



# Designing Web Programming Course with Trend of Labor Market in Gen AI Driven World

Reza Moosaei

Queen Mary, University of London, United Kingdom

## Abstract

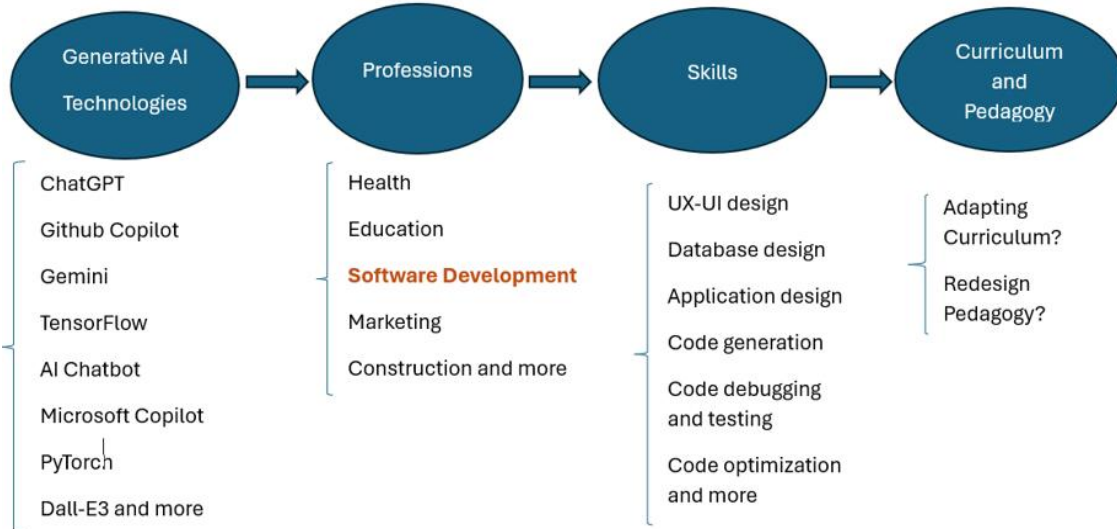
*Programming education remains a major challenge in undergraduate computing programs. Learning to program is not just about understanding a language's syntax; it also involves developing important cognitive skills such as computational thinking, abstraction, algorithmic reasoning, and structured problem-solving. In addition, students need soft skills like teamwork, critical thinking, and reflective thinking to succeed in real-world software development. The rapid growth of generative artificial intelligence (GenAI) is transforming how software is developed. While GenAI tools can automate routine programming tasks, they require developers to gain new skills in AI-assisted software development, system integration, and higher-level problem-solving. As a result, universities must rethink and update their programming curricula to meet changing industry demands. This paper explores how a programming course like web programming can be redesigned to include GenAI-related skills. Using web programming as a case study, we apply a proposed teaching framework to design a course that integrates both industry expectations and GenAI competencies. The goal is to better prepare students for GenAI-augmented software development roles and to support sustainable curriculum design in the age of generative AI.*

**Keywords:** *Generative AI, Effective Teaching Framework, Programming Pedagogy, Higher Education.*

## 1. Introduction

In higher education digital age, there is a rising demand for programming talent. Companies worldwide consistently recruit programmers to address their digital needs, leading to an increase in student interest in the field from a broad range of backgrounds [1]. In response, a wide range of educational providers including schools, universities, coding bootcamps, and online learning platforms have expanded their programming curricula [2]. Despite this growth, a challenge persists: the need to design and implement effective pedagogical frameworks that not only support foundational programming competencies but also align with the rapidly evolving labor market. This challenge is particularly significant in the context of generative artificial intelligence (GenAI), which is transforming software development practices and redefining the skills required of future computing professionals [4]. Generative AI technologies have two main impacts on the labor market: on the one hand, workforce substitution occurs as technology replaces certain specialists by performing tasks autonomously, which represents a negative impact; on the other hand, these technologies drive new employment creation and professional transformation, as industries and businesses demand new skills and experience to work with generative AI technologies, representing a positive impact. It is critical to investigate the changes in the list of expected skills on the labor market in software development associated with new generative AI technologies. This allows education institutions to analyze and adapt curriculum development according to new skills that are in demand in the job market [4][5]. Figure 1 shows what GenAI tools and skills are demanded by industry in area of software development from employees.

Existing research on GenAI in education has largely concentrated on issues such as academic integrity, ethical considerations, and the broader influence of GenAI on teaching and learning practices. To date, most educational applications of GenAI have emphasized AI literacy initiatives, the development of institutional policies, and its role in writing-based assessments. In contrast, comparatively little attention has been given to the incorporation of GenAI into teaching framework for software development and programming education [6]. It is, thus, essential to have an effective GenAI powered teaching framework for software development and programming in higher education (HE). A teaching framework that provides principles, guidelines, methods, and techniques to help instructors understand how students learn and, consequently, how to teach effectively [2]. This framework should utilize AI-powered educational technologies to enhance teaching and learning effectiveness.



**Fig. 1.** GenAI Related Skills in Software Development

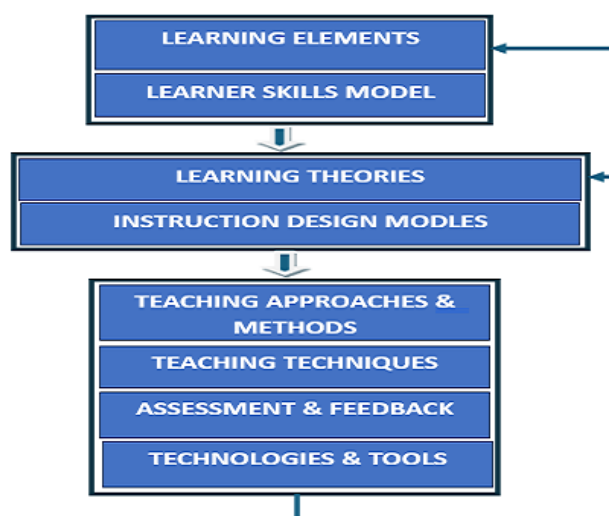
In this paper, we follow our proposed teaching framework in [7] and build a teaching framework for one of programming courses in HE called Web Programming. This framework presents a GenAI-empowered, group-based authentic designed for a Web Programming course. Specifically, the present work addresses the following research questions:

1. How can an effective teaching and learning framework be designed for programming courses to enhance student learning while aligning with industry-relevant skills?
2. How can generative AI be actively integrated into programming pedagogy to improve student learning outcomes while minimizing risks to academic integrity?

The remaining paper is organized as follows. Section 2 discusses our proposed teaching framework, and section 3 outlines the process of framework implementation for our web programming course. Finally, conclusions are presented in section 4.

## 2. Proposed Teaching Framework

Our previous paper [7] focused on programming pedagogy and in that work, we identified the essential components needed to build an effective teaching framework for programming which are elaborated further in the below (shown in figure 2).



**Fig. 2.** Essential Components for Teaching Framework

**A. Programming Learning Elements (Topics):** refers to the core components of knowledge that students must learn in a programming course. It represents the building blocks of programming learning



objects including syntactic knowledge (language rules), conceptual knowledge (how constructs program), and strategic knowledge (how to apply them to solve problems).

**B. Learners Skills Model:** describes the progression of skills that students are expected to develop throughout a programming course. This model provides a structured view of skill development that supports course design, instructional planning, and assessment.

**C. Learning Theories:** Learning theories provide the pedagogical foundation for designing effective programming courses by explaining how learners acquire knowledge and learn.

**D. Instructional Design Models:** Instructional design models provide systematic approaches for planning, developing, and evaluating teaching and learning processes.

**E. Teaching Approaches and Methods:** Teaching approaches are the overall educational philosophies or strategies that guide instruction. Examples include teacher-centered and learner-centred approaches. In contrast, teaching methods are the specific activities used to implement those approaches in the classroom. Examples include hands-on labs, coding exercises, projects, group work, and discussions.

**F. Teaching Techniques:** Teaching Techniques are the specific, practical actions and tactic an instructor uses in implementation of a teaching method to support learning. For instance, in pair programming methods, the instructor can use a driver/navigator strategy (one student codes and the other reviews it).

**G. Assessment and Feedback:** Assessment and Feedback refer to the processes used to measure student learning and provide information that helps learners improve their performance. In education, especially in programming courses, assessment and feedback are essential for evaluating learning outcomes and guiding skill development.

**H. Technologies and Tools:** Technology has become a powerful educational tool, transforming how students learn, and teachers instruct. Digital devices, software, and internet-based resources enable a more interactive, personalised, and engaging classroom learning experience.

In next section, we describe the process of teaching framework implementation for our web programming course.

### 3. Teaching Framework Implementation for Web Programming Course

We selected web programming as case study for our study. Web programming has become a foundational skill in modern computing education due to the rapid growth of web-based applications, services, and platforms across nearly all industries. From e-commerce and social networking to data visualization and cloud services, the web has evolved into a primary medium for software deployment. As a result, higher education institutions face increasing pressure to design web programming courses that not only teach technical skills but also prepare students to adapt to continuously changing technologies and development practices. Instructors must balance fundamental concepts such as client-server architecture, HTTP, and web security with practical skills including front-end and back-end development, database integration, and responsive design. Normally this course is offered for second year students for Bachelor of Computer Science related programs. Additionally, students often enter these courses with varied backgrounds in programming, which further complicates course design and instructional strategies [8][9]. This paper explores the design of a web programming course that addresses these challenges and considering industry and GenAI related skills adapted to labor market demand, emphasize on hands-on learning, progressive skill development, and the integration of real-world development practices. By focusing on both conceptual understanding and practical application, the course aims to enhance student engagement, improve learning outcomes, and better prepare students for professional web development environments. The remainder of this paper describes the development of teaching framework for a web programming course.

#### 3.1 Learning Objects Model for Web Programming

To design an effective teaching framework, instructors must first identify the core learning elements of programming courses. Prior works have classified programming knowledge into syntactic, conceptual, and strategic categories, encompassing language syntax, underlying principles, and logics [10][11]. Considering this, the learning elements of programming courses are organized into four categories: logic, programming language, programming (code), and tools. Each category includes key learning elements, such as writing, executing, and debugging code. Table 1 outlines learning objects for our web programming course and provides basis for defining learning objectives in web programming courses.



**Table 1.** Learning Object Model for Web Programming

Logic	Programing Language	Program/code	Tools
Computationally analyzing problem	Concepts and constructs	Write program	Code generation
Logic language	Syntax	Build and run	Build automation
Design algorithm	Design patterns and libraries	Debug and test	Debugging

### 3.2 Student Skills Model for Web Programming

We draw a skill model for web programming course under the models suggested by SFIA [12], CC2020 [13] and IT2017 reports [14]. The proposed model in table 2 consolidates the core hard and soft skills required for a web programming course. By categorizing skills according to type and proficiency level, the model provides structured guidance for syllabus design, informing the selection of learning materials, instructional methods, and classroom activities. Furthermore, the skill model serves as a reference point for evaluating the effectiveness of teaching methods and pedagogical techniques. By aligning instructional practices with clearly defined skill categories, instructors can more systematically articulate learning objectives and assess whether course outcomes are being achieved at both the lesson and program levels.

**Table 2.** Skill Model for Web Programming

Soft Skill	Professional Skills				
	Problem Solving			Critical Thinking	
	Behavior / Attitude				
	Motivation	Communication	Teamwork	Self-efficacy	Time Management
Hard Skill	Design Skills				
	Algorithm	Conceptual Knowledge		Syntactical	
	Implementation Skills				
	Strategical Knowledge	Test and debug		Tools	
	Generative AI Skills				
Algorithm design			Code generation		

### 3.3 Learning Theory for Web Programming

It is important for instructors to base their teaching on appropriate learning theories, as these theories provide clear principles for how learning should take place. Learning theories guide the design of teaching approaches, methods, and classroom activities, ensuring that instruction is effective and aligned with how students learn best [15]. One widely used theory in practical and skills-based courses is constructivism. This theory suggests that learners build new knowledge by connecting new ideas to their previous experiences and existing understanding. Students develop deeper understanding by reflecting on and refining their prior knowledge. In practice, constructivist teaching strategies include problem-based learning, project work, research activities, and group collaboration [16].

We selected constructive learning theory as effective learning theory for our web programming course because:

- Learning by doing mirrors real-world web development
- Encourages problem-solving and critical thinking
- Constructivism values social interaction in knowledge construction. web programming naturally supports this through pair programming, code reviews, and group projects.



### **3.4 Instructional Design Model**

Instructional design is a systematic process for developing effective teaching materials, learning activities, and instructional strategies to improve educational outcomes. Instructional design models provide structured guidance for instructors in planning and organizing educational programs, courses, or individual lessons. By following these models, educators can ensure that learning objectives are clearly defined, aligned with instructional activities, and effectively achieved [18]. We used ADDIE Model as a standard instructional design model including a five-phase model (Analysis, Design, Development, Implementation, and Evaluation) that provides a step-by-step approach to developing instructional materials. The ADDIE instructional design model is suitable for web programming courses because its flexibility supports continuous improvement and provides a systematic framework for analyzing learner needs, designing structured learning experiences and evaluating both learning outcomes and instructional effectiveness.

### **3.5 Teaching Approach, Methods and Strategies for Web Programming**

The student-centered approach is the most effective in involving learners and giving them responsibility for their learning path and progress [19]. Overarching within a teaching approach are the multiple teaching methods that develop depending on the principles of the teaching approach. A teaching method is a series of step-by-step actions teachers and students undertake to achieve specific learning objectives. Luxton-Reilly [10] describe a range of teaching methods based on a student-centered teaching approach for teaching programming. Practical examples of these methods for instructors in programming courses are project-, problem-, case-, and inquiry-based learning methods, as well as gamification, flipped-the-classroom, live-coding, and peer-instruction methods. In contrast, teaching strategies are the styles and tactics used in the classroom, sustaining a specific teaching method to accomplish a learning objective effectively. For instance, in pair programming, often used within project-based methods, the instructor can use a driver/navigator strategy (one student codes and the other reviews it), an unstructured strategy (two students work together in an ad hoc manner), or a ping-pong strategy (one student writes code and the other tests it, shifting roles regularly). Hence, instructors must be equipped with a palette of expertise to select and apply effective teaching methods and strategies, guided by the fundamental principles of the learning theory, instructional design model, learning elements, and an evaluation of students' skill levels, as discussed above [19][20].

Generative AI is transforming teaching and learning methods by shifting instruction from content delivery toward guided knowledge construction and higher-order thinking. Tools such as ChatGPT, Google Gemini, and Claude can generate explanations, examples, practice problems, simulations, and formative feedback on demand, enabling highly personalized and adaptive learning pathways. Instructors can use generative AI to design differentiated materials, create scenario-based learning activities, and automate feedback cycles, thereby freeing time for mentorship and conceptual discussion. For learners, these systems support idea generation, and iterative refinement of work, which can enhance metacognitive awareness and self-regulated learning [21][22][23].

Considering the effectiveness of various teaching methods and the emerging impact of GenAI, we adopt the following methods and strategies for delivering the web programming course in upcoming semester (as shown in figure 3):

- GenAI-powered flipped classroom: The flipped classroom is effective for web programming courses because it shifts content delivery outside the classroom and uses in-class time for hands-on coding, problem-solving, and collaboration. This approach supports active learning, accommodates diverse learner backgrounds, and enhances students' ability to apply programming concepts in real-world contexts. To support this, for each session we provide pre-class reading materials, short instructional videos, sample code demonstrations, and access to an online discussion forum. Additionally, students are encouraged to interact with GenAI as a "pre-class tutor," asking clarification questions and testing understanding before attending class. This Promotes self-regulated learning and prepares students for deeper in-class engagement. During class, generative AI is used to support higher-order learning activities such as analysis, evaluation, and collaborative problem-solving. Learners critically examine, refine, and compare AI-generated outputs with their own generated output. For mastery enhancement, students are encouraged to use generative AI after class to discover advanced solutions as well.
- Project based: Project-based teaching is effective for web programming courses because it integrates theoretical instruction with authentic, hands-on projects. This approach promotes active



learning, enhances problem-solving skills, increases student engagement, and aligns academic learning with real-world web development practices. We start with a real-world project that integrates core web technologies such as E-commerce web applications. We break the course into structured development phases and instruction is structured around iterative project milestones including requirements analysis, interface design, backend implementation, integration, testing, and deployment. Students are required to engage with GenAI as an object of critique and inquiry, which in turn strengthens their critical thinking skills, metacognitive awareness, collaborative engagement, and AI literacy.

- Live coding with GenAI powered pair programming: Live coding is effective in web programming courses because it shows real coding practices, makes problem-solving and debugging visible, and connects theory to hands-on work. Combining live coding with pair programming improves engagement, reduces cognitive overload, and models expert thinking. We use a driver/navigator strategy that one student will do coding, and the other student will review it. GenAI systems like ChatGPT add a new dimension by acting as adaptive collaborators that can generate code, explain logic, suggest alternatives, and simulate edge cases. In our approach, the instructor codes live while students query AI for alternatives and compare human versus GenAI reasoning.
- Lab learning activities provide a structured, hands-on environment that enhances both individual and collaborative learning in programming courses. During lab, instructors conduct workshops on coursework offer students dedicated time to apply theoretical concepts, experimenting with code, and receiving immediate guidance from instructor, fostering deeper comprehension and practical skills. Mini presentations of coursework progress during lab sessions, encourage students to articulate their problem-solving approaches, reflect on challenges, and gain constructive feedback, promoting self-regulation and communication skills essential for software development.
- knowledge check: End-of-lecture live quizzes are effective in web programming courses because they reinforce newly introduced concepts, provide immediate formative feedback, and support active engagement and early identification of learning gaps.

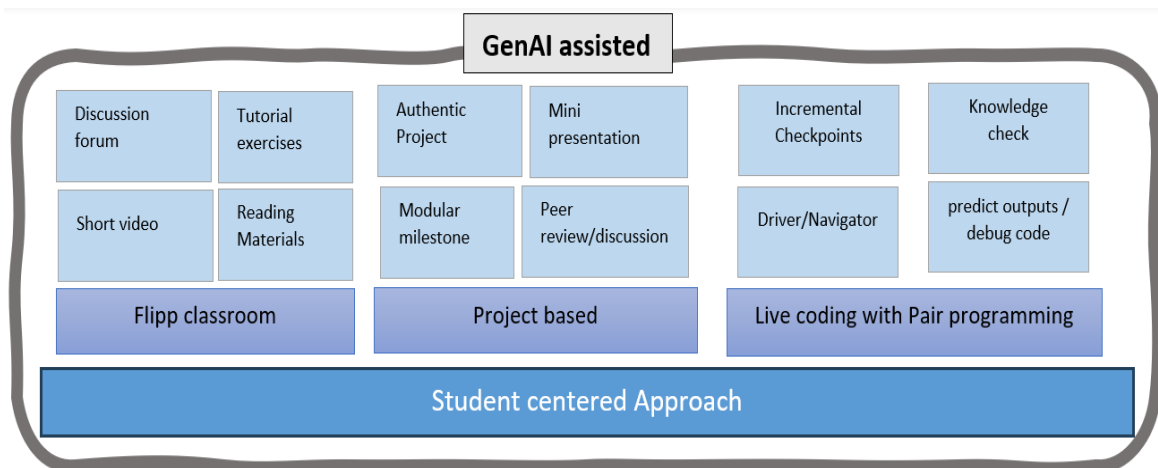


Fig. 3. Essential Components for Teaching Framework

### 3.6 Assessments

Assessments in programming courses should focus not only on code writing and correctness but also on problem solving processes, code optimization, testing and documentation practices. Well-designed assessments meet the course's learning outcomes and provide deeper learning, enhance critical thinking, and prepare students for the practical challenges of the industry [13]. GenAI technologies drive new professional transformation in programming, as software development industries demand new skills and experience to work with generative AI technologies. Therefore, it is critical to design assessments to cover the expected GenAI related skills in web programming such as web interface design, web database design and code generation. With the growing use of generative AI tools, some students may misuse these resources, increasing the risk of academic misconduct. To address this challenge, instructors should discuss AI with their students and consider revising their assessment designs. To reduce the risk of misconduct, instructors can rethink the types of assessments, break tasks into smaller components, provide sufficient time, ask for progress reports, and emphasize the process instead of the final product [21][22][23].

The assessment for our Web Programming course consists of two components: project coursework



(60%) and a final exam (40%).

The coursework requires students to design and implement a website for a small business (such as car sales, real estate, or an online shop). The project is completed in groups of four students. Throughout the project, students must use GitHub as a collaboration and version control tool. This enables instructors to monitor the development process and track individual contributions. Assessment focuses on the development process as well as the final product. Each group is provided with a set of required system features, and every student is assigned responsibility for implementing a specific feature. Students must hold weekly meetings and submit meeting minutes along with progress reports. In addition, each group is required to deliver a live presentation demonstrating both the development process and the final product. This ensures that all group members have actively contributed to the project. Coursework project is broken down to three deliveries as follows:

Delivery 1 – Student generated website: students need to design and build website by using their learning knowledge from class. Students are not allowed to use AI tools and project must be entirely original and fully designed and developed by students.

Delivery 2 - GenAI-powered generated website: Students are allowed to use GenAI to build their website. By this way instructor makes sure that students have learned the GenAI related skills for designing web pages, database and generating codes.

Delivery 3 - AI powered platform for building websites: To keep students remaining up to date with current web technology in area of website development, students are required to investigate recent AI powered platform for website developments and present their findings through a live presentation.

### **3.7 Feedback and Survey**

The structured feedback and survey results provide insight into how active learning strategies such as live coding, pair programming, and project-based learning impact student engagement and conceptual understanding. The findings can inform iterative course refinement and evidence-based instructional design. To evaluate the effectiveness of the implemented programming pedagogy, a feedback framework was employed. The feedback framework combines formative feedback, instructor feedback, and a post-course evaluation. Formative feedback was integrated throughout the course to support continuous improvement. The instructor feedback was used to assess depth of students' understanding and learning. A post evaluation was conducted to assess student perceptions of the teaching methodology. The evaluation included two separate surveys: one assessing students' perceptions of teaching effectiveness, and another investigating the reasons for students with a low-class attendance.

### **3.8 Conclusion**

Programming education must adapt to both learning challenges and the growing impact of GenAI on software development. This paper introduced a redesigned web programming course that combines active learning methods such as flipped classrooms, live coding, pair programming, project-based learning, and authentic assessment with a GenAI-powered teaching framework. Rather than replacing foundational skills, GenAI is used to support critical thinking, reflection, collaboration, and evaluation of AI-generated code. The framework aims to strengthen core programming competencies while also developing industry-relevant AI skills, including prompt engineering, validation of AI outputs, and responsible AI use. Findings suggest that GenAI can improve engagement and learning when integrated thoughtfully within structured and reflective learning environments. The study highlights the importance of balancing strong programming foundations with AI-augmented development practices and positions GenAI as an opportunity to innovate programming pedagogy and curriculum design in higher education.

## **REFERENCES**

- [1] Tony Jenkins (2002). On the difficulty of learning to program. Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences, Pages 53-58, Loughborough University, United Kingdom.
- [2] Wilfred W.F. Lau, & Allan H.K. Yuen (2009). Toward a Framework of Programming Pedagogy, In Book Chapter of Encyclopedia of Information Science and Technology, Second Edition, Pages 3772-3777, Publisher IGI Global.
- [3] Literature Review on the Integration of Generative AI in Programming Education. Int. J. Artificial Intelligence in Education, 2025. (Systematic review on GenAI integration and structured pedagogy)

- [4] Garousi, V., Jafarov, Z., Movsumova, A., Namazov, A., & Mirzayev, H. Encouraging Students' Responsible Use of GenAI in Software Engineering Education. arXiv 2025. (Model for responsible GenAI use in SE education)
- [5] Chen, Yue & Chai, Michael & Loo, Jonathan & Moosaei, Reza & Obstfeld, Joel. (2025). GenAI-Empowered Group-Based Authentic Assessment for Network Engineering Courses. 1-7. 10.1109/EDUCON62633.2025.11016382.
- [6] Moosaei, Reza & Molinero, Gloria. (2025). Pedagogical Framework for Programming Courses in Higher Education. 1-5. 10.1109/EDUCON62633.2025.11016289.
- [7] Hauser, K., Olsen, D., & Fadel, K. (2018). An integrated approach to teaching web development. Review of Business Information Systems. (Explores curriculum design that integrates web programming, databases, and application integration)
- [8] Chang, C., Tseng, K., Lin, Y., & Liang, J. (2021). Applying the design-based learning model to foster undergraduates' web design skills: The role of knowledge integration. International Journal of Educational Technology in Higher Education, 18(1). <https://doi.org/10.1186/s41239-021-00308-4> (Discusses a structured teaching model for web design and programming that integrates active learning strategies)
- [9] SFIA Skills Framework for Information Age, (2011). Published by SFIA Foundation.
- [10] ACM and IEEE-CS Release Computing Curricula 2020, (Published 2021). Paradigms for Global Computing Education, Association for Computing Machinery (ACM) IEEE Computer Society (IEEE-CS). ISBN: 978-1-4503-9059-0, <https://dl.acm.org/citation.cfm?id=3467967>.
- [11] Information Technology Curricula 2017. Curriculum Guidelines for Baccalaureate Degree Programs in Information Technology, A Report in the Computing Curricula Series. Task Group on Information Technology Curricula Association for Computing Machinery (ACM), IEEE Computer Society (IEEE-CS). <https://www.acm.org/binaries/content/assets/education/curricula-recommendations/it2017.pdf>
- [12] Andrew Luxton-Reilly, Simon, Ibrahim Albluwi, Brett A. Becker, Michail Giannakos, Amruth N. Kumar, Linda Ott, James Paterson, Michael James Scott, Judy Sheard, & Claudia Szabo (2018). Introductory programming: a systematic literature review, In ITiCSE 2018: Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education, Pages 55-106, University of Central Lancashire, Cyprus.
- [13] S.C. dos Santos, Patricia Azevedo Tedesco, Matheus Borba, & Matheus Brito (2020). Innovative Approaches in Teaching Programming: A Systematic Literature Review. In Proceedings of the 12th International Conference on Computer Supported Education (CSEDU), Vol. 1, ISBN 978-989-758-417-6, SciTePress, Pages 205-214, Czech Republic, <https://doi.org/10.5220/0009190502050214>
- [14] Sue Sentance, Jane Waite, & Maria Kallia (2019). Teachers' Experiences of Using PRIMM to Teach Programming in School. In SIGCSE '19: The 50th ACM Technical Symposium on Computer Science Education, Pages 476-482. ISBN 9781450358903, Minneapolis, USA, <https://doi.org/10.1145/3287324.3287477>
- [15] Mohammad Tafiqur Rahman (2023). Blended Approach for Deep Learning: A Framework for Teaching Undergraduate Computer Programming Courses, Proceedings of the 15th International Conference on Computer Supported Education - Volume 2: CSEDU, 394-402, 2023, Prague, Czech Republic.
- [16] Mary Tom (2015). Five Cs Framework: A Student-centered Approach for teaching programming courses to students with diverse disciplinary background, in Journal of Learning Design, Vol. 8, Pages 21-37, ISSN 1832-8342, <http://dx.doi.org/10.5204/jld.v8i1.193>
- [17] Molly Zhou, & David Brown (2015), Educational Learning Theories: 2nd Edition. Education Open Textbooks.
- [18] Anthony Robins, Janet Rountree, & Nathan Rountree (2003). Learning and Teaching Programming: A Review and Discussion, Journal of Computer Science Education, 13(2), Pages 137—172, <https://doi.org/10.1076/csed.13.2.137.14200>
- [19] The effect of AI-assisted pair programming on student motivation, programming anxiety, collaborative learning, and programming performance. (2025). International Journal of STEM Education, 12, Article 37. <https://doi.org/10.1186/s40594-025-00537-3>
- [20] Kohen-Vacs, D., Usher, M., & Jansen, M. (2025). Integrating generative AI into programming education: Student perceptions and the challenge of correcting AI errors. International Journal of Artificial Intelligence in Education, 35, 3166–3184.
- [21] McCulloh, I., Rodriguez, P., Kumar, S., Gupta, M., Sharma, V. R., Johnson, B., & Johnson, A. N. (2025). Generative AI in computer science education: Accelerating Python learning with ChatGPT.