Journal of Information Technology Education: Research

An Official Publication of the Informing Science Institute InformingScience.org

JITEResearch.org

Volume 24, 2025

CONQUERING CODING FEARS: A SYSTEMATIC REVIEW OF PROGRAMMING ANXIETY IN HIGHER EDUCATION

Ying Pan*	Faculty of Educational Sciences and Technology, Universiti Teknologi Malaysia, Johor, Malaysia	panying@graduate.utm.my
Jamalludin Harun	Faculty of Educational Sciences and Technology, Universiti Teknologi Malaysia, Johor, Malaysia	p-jamal@utm.my

Corresponding author*

ABSTRACT

Aim/Purpose	This review analyzes the causes and effects of programming anxiety in higher education, evaluates existing assessment tools, and explores mitigation strategies to guide interventions that improve student outcomes.
Background	Programming anxiety impairs educational outcomes by reducing student engagement, performance, and retention. Prior studies have typically focused on its correlations with performance, programming language selection, and gender; a more comprehensive investigation of this topic is demanded.
Methodology	Following PRISMA guidelines, this systematic review analyzed 21 empirical studies from databases including ScienceDirect, Springer, Scopus, WOS, ACM Digital Library, and IEEE Xplore, based on predefined inclusion criteria.
Contribution	This review summarizes existing knowledge about programming anxiety, identifies key gaps, establishes its causes and effects, evaluates measurement tools, and proposes interventions that have not previously been systematically reviewed.
Findings	Both intrinsic and extrinsic factors contribute to programming anxiety. It harms learning outcomes and may lead to lower course completion rates. The current approaches for evaluating programming anxiety lack standardization and validity. In addition, findings suggest that effective interventions require a supportive learning environment combined with pedagogy to alleviate programming anxiety.

Accepting Editor Kay Fielden │ Received: February 20, 2025 │ Revised: April 23, April 26, 2025 │ Accepted: May 25, 2025.

Cite as: Pan, Y., & Harun, J. (2025). Conquering coding fears: A systematic review of programming anxiety in higher education. *Journal of Information Technology Education: Research, 24,* Article 20. https://doi.org/10.28945/5543

Recommendations for Practitioners	Practitioners are advised to use instructional strategies that reduce programming anxiety, such as collaborative learning, supportive interactions, appropriate IDEs (Integrated Development Environments), and anxiety-specific assessment tools to tailor educational approaches.
Recommendations for Researchers	Researchers are encouraged to develop validated, reliable instruments for evaluating programming anxiety and investigating the efficacy of specific solutions in a variety of educational settings.
Impact on Society	Mitigating programming anxiety has significant implications for educational practices, as it can improve the quality and accessibility of computer science education, potentially enhancing diversity and inclusion in technology-related disciplines.
Future Research	Future research should investigate the long-term impacts of programming anxiety, design interventions that target both psychological and pedagogical components of anxiety and evaluate the effectiveness of these interventions across different demographic groups.
Keywords	programming anxiety, cause and effect, assessment tool, instructional interventions, higher education, systematic literature review

INTRODUCTION

BACKGROUND

Computer science, which is a vital skill in the 21st century, has permeated technology, business, and even everyday life, bringing about significant transformations in our way of life. All these transformations are built on the same foundation: programming (Dirzyte et al., 2021; Wang et al., 2019). Beyond being a technical skill, programming also represents a problem-solving methodology. Learning programming can foster learners' logical thinking, creativity, and ability to tackle complex problems, thus enhancing their learning capabilities and computer thinking (Huang et al., 2021). As a result, many higher education institutions have added introductory programming courses to their curricula (Cawthorne, 2021; Xu & Correia, 2023).

However, as computer programming courses rapidly expand to meet industry demands, many students have little exposure to programming concepts before taking these classes, exacerbating the challenge of mastering this skill (Inventado, 2019). Failure rates in introductory programming courses range from 25% to 33%, underscoring the urgency of addressing potential difficulties in programming education (Bennedsen & Caspersen, 2007, 2019; F. Demir, 2022; Nolan & Bergin, 2016). For college students, beyond the need to comprehend and apply a wide array of new concepts, terminology, and syntactic rules (Figueiredo & Garcia-Penalvo, 2018), psychological obstacles to the abstract concepts being learned may arise due to a lack of understanding of how programs work. This can lead to students becoming anxious about, and even afraid of, programming (Connolly et al., 2007; Melcer & Isbister, 2018).

PROGRAMMING ANXIETY AND ITS EFFECTS

Spielberger (2013) defined anxiety as an unpleasant emotional state marked by tension, apprehension, and worry. Anxiety comprises two types: state anxiety, which is temporary and situation-specific, and trait anxiety, which is stable and reflects a persistent tendency to feel anxious across situations. Programming anxiety, a subset of computer anxiety (itself a form of general anxiety), occurs when students incorrectly assess their ability to learn computer programming (Connolly et al., 2007, 2009). Figure 1 illustrates the relationship between these forms of anxiety.

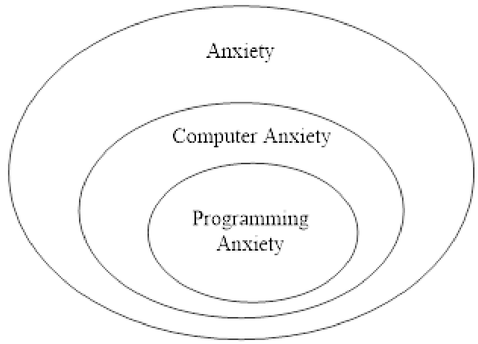

Figure 1. The relationship between anxiety, computer anxiety, and programming anxiety

Psychological research consistently identified anxiety as a prominent factor in psychological problems, hindering learning and reducing both motivation and performance (Li et al., 2022; Suren & Kandemir, 2020). In programming contexts, according to the Cognitive Model of Programming Anxiety (Connolly et al., 2007), most students perceive programming to be difficult at the beginning, which in turn creates a fear of programming (referred to as core beliefs). In a classroom learning environment, these inherent beliefs are exacerbated by the fear of how others will perceive one's programming abilities, which results in negative automatic thinking about programming. In addition, other external factors such as the laboratory equipment, the instructor's teaching style, and the overall teaching methodology can further contribute to these automatic thoughts, triggering a cascade of negative thoughts and self-reflection, as shown in Figure 2.

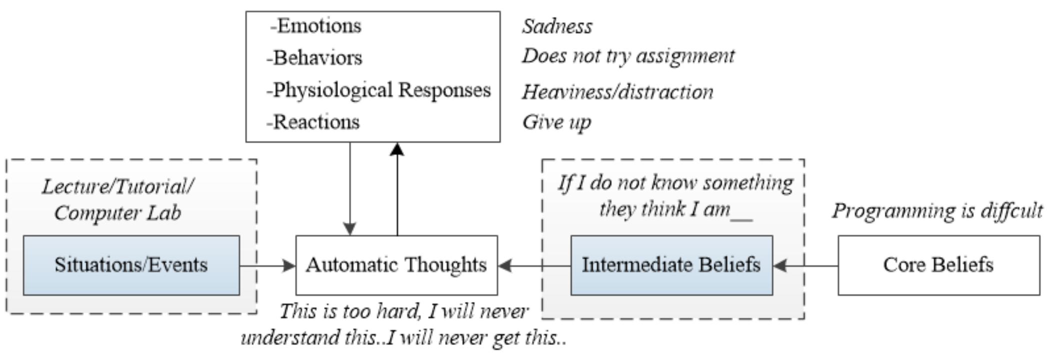

Figure 2. Cognitive model of programming anxiety (Connolly et al., 2007)

Moreover, introducing novel concepts and materials during the early stages of programming learning will increase the risk of triggering programming anxiety among students, thus hindering programming learning and reducing retention rates (Connolly et al., 2009; Scott & Ghinea, 2014). Forrester et al. (2022) have found that low levels of anxiety can enhance student motivation, whereas moderate to high levels of anxiety adversely affect learning. Therefore, overcoming programming anxiety is key to helping students succeed and grow in programming. Students can attain more programming progress by comprehending, recognizing, controlling, and dealing with programming anxiety (de Siqueira et al., 2022).

Research Gap and Objectives

At the time of writing, there was only one published review on programming anxiety, which focused on its relationship with factors such as programming languages, syntax, math anxiety, computer use, and test anxiety (Nolan & Bergin, 2016). However, with the evolving landscape of programming education, driven by new tools, technologies, and pedagogy, there is a greater need to investigate a broader range of causes and impacts, assess reliable tools for measuring programming anxiety, and identify effective strategies for alleviating it. This systematic review seeks to bridge these gaps and provide a more comprehensive understanding of programming anxiety in higher education. To further this investigation, the following research questions guided this study:

1. What are the causes of programming anxiety among college students?
2. What are the effects of programming anxiety on college students?
3. What are the tools for assessing programming anxiety among college students?
4. What coping mechanisms and educational strategies effectively alleviate programming anxiety?

METHODOLOGY

A systematic review was used to identify, select, and analyze relevant literature on programming anxiety in higher education settings. The publications cited in this study contain those that met the screening criteria for the research questions, while additional works related to programming anxiety were included to further enrich the content.

RESEARCH STRATEGY

This study followed the PRISMA (Page et al., 2021) guidelines for conducting and reporting a literature review. ScienceDirect, Springer, Scopus, and Web of Science (WOS) were selected for their extensive coverage of peer-reviewed journals in both education and computer science disciplines. In addition, as noted by Pears and Seidman (2007), "programming" frequently gets "hidden" under the "Computer Science" title in literature, which makes it more challenging to conduct a literature search, so the ACM Digital Library and IEEE Xplore were also searched so that relevant articles would not be omitted.

Table 1. Articles identified and included in the review of databases

Database	Search Terms	Number of articles		
		Initial Count	Reviewed Abstracts	Included Articles
Scopus	TITLE-ABS-KEY ("programming" OR "coding") AND TITLE-ABS-KEY ("anxiety") AND TITLE-ABS-KEY ("teach*" OR "learn*")	721	44	8
WOS	TS = ("programming" OR "coding") AND TS = ("anxiety") AND TS = ("teach*" OR "learn*")	250	37	9
IEEE	"All Metadata": "programming" OR "All Metadata": "coding") AND ("All Metadata": "anxiety") AND ("All Metadata": "teach" OR "All Metadata": "learn"	27	8	4
ACM	[Abstract: "programming"] OR [Abstract: "coding"] AND [Abstract: "anxiety"] AND [[Abstract: "learn*"] OR [Abstract: "teach*"]	34	9	0
Springer	"Programming" AND "anxiety" AND ("learn" OR "teach")	269	16	7
Elsevier	("Programming" OR "coding") AND "anxiety" AND ("teach" OR "learn")	145	3	0
Others	Citation searching	27	6	3
Articles with Duplicates	-	1473	123	31

The search strategy used various terms, which were combined into the following search statements: ("programming" or "coding"), "anxiety," and (teach* or learn*). Searches were limited to document titles, abstracts, and keywords. This approach ensured the inclusion of studies exploring the intersection of programming anxiety with approaches to learning or teaching. Table 1 illustrates the databases used, search strings, initial search results, articles left after title and abstract screening, and the final number of articles that met the inclusion criteria. The article will include the selected research that meets the inclusion criteria, but the citations may also include other publications about programming anxiety.

INCLUSION AND EXCLUSION CRITERIA

All studies that appeared to meet the inclusion criteria were screened based on document type, language, peer review status, study population, page number, and relevance to research questions, as shown in Table 2. The full text was reviewed if the title or abstract did not provide enough information to decide.

Table 2. Inclusion and exclusion criteria

Criteria	Included	Excluded
Document Type	Primary research	Reports, letters, editorials, abstracts, reviews
Language	English	Written in other languages
Population	College students	All other population settings
Peer Reviewed	Peer-reviewed papers	Not peer-reviewed papers
Data-based	Qualitative, quantitative, or mixed methodology	Without empirical data, such as theoretical, conceptual, or opinion pieces
Page Number	> 4	<= 4
Research Questions related	Articles addressing at least one of the four research questions	Articles not related to any of the four research questions

DATA EXTRACTION AND REFINEMENT

To systematically manage and analyze the literature, Zotero, a free, open-source reference management software (https://www.zotero.org) (Courraud, 2014), was used to organize and categorize the studies, and an Excel spreadsheet was created to extract and record key data. A standardized screening checklist based on the inclusion and exclusion criteria (Table 2) was developed and pilot-tested on a subset of records to guarantee clarity and consistency. To maintain consistency and reduce selection bias in research inclusion, two authors independently screen the title/abstract and full-text, and any disagreements can be addressed through discussion and, if needed, contact with a third reviewer. Initially, 1,446 records were identified from six major databases, with an additional 27 records obtained through citation searches. After removing irrelevant, non-empirical, or duplicate entries, 437 records moved to the title and abstract screening stage. Following the review process above, only 21 studies met the inclusion criteria aligned with the research questions. This careful selection and analysis process ensured the rigor and comprehensiveness of the review, as illustrated in the PRISMA flowchart in Figure 3.

DATA ANALYSIS

After data extraction, both macro-descriptive and thematic approaches were used to establish a firm foundation for synthesis. In the macro-descriptive analysis, publication trends by year, database source, and country of origin were quantified using Excel to produce clear frequency distributions. In the thematic analysis, a coding framework, which was directly derived from the four research questions, was applied to each study's "Focus" and "Key Findings Related to this Study" items. Initial codes (such as causes, effects, assessment tools, and interventions) were iteratively refined. Any differences were then settled by discussion or, if required, a third reviewer. Finally, the themes for each research question were synthesized by grouping similar codes across studies, ensuring that the resulting answers present a thorough explanation of the causes, impacts, assessment techniques, and mitigation measures of programming anxiety.

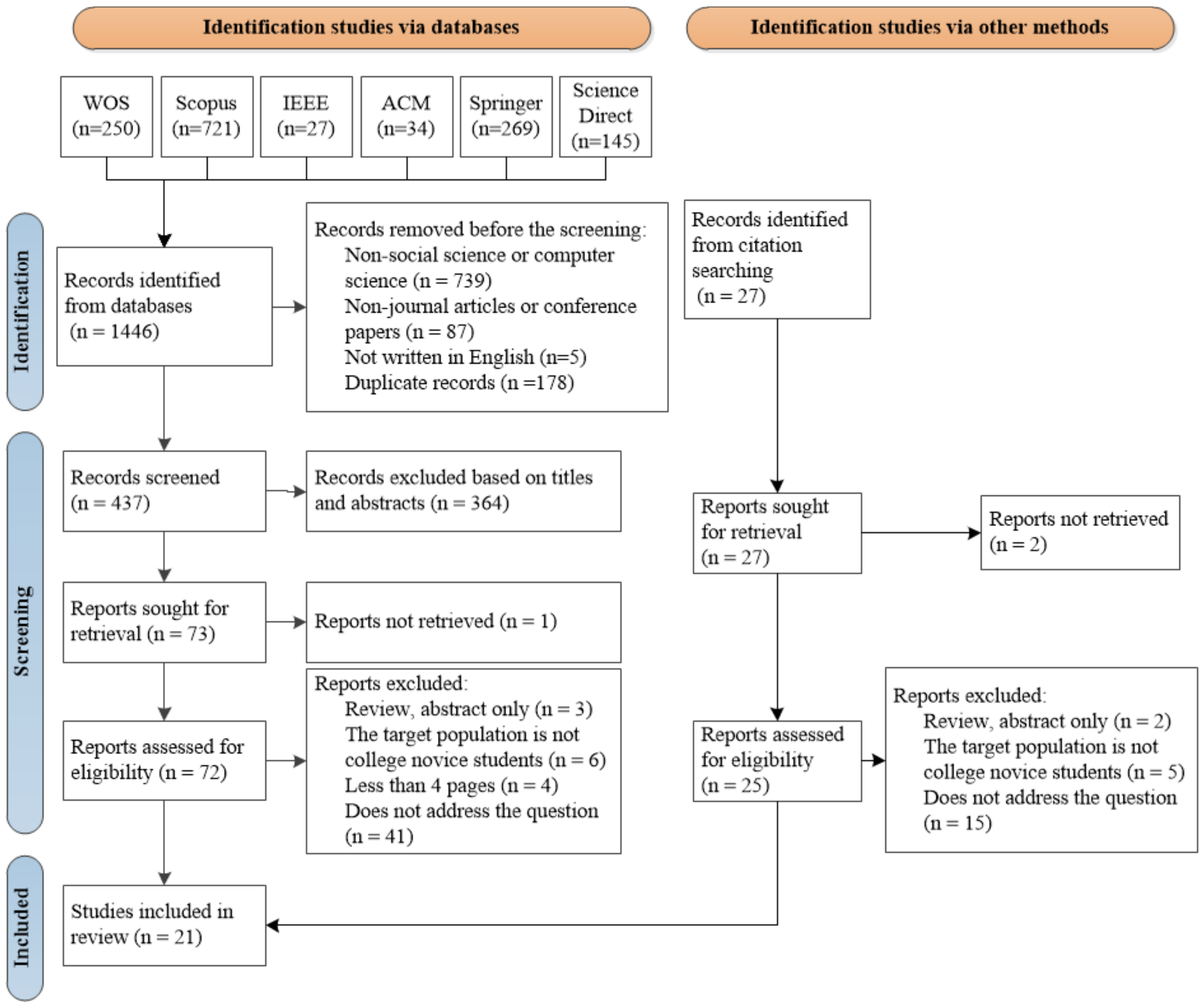

Figure 3. Flowchart of the study selection process

RESULT

PUBLICATION DISTRIBUTION AND TRENDS ANALYSIS

Table 3 summarizes 21 primary studies on programming anxiety, detailing author and publication year, geographic region, sample size with population characteristics, thematic focus, and key findings related to this review. The studies cover a variety of areas, including the effects of programming anxiety, instructional strategies, and technology tools, providing valuable insights into the factors that influence programming anxiety and suggesting possible interventions to improve students' learning experiences.

Table 3. Systematic review articles on programming anxiety

Author, data	Region	Sample Size	Citations	Focus	Key Findings Related to this Study
Durak and Bulut (2024b)	Japan	763 students with prior programming experience.	12	Cause /Impact	Programming anxiety affects high performers but not those with lower performance levels.
Chang et al. (2024)	Taiwan, China	149 college students	16	Pedagogy	A proposed SOP for computational thinking helps delineate logical steps before coding, significantly improving learning outcomes, motivation, and anxiety reduction, particularly in female students.
Durak and Bulut (2024a)	Japan	763 students with prior programming experience.	1	Cause/ Impact	No statistical difference in programming anxiety scores based on students' performance levels was observed
Fernalld et al. (2023)	United States	42 undergraduate students	6	Tools	Decentralized platforms that enhance stability and focus on learning goals help lower technical issues and programming anxiety.
Charles and Gwilliam (2023)	United Kingdom	149 college students	8	Tools	High anxiety in students correlates with poor performance. Simplified error messages and supportive measures like ERROR EXPLAINER, mastery-oriented grading, and retesting options effectively lower anxiety and boost performance.
Paredes-Velasco et al. (2022)	Spain	57 university students in an introductory programming course	8	Pedagogy	Students' anxiety may stem from a sense of difficulty in programming, and interventions should simplify the task and increase confidence.
Cheng et al. (2022)	Taiwan, China	184 undergraduate students	15	Pedagogy / Tools	Integration of online technology and pedagogy enhances programming skills and reduces anxiety, while Content-based Knowledge Awareness (CoKA) did not meet expectations.
de Siqueira et al. (2022)	United States	29 college students	23	Tools	A block-based programming environment is proposed to reduce college students' programming anxiety
Yildirim and Ozdener (2022)	Japan	687 university students	4	Assessment	A newly developed programming anxiety scale for college students measures the impact of classmates and self-confidence through 11 items.

Author, data	Region	Sample Size	Citations	Focus	Key Findings Related to this Study
(Petrie, 2022)	Finland	22 begineer programmers	34	Tools	The Sonic Pi programming platform significantly reduced programming anxiety for all students and eliminated initial anxiety gaps between experienced and novice programmers.
Demir (2022)	Japan	87 university students (61 males and 26 females)	30	Pedagogy / Tools	Using Scratch with a blend of theory and practice improves programming outcomes and reduces anxiety.
Forrester et al. (2022)	United States	362 undergraduate students	1	Cause/ Tools	In an undergraduate R class, women showed higher programming anxiety than men, but novices benefited most from frequent tool support.
Olipas, et al. (2021)	Philippines	348 students enrolled in the Computer Programming 2 course	3	Cause/ Impact	A negative correlation was found between student performance and programming anxiety.
Olipas and Luciano (2020)	Philippines	120 sophomore Information Technology students	14	Cause/ Pedagogy	Gender and high school types affect programming anxiety. Regular practice reduces anxiety, while countdown timers increase it, especially for students unused to timed tasks.
Figueroa and Amoloza (2015)	Japan	141 sophomores	4	Tools	An online interactive platform in an introductory programming course effectively reduces anxiety in non-computer science students.
Jiang et al. (2020)	China	93 non-computer major students	12	Pedagogy	Technology-enhanced teaching strategies, like Rain Classroom, reduce programming anxiety, with extensive practice and out-of-class study improving learning outcomes.
Owolabi et al. (2014)	Nigeria	160 Computer/Mathematics students	58	Cause	Age is negatively correlated with programming anxiety; older students experience less anxiety.
Orehovacki et al. (2012)	Croatia	89 sophomores	11	Tools	Programming anxiety impacts novice acceptance of tools like verifiers, making it important to develop user-friendly environments that reduce anxiety.

Author, data	Region	Sample Size	Citations	Focus	Key Findings Related to this Study
Connolly et al. (2009)	Ireland	86 first-year computing undergraduate students	91	Cause/ Impact	Programming anxiety results from novices misjudging their abilities, worsened by programming's abstract nature. Strategic learning and a supportive environment can alleviate it.
Connolly et al. (2007)	Ireland	86 first-year computing undergraduate students	27	Cause/ Impact	The complexity and abstraction of programming, combined with a mismatch between student needs and resources, trigger anxiety, which negatively affects performance.
Choo and Cheung (1991)	Macau, China	200 Computer Science students	13	Assessment	Development of the Computer Programming Anxiety Scale.

The distribution of this research by region is relatively diverse, with particularly notable contributions from countries in Asia, Europe, and North America. Specifically, Japan and China have the most publications, with five and four articles, indicating a significant academic interest in programming anxiety in these regions. Regarding citation impact, Ireland comes out with an average of 59 citations per piece, highlighting the considerable influence of its few contributions. Other areas, such as Nigeria and Finland, also have a high average number of citations per article, despite their low overall publication rate.

Figures 4 and 5 further illustrate the distribution of articles in terms of publication year and database. The annual trend in the number of research articles published on programming anxiety in Figure 4 shows a rise after 2022. This indicates a growing academic interest in this field and the expansion of such investigations.

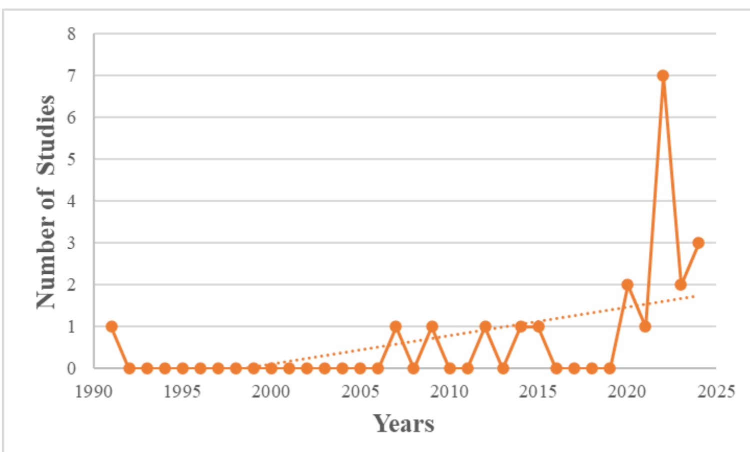

Figure 4. Publication year distribution

To determine the distribution of articles across different academic databases, a specific counting method was used to avoid duplication. If an article first appeared in an earlier database on the list (e.g., Springer), it was not recounted if it appeared in subsequent databases (e.g., WOS). This approach highlights the unique contributions of each database while acknowledging overlaps, particularly between WOS and Scopus. Figure 5 shows that Springer (34%) and IEEE (24%) dominate the literature, indicating that most empirical studies on programming anxiety appear in these repositories. This concentration guides future research toward the most influential sources in this field.

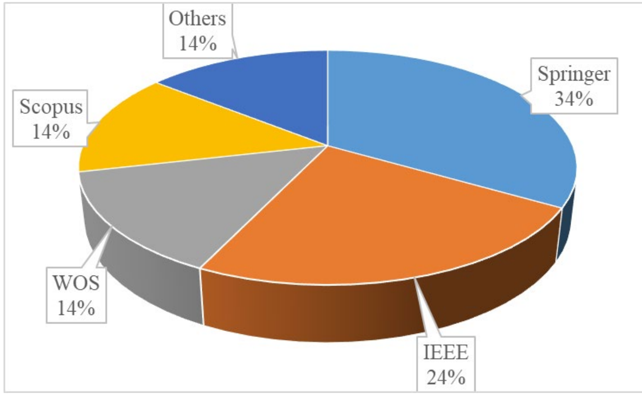

Figure 5. Database distribution

RQ1: What are the Causes of Programming Anxiety Among College Students?

Of the 21 primary studies, 2 (9.5%) identified intrinsic causes, such as cognitive load and negative self-assessment. In contrast, 10 (48%) provided empirical support for extrinsic influences, such as learning environments, tool characteristics, and demographic variables (see Figure 6). Thus, factors contributing to programming anxiety can be broadly categorized into intrinsic and extrinsic factors, representing the authors' thematic synthesis of the primary studies rather than an existing classification used by any source. Table 4 summarizes the factors that influence programming anxiety.

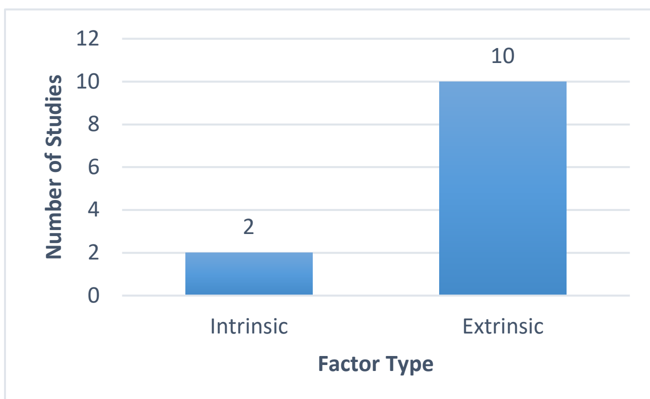

Figure 6. Distribution of studies by factor type

Intrinsic factors

Intrinsic causes of programming anxiety can be attributed to students' incorrect self-assessment of their ability to learn computer programming, combined with the abstract structure of programming languages, which increases students' fear of programming (Connolly et al., 2009). More specifically, students often struggle to develop effective mental models—conceptual frameworks that include skills such as debugging, understanding data structures, and algorithms. This increases their cognitive load, making it challenging for them to complete programming tasks (Rosenberg-Kima et al., 2022). Additionally, the non-negotiable nature of syntax errors and the continuous feedback from compilers, which often result in frustrating error messages, further exacerbate programming anxiety (Forrester et al., 2022; Rosenberg-Kima et al., 2022).

Extrinsic factors

Demographic characteristics such as gender and age play significant roles in influencing programming anxiety. Studies indicate that females have higher levels of programming anxiety than males

(Olipas & Luciano, 2020; Yildirim & Ozdener, 2022). Aside from gender, age also influences programming anxiety among students, with older learners experiencing it at lower levels (Owolabi et al., 2014).

Academic background is another key factor. Math achievement has been consistently linked to programming anxiety, with students who perform well in mathematics usually displaying lower levels of anxiety (Nolan & Bergin, 2016; Owolabi et al., 2014). Meanwhile, Forrester et al. (2022) noted that computer anxiety may overlap with math anxiety, subsequently triggering programming anxiety. Furthermore, being a particular type of high school graduate also affects programming anxiety, with students graduating from public high schools having higher programming anxiety than those graduating from private high schools (Olipas & Luciano, 2020). Programming language selection is also crucial, as choosing a proper programming language can effectively reduce programming anxiety (Demir, 2022).

Environmental factors, including computer configuration and the choice of IDE, further contribute to programming anxiety. IDE is a collection of processes and tools used for software development. It facilitates the use of numerous tools on one platform, which simplifies the software development process (Althar & Samanta, 2021; Chopra, 2017). Local IDEs generally take up more memory and require higher computer configurations (Mesihovic et al., 2020). When performing programming tasks in an IDE, a responsive computer during programming tasks can alleviate or, at the very least, not exacerbate anxiety levels (Nolan & Bergin, 2016).

Numerous educational programming environments have been developed to cater to students of various age groups. As a result, Lin and Weintrop (2021) build upon Weintrop and Wilensky's (2015) concept of modality and categorize programming languages and environments into two main groups based on the presentation of code and the method of input, namely block-based (or visual) and text-based. In a block-based programming environment (BBE), programming consists of dragging visual blocks into a script area and adhering to syntax rules to construct a program (Noone & Mooney, 2018), which can alleviate anxiety by abstracting away traditional code structure (Paredes-Velasco et al., 2022). Additionally, Unal and Topu (2021) indicate no significant difference in programming anxiety between students using a BBE and those using a traditional text-based programming environment (TBE), which introduces students to syntactic, semantic, and type errors from the outset, requiring them to write plain-text scripts and follow exact conventions without visual aids (Saygıner & Tüzün, 2023).

Table 4. Factors contributing to programming anxiety

Aspect	Category	Factors	Reference(s)
Intrinsic Characteristics of Programming Learning			(Connolly et al., 2009; Forrester et al., 2022)
Extrinsic	Demographic Characteristics	Gender	(Olipas et al., 2021; Yildirim & Ozdener, 2022)
		Age	(Owolabi et al., 2014)
	Academic Background	Math achievement	(Forrester et al., 2022; Owolabi et al., 2014)
		Programming language	(Demir, 2022)
		Computer anxiety	(Forrester et al., 2022)
		Type of high school graduated	(Olipas & Luciano, 2020)
	Learning Environment	IDE selection	(de Siqueira et al., 2022; Forrester et al., 2022; Paredes-Velasco et al., 2022; Petrie, 2022; Unal & Topu, 2021)
	Pedagogical Approaches	Instructional methods	(Demir, 2022; de Siqueira et al., 2022; Forrester et al., 2022; Petrie, 2022)

Specific coding tools, such as the block-based fableBlocks environment and the music-integrated Sonic Pi programming platform, have been shown to reduce programming anxiety through innovative teaching environments (de Siqueira et al., 2022; Petrie, 2022). Additionally, seeking tool support more frequently can effectively alleviate student programming anxiety (Forrester et al., 2022).

Finally, **pedagogical approaches** can also influence programming anxiety. Educators can break the negative cycle in which some students fall by developing learning strategies and creating encouraging and supportive environments that reduce programming anxiety (Ayersman & Michael Reed, 1995; Connolly et al., 2009). Collaborative teaching strategies, in particular, have been found to effectively alleviate programming anxiety and create more positive learning experiences for students (Forrester et al., 2022).

RQ2: WHAT ARE THE EFFECTS OF PROGRAMMING ANXIETY ON COLLEGE STUDENTS?

Current research on the effects of programming anxiety has focused on how anxiety affects programming performance and retention. Five major studies have directly addressed this issue, with four (80%) suggesting that anxiety negatively affects academic performance, and one (20%) reporting no significant difference between anxiety and academic performance. Olipas et al. (2021) found a significant negative correlation between anxiety levels and student performance, suggesting that higher levels of anxiety were associated with lower performance. Connolly et al. (2007) showed that the complexity and abstraction of programming exacerbate anxiety, which in turn affects task completion. Similarly, Connolly et al. (2009) linked increased anxiety to decreased course retention, demonstrating that strategic learning support mitigates these effects. Durak and Bulut (2024b) observed that programming anxiety predominantly affects high-achieving students. Although their peer study (Durak & Bulut, 2024a) utilized categorical achievement bands, they did not find differences in overall anxiety among students at different achievement levels. Differences between these findings likely reflect differences in analytic methodology, but the overwhelming evidence suggests that increased programming anxiety typically affects student achievement and retention.

RQ3: WHAT ARE THE TOOLS FOR ASSESSING PROGRAMMING ANXIETY AMONG COLLEGE STUDENTS?

Several programming anxiety measurement tools have been developed and refined over time. These tools vary in the dimensions assessed, number of items, and target population. Table 5 provides a comparative overview of the four main tools used to assess programming anxiety. More specifically, the Computer Programming Anxiety Scale (Choo & Cheung, 1991) is the foundational measure, consisting of 19 items across three dimensions: error responses, influence of significant others, and self-confidence, and has demonstrated validity among novice learners. Building on this work, the Programming Anxiety Scale (Yildirim & Ozdener, 2022) offers a more streamlined, 11-item version tailored to students with prior programming experience; it focuses on peer influence and self-confidence to capture anxiety in more advanced cohorts.

In contrast, the Programming Anxiety Survey developed by Figueroa and Amoloza (2015) targeted non-computer science college students and consisted of 6 items. The instrument explores the positive and negative effects of programming-related anxiety but lacks validation evidence. Similarly, the Computer Programming Anxiety Questionnaire (Connolly et al., 2009) measures anxiety among students by focusing on their acquisition of initial computing skills, sense of control, and computer self-concept. Although the instrument, with 15 items, is comprehensive, its validity has not been demonstrated.

These tools provide distinct lenses through which programming anxiety can be understood. Firstly, the tools developed by Yildirim and Ozdener (2022) and Choo and Cheung (1991) measure both state and trait anxiety, recognizing that while programming anxiety primarily manifests as state anxiety, an underlying trait anxiety can significantly influence it. Moreover, the Programming Anxiety

Survey developed by Figueroa and Amoloza (2015) distinguishes between the positive and negative impacts of anxiety, emphasizing how both programming triumphs and obstacles can notably affect anxiety levels. Finally, the Computer Programming Anxiety Questionnaire developed by Connolly et al. (2009) focuses on basic programming skills and personal control as pivotal elements that shape programming anxiety levels.

Table 5. Tools for assessing programming anxiety

Tool Name	Tool Dimensions	Tool Items	Target Population
Programming Anxiety Scale (Yildirim & Ozdener, 2022)	-Classmates -Self-confidence	11	Students with some programming experience
Programming Anxiety Survey (Figueroa & Amoloza, 2015)	-Positive contribution -Negative contribution	6	Non-computer science students
Computer Programming Anxiety Questionnaire (Connolly et al., 2009)	-Gaining initial computing skills -Sense of control -Computer self-concept -State of anxiety in computer situations	15	Computer Science majors
Computer Programming Anxiety Scale (Choo & Cheung, 1991)	-Errors -Significant others -Confidence	19	Students without prior programming experience

In summary, these tools capture the multifaceted nature of programming anxiety, considering peer influence, confidence, errors, computer skill acquisition, and sense of control. However, the focus remains fragmented, emphasizing particular aspects rather than establishing a unified framework for understanding how learning environments and personal traits influence programming anxiety. Thus, developing a unified programming anxiety scale that integrates self-efficacy, emotional state, peer influence, and error perception would enhance the comprehensiveness of anxiety assessment.

RQ4: What Coping Mechanisms and Educational Strategies Effectively Alleviate Programming Anxiety?

Several studies have examined approaches to alleviating programming anxiety, particularly using different development environments and pedagogical strategies.

Proper development environment selection

The primary studies have investigated how employing various development environments impacts learner programming anxiety. de Siqueira et al. (2022) combined storytelling with a tangible BBE to reduce programming anxiety among college students who had never programmed before. The study revealed a significant reduction in programming anxiety for students using fableBlocks but an increase in programming anxiety for students using Google Blockly, indicating that the type of development environment plays a crucial role in learners' emotional responses to programming. Similarly, Demir (2022) found that the use of a block-based programming language (Scratch) and a combination of theoretical and practical teaching methodologies could help improve learning outcomes in programming and alleviate programming anxiety. This suggests that a balanced approach combining hands-on practice with a programming environment can help learners cope with programming-related challenges.

An alternative approach combining programming and creative arts was investigated by Petrie (2022). In this study, programming learning was combined with music making by teaching programming using the Sonic Pi programming platform. The results showed that students with prior programming experience had significantly lower levels of programming anxiety at the start of the unit. Despite these initial lower levels of anxiety, both novice and experienced students showed substantially decreased anxiety levels by the end of the experiment. This suggests that engaging learners in creative and nontraditional programming activities can foster positive attitudes and alleviate anxiety, regardless of their prior experience.

In addition to these approaches, Forrester et al. (2022) investigated the correlation between programming anxiety and gender in an undergraduate R language classroom and the corresponding mitigation strategies. The findings revealed that programming anxiety among students was more substantially mitigated when instrumental support was sought more frequently.

Pedagogical approaches

Pedagogy has also been used to alleviate programming anxiety. Shang (2024) indicated that traditional teaching methods lack flexibility and are more likely to make students anxious about programming learning. In contrast, teaching methods incorporating collaborative elements can effectively mitigate programming anxiety. For example, Cheng et al. (2022) compared the efficacy of two pedagogical approaches—content-based knowledge awareness and team-based learning—in alleviating programming anxiety during an online programming course. While the former approach yielded unexpected outcomes, team-based learning was notably successful in reducing programming anxiety, improving programming skills, and managing cognitive load. Furthermore, collaborative learning environments in programming education have demonstrated the potential to reduce anxiety in learners and foster problem-solving skills development (Erdei et al., 2017; Jiang et al., 2020).

DISCUSSION

This study analyzed 21 academic papers and showed that annual publications on programming anxiety increased from an average of less than 1 per year during the 2010-2021 period to 4 per year during the 2022-2024 period (see Figure 4). The growth in the number of studies can be attributed to the demand for technical skills and the expansion of programming into different fields involving diverse learners (Yildirim & Ozdener, 2022). As programming becomes an essential part of various fields, it has become increasingly important to understand the psychological obstacles that learners face (Zhang, 2017). While earlier studies focused on the factors and impact of programming anxiety regarding programming learning, more recent studies have shifted toward mitigation strategies.

Countries like Japan, China, the United States, and the Philippines are leading contributors to research in this field, reflecting the widespread international interest in programming anxiety. Many of these studies were published in highly reputable databases such as Springer and IEEE and have been indexed in WOS or Scopus. The results of these studies confirm that programming anxiety is a prevalent challenge that requires international and interdisciplinary collaboration to address it effectively.

FACTORS IMPACTING ANXIETY AND MITIGATION

This review summarizes five primary factors that contribute to programming anxiety: learning environments, instructional methods, the inherent nature of programming, demographic characteristics (such as gender and age), and academic background (including math achievement, choice of programming language, computer anxiety, and high school type). Importantly, the first two factors are changeable, while the last two are largely immutable.

Current research on the impact of programming anxiety focuses on programming performance, with most studies suggesting the negative character of this relationship. However, education should not be limited to knowledge acquisition; it should also focus on students' mental health and broader skills development (Bennedsen & Caspersen, 2019; Lai & Wong, 2022; Melcer & Isbister, 2018). Interventions that address the multifaceted nature of anxiety, including the selection of user-friendly IDEs and the adoption of collaborative teaching methods, can significantly reduce programming anxiety.

A more comprehensive understanding of how programming anxiety is affected by particular factors could provide evidence for strategies that would better support student learning. Future research referring to anxiety research in other disciplines (Archambault et al., 2022; Liu, 1997; Wu et al., 2022), employing longitudinal designs to explore how programming anxiety interacts with self-efficacy, engagement, and problem-solving skills over time, thereby providing evidence for strategies that more comprehensively support student learning.

Effectiveness of Tools for Measuring Programming Anxiety

The instruments used to measure programming anxiety vary in their target populations. For instance, the Programming Anxiety Scale, designed by Yildirim and Ozdener (2022), specifically addresses students with programming experience, while other scales are generally intended for those without prior programming exposure. The four tools reviewed assess programming anxiety from various angles and lack standardized criteria, highlighting the diversity yet inconsistency of the current approaches. Despite advancements being made, the limited cross-validation of these instruments suggests the need for more rigorous, standardized measurement tools that can be applied to different settings and populations.

Despite the existence of four distinct instruments to assess programming anxiety (Table 5), their different measurement properties and target populations hinder cross-study comparisons. The Programming Anxiety Inventory (Yildirim & Ozdener, 2022) is more suitably applied to experienced learners. Moreover, the Programming Anxiety Survey (Figueroa & Amoloza, 2015) remains unvalidated, and its limited number of items (6) may lack the sensitivity needed to detect subtle anxiety changes. The Computer Programming Anxiety Questionnaire (Connolly et al., 2009) and the Programming Anxiety Inventory (Choo & Cheung, 1991) encompass a wider range of dimensions but are tailored to groups with differing programming experiences.

Instrument rigor affects the reported results. Studies using well-validated scales produce more consistent and interpretable results, whereas using untested questionnaires tends to produce ambiguous results (Svensson, 2011). In addition, the overlap of measurement dimensions, such as self-efficacy, emotional state, misperceptions, and peer influence, suggests that a unified instrument can simplify assessment and improve comparability. Standardizing measurement tools using a single, reliable scale can contribute to more reliable tracking of programming anxiety across educational contexts. This provides an opportunity for future research to refine these tools to ensure an accurate capture of the nature of programming anxiety in a variety of ways.

Mitigation Strategies for Programming Anxiety

The findings suggest the great promise offered by strategies that aim to reduce programming anxiety through learning environments and pedagogical approaches. The effectiveness of creative programming platforms such as BBEs, storytelling platforms, and Sonic Pi suggests that engaging students in novel and interactive ways can mitigate anxiety. These findings reinforce the idea that nontraditional, collaborative, and flexible learning environments can reduce the anxiety associated with programming tasks (Demir, 2022; Forrester et al., 2022; Jiang et al., 2020).

Pedagogical strategies that emphasize collaboration and teamwork have also proven particularly effective. This suggests that anxiety stems not only from technical challenges but also from the isolation that students often experience while learning to program. Collaborative learning approaches, encouraging peer-supported problem-solving, could mitigate these emotional challenges, fostering more supportive and positive learning experiences (Cheng et al., 2022).

To translate these findings into practice, programming instructors should prioritize IDEs that are user-friendly and those with low technical barriers. Additionally, embedding structured pair programming or group coding exercises further reduces student isolation and builds confidence. Implementing a formative anxiety self-assessment during the course progression allows for early identification of students in need of support, which can lead to targeted interventions such as instructor-led or scaffolded coding tutorials. Moreover, providing teachers with specialized training on instructional practices related to programming anxiety and designing lessons with gradual complexity and reflective prompts can help students gradually develop adaptability and self-efficacy as they progress through the programming process. Future investigations should compare the effects of different types of IDEs (e.g., BBE and TBE) on students' anxiety levels and qualitative interviews with learners from underrepresented backgrounds to identify which instructional combinations yield the greatest anxiety reduction and learning gains.

CONCLUSION

This review was guided by four research questions that led to the following findings: (1) intrinsic factors, such as cognitive load and negative self-assessment, and extrinsic factors, including demographics, academic background, learning environment, and instructional methodology, all contribute to programming anxiety; (2) higher levels of anxiety are associated with lower grades and retention in programming courses; (3) assessment instruments vary widely in terms of dimensionality, items number, and validation, indicating a lack of standardization; and (4) mitigation strategies involving supportive technology and collaborative pedagogy hold great promise for reducing programming anxiety and improving outcomes.

The findings emphasize the importance of using holistic teaching strategies that not only convey programming knowledge but also address the emotional well-being of students to mitigate anxiety. To put this approach into practice, higher education institutions could conduct workshops on anxiety evaluation and mitigation strategies for programming teachers. Teachers are encouraged to adopt collaborative and flexible learning models to create supportive learning environments that meet the emotional needs of their students, as well as to monitor their students' programming anxiety levels and take early action when needed. The review also emphasizes the significance of selecting an IDE that is appropriate for the backgrounds and learning requirements of students because the choice of IDE can directly affect their programming experience and anxiety levels.

We further encourage educators to integrate these evidence-based strategies into their curricula. We also urge researchers to address identified gaps in this field, such as conducting longitudinal studies, comparative assessments of IDEs, qualitative interviews, and cross-cultural validation.

Overall, this review contributes to the field by synthesizing current research on programming anxiety, identifying key gaps, and providing practical recommendations for educators and researchers. It provides the first comprehensive systematic review of programming anxiety in higher education, thus filling an important gap in literature.

LIMITATIONS OF THE STUDY

Despite its strengths, this review has three main limitations. First, this review only considered English-language publications from six major databases, which may have omitted relevant non-English-language studies and studies from less-indexed sources. Second, although two authors independently screened titles, abstracts, and full texts, formal inter-rater reliability statistics (e.g., Cohen's κ) were not computed, which may have affected the consistency of study selection. Finally, all theme coding was done independently by the author team and was not externally validated, possibly leading to interpretive bias during the theme identification and categorization process.

Declaration of Competing Interest

The authors declare no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding

This work was supported by the UTM-NGA Research Grant under Grant No. R.J130000.7353.4B763.

References

Althar, R. R., & Samanta, D. (2021). Building intelligent integrated development environment for IoT in the context of statistical modeling for software source code. In R. Kumar, R. Sharma, & P. K. Pattnaik (Eds.) Multimedia Technologies in the Internet of Things Environment. Studies in Big Data, vol 79.In *Multimedia Technologies in the Internet of Things Environment. Studies in Big Data 79*, 95–115. Springer, Singapore. https://doi.org/10.1007/978-981-15-7965-3_7

Archambault, I., Pascal, S., Olivier, E., Dupéré, V., Janosz, M., Parent, S., & Pagani, L. S. (2022). Examining the contribution of student anxiety and opposition-defiance to the internal dynamics of affective, cognitive and behavioural engagement in Math. *Learning and Instruction*, *79*(101593). https://doi.org/10.1016/j.learninstruc.2022.101593

Ayersman, D. J., & Michael Reed, W. (1995). Effects of learning styles, programming, and gender on computer anxiety. *Journal of Research on Computing in Education*, *28*(2), 148–161. https://doi.org/10.1080/08886504.1995.10782157

Bennedsen, J., & Caspersen, M. E. (2007). Failure rates in introductory programming. *ACM SIGCSE Bulletin*, *39*(2), 32–36. https://doi.org/10.1145/1272848.1272879

Bennedsen, J., & Caspersen, M. E. (2019). Failure rates in introductory programming—12 years later. *ACM Inroads*, *10(2)*, 30–36. https://doi.org/10.1145/3324888

Cawthorne, L. (2021). Invited viewpoint: Teaching programming to students in physical sciences and engineering. *Journal of Materials Science*, *56*(29), 16183–16194. https://doi.org/10.1007/s10853-021-06368-1

Chang, L. C., Lin, H. R., & Lin, J. W. (2024). Learning motivation, outcomes, and anxiety in programming courses—A computational thinking–centered method. *Education and Information Technologies*, *29*(1), 545–569. https://doi.org/10.1007/s10639-023-12313-3

Charles, T., & Gwilliam, C. (2023). The effect of automated error message feedback on undergraduate physics students learning python: Reducing anxiety and building confidence. *Journal for STEM Education Research*, *6*(2), 326–357. https://doi.org/10.1007/s41979-022-00084-4

Cheng, Y. P., Shen, P. D., Hung, M. L., Tsai, C. W., Lin, C.-H., & Hsu, L. C. (2022). Applying online content-based knowledge awareness and team learning to develop students' programming skills, reduce their anxiety, and regulate cognitive load in a cloud classroom. *Universal Access in the Information Society*, *21*(2), 557–572. https://doi.org/10.1007/s10209-020-00789-6

Choo, M. L., & Cheung, K. C. (1991). On meaningful measurement: Junior college pupils' anxiety towards computer programming. *Journal of Educational Technology Systems*, *19*(4), 327–343. https://doi.org/10.2190/3LM5-W4VN-WAJM-JL2E

Chopra, R. (2017). *C programming: A self-teaching introduction*. Mercury Learning and Information.

Connolly, C., Murphy, E., & Moore, S. (2007). Second chance learners, supporting adults learning computer programming. *International Conference on Engineering Education–ICEE*, 3-7. https://icee2007.dei.uc.pt/proceedings/papers/407.pdf

Connolly, C., Murphy, E., & Moore, S. (2009). Programming anxiety amongst computing students—A key in the retention debate? *IEEE Transactions on Education*, *52*(1), 52–56. https://doi.org/10.1109/TE.2008.917193

Courraud, J. (2014). Zotero: A free and open-source reference manager. *Medical Writing*, *23*(1), 46–48. https://doi.org/10.1179/2047480614Z.000000000190

Demir, F. (2022). The effect of different usage of the educational programming language in programming education on the programming anxiety and achievement. *Education and Information Technologies*, *27*(3), 4171–4194. https://doi.org/10.1007/s10639-021-10750-6

de Siqueira, A. G., Feijóo-García, P., Carnell, S., Palmeira, E., & Maxim, A. (2022). fableBlocks: Toward mitigating programming anxiety with storytelling-based tangible block programming environments. *2022 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 1–4. https://doi.org/10.1109/VL/HCC53370.2022.9833102

Dirzyte, A., Vijaikis, A., Perminas, A., & Rimasiute-Knabikiene, R. (2021). Associations between depression, anxiety, fatigue, and learning motivating factors in e-learning-based computer programming education. *International Journal of Environmental Research and Public Health*, *18*(17), 9158. https://doi.org/10.3390/ijerph18179158

Durak, A., & Bulut, V. (2024a). Classification and prediction-based machine learning algorithms to predict students' low and high programming performance. *Computer Applications in Engineering Education*, *32*(1). https://doi.org/10.1002/cae.22679

Durak, A., & Bulut, V. (2024b). Predicting low and high student performance in programming education using PLS-SEM Algorithms. *Technology, Knowledge and Learning*. https://doi.org/10.1007/s10758-024-09737-2

Erdei, R., Springer, J. A., & Whittinghill, D. M. (2017). An impact comparison of two instructional scaffolding strategies employed in our programming laboratories: Employment of a supplemental teaching assistant versus employment of the pair programming methodology. *2017 IEEE Frontiers in Education Conference (FIE)*, 1–6. https://doi.org/10.1109/FIE.2017.8190650

Fernalld, K., Oconnor, T. J., Sudhakaran, S., & Nur, N. (2023). Lightweight symphony: towards reducing computer science student anxiety with standardized docker environments. *Proceedings of the 24th Annual Conference on Information Technology Education* ,15-21. https://doi.org/10.1145/3585059.3611432

Figueiredo, J., & Garcia-Penalvo, F. J. (2018). Building skills in introductory programming. *Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality*, 46–50. https://doi.org/10.1145/3284179.3284190

Figueroa, R. B., Jr., & Amoloza, E. M. (2015). Addressing programming anxiety among non-computer science distance learners: A UPOU case study. *International Journal for Educational Media and Technology*, *9*(1). https://ijemt.org/index.php/journal/article/view/247

Forrester, C., Schwikert, S., Foster, J., & Corwin, L. (2022). Undergraduate R programming anxiety in ecology: Persistent gender gaps and coping strategies. *CBE—Life Sciences Education*, *21*(2), ar29. https://doi.org/10.1187/cbe.21-05-0133

Huang, S. B., Jeng, Y. L., & Lai, C. F. (2021). Note-taking learning system: The use of the learning style theory and the peer learning method on computer programming course. *Journal of Educational Computing Research*, *59*(5), 870–895. https://doi.org/10.1177/0735633120985235

Inventado, P. S. (2019). Promoting Mastery Learning in an Introductory Programming Course. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 1285–1285. https://doi.org/10.1145/3287324.3293780

Jiang, Y., Zhao, Z., Wang, L., & Hu, S. (2020). Research on the influence of technology-enhanced interactive strategies on programming learning. *2020 15th International Conference on Computer Science & Education (ICCSE)*, 693–697. https://doi.org/10.1109/ICCSE49874.2020.9201627

Lai, X., & Wong, G. K. (2022). Collaborative versus individual problem solving in computational thinking through programming: A meta-analysis. *British Journal of Educational Technology*, *53*(1), 150–170. https://doi.org/10.1111/bjet.13157

Li, G., Li, Z., Wu, X., & Zhen, R. (2022). Relations between class competition and primary school students' academic achievement: Learning anxiety and learning engagement as mediators. *Frontiers in Psychology*, *13*. https://doi.org/10.3389/fpsyg.2022.775213

Lin, Y., & Weintrop, D. (2021). The landscape of block-based programming: Characteristics of block-based environments and how they support the transition to text-based programming. *Journal of Computer Languages*, *67*, 101075. https://doi.org/10.1016/j.cola.2021.101075

Liu, M. (1997). The effects of hypercard programming on teacher education students' problem-solving ability and computer anxiety. *Journal of Research on Computing in Education*, *29*(3), 248–262. https://doi.org/10.1080/08886504.1997.10782197

Melcer, E. F., & Isbister, K. (2018). Bots & (main)frames: Exploring the impact of tangible blocks and collaborative play in an educational programming game. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 1–14. https://doi.org/10.1145/3173574.3173840

Mesihovic, M., Ljubovic, V., & Muharemovic, I. (2020). Using WebIDE as a distance learning tool for high school programming. *2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO)*, 893–898. https://www.webofscience.com/wos/woscc/summary/1e48e890-d1e8-43b5-a43b-0c3bd921bac7-981cd014/relevance/1

Nolan, K., & Bergin, S. (2016). The role of anxiety when learning to program: A Systematic review of the literature. *Proceedings of the 16th Koli Calling International Conference on Computing Education Research*, 61–70. https://doi.org/10.1145/2999541.2999557

Noone, M., & Mooney, A. (2018). Visual and textual programming languages: A systematic review of the literature. *Journal of Computers in Education*, *5*(2), 149–174. https://doi.org/10.1007/s40692-018-0101-5

Olipas, C. N. P., Leona, R. F., Villegas, A. C. A., Cunanan, A. I., Jr., & Javate, C. L. P. (2021). The academic performance and the computer programming anxiety of BSIT students: A basis for instructional strategy improvement. *International Journal of Advanced Engineering, Management and Science*, *7*(6), 125–129. https://doi.org/10.22161/ijaems.76.15

Olipas, C. N. P., & Luciano, R. G. (2020). Understanding the impact of using countdown timer on the academic motivation and computer programming anxiety of IT students: The case of a state university in the Philippines. *International Journal of Scientific & Technology Research*, *9*(03), 235–240.

Orehovacki, T., Radosevic, D., & Konecki, M. (2012). Acceptance of verificator by information science students. In V. LuzarStiffler, I. Jarec, & Z. Bekic (Eds.), *Proceedings of the ITI 2012 34th International Conference on Information Technology Interfaces (ITI)* (pp. 223–230). IEEE. https://doi.org/10.2498/iti.2012.0451

Owolabi, J., Olanipekun, P., & Iwerima, J. (2014). Mathematics ability and anxiety, computer and programming anxieties, age and gender as determinants of achievement in basic programming. *GSTF Journal on Computing (JoC)*, *3*(4), 47. https://doi.org/10.7603/s40601-013-0047-4

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., … Moher, D. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *Journal of Clinical Epidemiology*, *134*, 178–189. https://doi.org/10.1016/j.jclinepi.2021.03.001

Paredes-Velasco, M., Lozano-Osorio, I., Perez-Marin, D., & Santacruz-Valencia, L. P. (2022). A case study on learning visual programing with TutoApp for composition of tutorials: An approach for learning by teaching. *IEEE Transactions on Learning Technologies*, 1–16. https://doi.org/10.1109/TLT.2022.3226122

Pears, A., & Seidman, S. (2007). A survey of literature on the teaching of introductory programming. *Working Group Reports on ITiCSE on Innovation and Technology in Computer Science Education*. https://doi.org/10.1145/1345443.1345441

Petrie, C. (2022). Programming music with Sonic Pi promotes positive attitudes for beginners. *Computers and Education*, *179*(104409). https://doi.org/10.1016/j.compedu.2021.104409

Rosenberg-Kima, R. B., Merrill, M. D., Baylor, A. L., & Johnson, T. E. (2022). Explicit instruction in the context of whole-tasks: The effectiveness of the task-centered instructional strategy in computer science education. *Educational Technology Research and Development*, *70*(5), 1627–1655. https://doi.org/10.1007/s11423-022-10143-7

Saygıner, Ş., & Tüzün, H. (2023). The effects of block-based visual and text-based programming training on students' achievement, logical thinking skills, and motivation. *Journal of Computer Assisted Learning*, *39*(2), 644–658. https://doi.org/10.1111/jcal.12771

Scott, M. J., & Ghinea, G. (2014). Measuring enrichment: The assembly and validation of an instrument to assess student self-beliefs in CS1. *Proceedings of the Tenth Annual Conference on International Computing Education Research*, 123–130. https://doi.org/10.1145/2632320.2632350

Shang, H. (2024). A study of English learning anxiety regulation strategies based on the deep learning model. *Applied Mathematics and Nonlinear Sciences*, *9*(1), 6 https://doi.org/10.2478/amns.2023.1.00381

Spielberger, C. D. (2013). *Anxiety: Current trends in theory and research*. Elsevier.

Suren, N., & Kandemir, M. A. (2020). The effects of mathematics anxiety and motivation on students' mathematics achievement. *International Journal of Education in Mathematics Science and Technology*, *8*(3), 190–218. https://doi.org/10.46328/ijemst.v8i3.926

Svensson, E. D. (2011). Validity of Scales. In M. Lovric (Ed.), *International Encyclopedia of Statistical Science* (pp. 1637–1639). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-04898-2_98

Unal, A., & Topu, F. B. (2021). Effects of teaching a computer programming language via hybrid interface on anxiety, cognitive load level and achievement of high school students. *Education and Information Technologies*, *26*(5), 5291–5309. https://doi.org/10.1007/s10639-021-10536-w

Wang, G., Zhao, H., Guo, Y., & Li, M. (2019). Integration of flipped classroom and problem based learning model and its implementation in university programming course. *2019 14th International Conference on Computer Science & Education (ICCSE)*, 606–610. https://doi.org/10.1109/ICCSE.2019.8845525

Weintrop, D., & Wilensky, U. (2015). Using commutative assessments to compare conceptual understanding in blocks-based and text-based programs. *Proceedings of the Eleventh Annual International Conference on International Computing Education Research*, 101–110. https://doi.org/10.1145/2787622.2787721

Wu, X., Yang, H. M., Liu, J. X., & Liu, Z. Y. (2022). English use anxiety, motivation, self-efficacy, and their predictive effects on Chinese top university students' English achievements. *Frontiers in Psychology*, *13*(953600). https://doi.org/10.3389/fpsyg.2022.953600

Xu, F., & Correia, A. P. (2023). Adopting distributed pair programming as an effective team learning activity: A systematic review. *Journal of Computing in Higher Education*, 36(2). https://doi.org/10.1007/s12528-023-09356-3

Yildirim, O. G., & Ozdener, N. (2022). Development and validation of the programming anxiety scale. *International Journal of Computer Science Education in Schools*, *5*(3), 17–34. https://doi.org/10.21585/ijcses.v5i3.140

Zhang, J. (2017). Psychological predicament of freshmen in the course of computer programming and its countermeasures. *Proceedings of the 2nd International Conference on Contemporary Education, Social Sciences and Humanities (ICCESSH 2017)*, 307-309. https://doi.org/10.2991/iccessh-17.2017.72

AUTHORS

Ying Pan is a PhD candidate in Educational Technology at Universiti Teknologi Malaysia (UTM). She holds both Bachelor's and Master's degrees in Computer Science and Technology, and has 3 years of experience as a software developer. Her research focuses on exploring the factors that influence programming learning, to identify effective instructional strategies and promote student success in programming education.

Jamalludin Harun is a Professor of Educational Technology at Universiti Teknologi Malaysia (UTM). He holds a Bachelor of Science in Computer with Education (Chemistry) from UTM, a Master's in Educational Media and Computers from Arizona State University, and a PhD in Educational Technology from UTM. His research focuses on problem-based learning, collaborative learning, and the integration of technology in education. He has a special interest in designing active, effective, and student-centered online learning environments. Additionally, he explores innovative teaching approaches, such as gamification, mobile learning, and social learning, to enhance online and technology-driven education.