportunities and feedback	”Students learn more from doing activities and practicing their skills on assessments, yet it can be challenging and time consuming to generate such practice opportunities.” (Nguyen et al., 2022); ”Programming is an essential cross-disciplinary skill, yet teaching it effectively in large classes can be challenging due to the need for close feedback loops.” (Jell et al., 2023)
Testing GenAI capabilities	Work in a language other than English	”we report our experience using GPT-3 to solve 6 real-world tasks used in an Object Oriented Programming course at a Portuguese University and written in Portuguese.” (Cipriano and Alves, 2023)
	Solve programming examples and their variations	”We then explore how Codex handles subtle variations in problem wording” (Finnie-Ansley et al., 2022) ”evaluating the performance of Copilot on a publicly available dataset of 166 programming problems” (Denny et al., 2023) ”we conducted different coding-related experiments with ChatGPT, including code generation from problem descriptions, pseudocode generation of algorithms from texts, and code correction” (Rahman and Watanobe, 2023)
	Pass assessments or exams	”We evaluated the capability of generative pre-trained transformers (GPT), to pass assessments in introductory and intermediate Python programming courses at the postsecondary level.” (Savelka et al., 2023) ” exploring ChatGPT responses to existing assessment prompts from ten subjects across seven Australian universities” (Sasha Nikolic and Sandison, 2023) ”we present results detailing how Codex performs on more advanced CS2 (data structures and algorithms) exam questions taken from past exams” (Finnie-Ansley et al., 2023)

	Apply to other computing subjects	”we present a pipeline for generating and evaluating questions from text-based learning materials in an introductory data science course” (Nguyen et al., 2022) ”we research to what extent does using chatbots improve higher education students’ essential knowledge of computer networks?” (Ahmed and Hasnine, 2023)
	Help/tutor students	”how an LLM like ChatGPT responds to students seeking help with their introductory programming tasks”(Kiesler et al., 2023) we compare chatGPT and mentor responses in the process of learning a programming language (Moon et al., 2023)
	Prepare course materials/grade	”we explore the capabilities of an LLM - OpenAI’s GPT-3 model to provide feedback for student written code.” (Balse et al., 2023) ”we developed an LLM-powered (GPT-4) system for generation of MCQs from high-level course context and module-level LOs” (Doughty et al., 2024) ”we investigate the use of AI-generated exercises for beginner and intermediate programming courses in higher education using ChatGPT” (Speth et al., 2023)
GenAI-based innovative tools	Track and summarise code edits	”it tracks and summarizes code edits in-line with messages, allowing instructors to create explanations in stages” (Oney et al., 2018)
	Detect misconception	”a novel platform centered on addressing misconceptions” (Jell et al., 2023)
	Personalised and timely feedback	”AIPTs that offer personalised feedback through adaptive learning, accommodating diverse student backgrounds and proficiency levels” (Shaka et al., 2023) prompt practices that lead to effective next-step hints and use these insights to build our StAP-tutor (Roest et al., 2024)
	Guided explanations / assistance	”chat.codes, a new tool for creating guided explanations about code” (Oney et al., 2018) ” a system to assist programming language learning using chatGPT”(Moon et al., 2023) ”KOGI, a learning support system that integrates ChatGPT” (Kuramitsu et al., 2023) ”innovative LLM-powered tool that provides on-demand programming assistance” (Sheese et al., 2024)
	Human-like tutoring	”level-specific feedback to emulate human-like tutoring (Jell et al., 2023)
	Interactive worked examples	”WorkedGen’, which uses an LLM to generate interactive worked examples” (Jury et al., 2024)
	Prompt problems tool	”a new web-based tool called Promptly which hosts a repository of Prompt Problems and supports the automated evaluation of prompt-generated code” (Denny et al., 2024)

Good performance with limitations	Positive user experience	”beneficial for both instructors and students” (Oney et al., 2018) ”students using ChatGPT had an advantage in terms of earned scores” (Qureshi, 2023) ”using Codex significantly increased code-authoring performance” (Kazemitabaar et al., 2023)
	GenAI outperforms students; has high performance	” Codex outscores most students” (Finnie-Ansley et al., 2022) ”the majority of the automatically generated content is both novel and sensible, and in some cases ready to use as is” (Sarsa et al., 2022) ”with little modification to the input prompts, ChatGPT could generate passable responses to many of the assessments” (Sasha Nikolic and Sandison, 2023) ”[CoPilot] successfully solves around half of these problems on its very first attempt, and that it solves 60% of the remaining problems using only natural language changes to the problem description” (Denny et al., 2023)
	Limitations in handling complex input and reasoning	”poor handling of exercises requiring complex chains of reasoning step” (Savelka et al., 2023) ”ChatGPT struggles with coding exercises containing non-textual descriptions or class files, leading to invalid solutions” (Ouh et al., 2023) ”[GPT3] tends not to give the best solution in terms of object oriented design” (Cipriano and Alves, 2023)
Mixed instructional implications	Positives	”ChatGPT, as an AI-based tool, provides various advantages, such as heightened student involvement, cooperation, accessibility, and availability.” (Qureshi, 2023) ”Educators can also use ChatGPT to provide personalized learning experiences, including immediate feedback and context-based questions.” (Morsy et al., 2023)
	Negatives	”there remains a need for some oversight to ensure the quality of the generated content before it is delivered to students” (Sarsa et al., 2022) ”it has raised many questions about the authenticity of assessment and challenges in detecting plagiarism (Sasha Nikolic and Sandison, 2023) ”we expect students to use and abuse this tool in their academic work” (Cipriano and Alves, 2023) ”however, over-reliance might negatively impact learning and retention” (Kazemitabaar et al., 2023)

	Future	” research-based course-specific standard chatbots are required to teach” (Ahmed and Hasnine, 2023) ”this news should not, however, suggest that students not learn how to program but instead that instructors rethink how they teach programming.” (Jacques, 2023) ”Findings suggest that students are likely to use ChatGPT, but there is a need for specific guidelines, more classroom assessments, and mandatory reporting of ChatGPT use” (Rajabi et al., 2023)