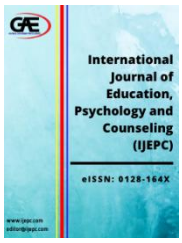
International Journal of Education, Psychology and Counseling
IJEPC
EISSN : 0128-164X
Volume 11 Issue 65 (December 2026) PP. 01-19



INTERNATIONAL JOURNAL OF EDUCATION, PSYCHOLOGY AND COUNSELLING (IJEPC)
https://gaexcellence.com/ijepc

TEACHING JAVA: A RESEARCH TREND

Mohd Rizal Razak[1], Nor Zalina Ismail[2]*, Nor 'aqilah Misman[3], Mohd Azim Zainal[4], Khairunnisa Abdul Kadir[5], Ahmad Subhi Zolkafly[6], Noor Izyan Mohamad Adnan[7], Mohamad Affendi Abdul Malek[8]

[1]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA (UiTM), Pahang Branch Raub Campus, Pahang, Malaysia
dragon_admire007@uitm.edu.my https://orcid.org/0000-0002-5052-756X
[2]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA (UiTM), Pahang Branch Raub Campus, Pahang, Malaysia
nza1601@uitm.edu.my https://orcid.org/0000-0001-7798-1941
[3]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA (UiTM), Pahang Branch Raub Campus, Pahang, Malaysia
noraqilahmisman@uitm.edu.my https://orcid.org/0009-0007-2119-3517
[4]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA (UiTM), Pahang Branch Raub Campus, Pahang, Malaysia
azim90@uitm.edu.my https://orcid.org/0009-0006-3630-3523
[5]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA (UiTM), Pahang Branch Raub Campus, Pahang, Malaysia
khairunnisa.kadir@uitm.edu.my https://orcid.org/0009-0000-5488-7493
[6]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA (UiTM), Pahang Branch Raub Campus, Pahang, Malaysia
subhizolkafly@uitm.edu.my https://orcid.org/0009-0001-3310-7985
[7]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA (UiTM), Pahang Branch Jengka Campus, Pahang, Malaysia
noorizyan@uitm.edu.my https://orcid.org/0000-0002-9938-6905
[8]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA (UiTM), Pahang Branch Raub Campus, Pahang, Malaysia
maffendi@uitm.edu.my https://orcid.org/0009-0001-3869-2546
*Corresponding Author

International Journal of Education, Psychology and Counseling
IJEPC
EISSN : 0128-164X
Volume 11 Issue 65 (December 2026) PP. 01-19

Article Info:

Article history:
Received date: 28.09.2026
Revised date: 13.10.2026
Accepted date: 02.11.2026
Published date: 01.12.2026

To cite this document:

Razak, M. R., Ismail, M. Z., Misman, N., Zainal, M. A., Kadir, K. A., Zolkafly, A. S., Adnan, N. I. M., & Malek, M. A. A. (2026). Teaching Java: A Research Trend. *International Journal of Education, Psychology and Counselling, 11*(65), 01-19.

Abstract:

The rise of the significance of programming education has sparked academic interest in the best methods of teaching Java, especially since Java is now used extensively in object-oriented programming education and in the acquisition of fundamental computational skills. Nonetheless, the intellectual growth, research tendencies, geographical dispersion, thematic organization, and patterns of collaboration in the teaching Java literature are not synthesized adequately. This paper thus seeks to offer a thorough bibliometric review of academic literature on the teaching of Java and to determine its evolution, publications that have contributed most, the most contributing countries, the prevailing research topics and patterns of international collaboration. The bibliometric analysis was performed based on 239 academic articles found in the Scopus database with the help of an advanced search strategy that included three main concepts: Java, object-oriented programming, and teaching. The retrieved records were then analyzed and refined with the help of OpenRefine, and Scopus Analyzer and VOSviewer were used to analyze the trends in publications, citation performance, country contributions, co-occurrence of keywords, and international co-authorship networks. The results indicate how the interest in teaching Java research has evolved, the most impactful articles and the most contributing countries, and the key thematic groups and future research directions in the area. The keyword and collaboration analyses also emphasize the intellectual and geographical frameworks on which research on Java education is based. On the whole, the current research offers a systematic review of the current research environment in the field of teaching Java and a rich source of knowledge to teachers, researchers, and institutions interested in determining the current research directions, the emergent themes, and the opportunities to engage in international research in the future.

DOI: 10.35631/IJEPC.1165001

Keyword:

Java, Object-Oriented Programming, Teaching

© The authors (2026). This is an Open Access article distributed under the terms of the Creative Commons Attribution (CC BY NC) (http://creativecommons.org/licenses/by-nc/4.0/), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited. For commercial re-use, please contact ijepc@gaexcllence.com.

Introduction

An object-oriented pedagogy of teaching Java involves more than simply introducing a programming language; it involves introducing a mode of thought that is conceptually challenging to many novices. Java is still the focus of most undergraduate programs because object-oriented programming is a fundamental aspect of computing education and a popular paradigm in software development (Guo and Hu, 2024; Khudhur et al., 2023; Lian et al., 2022). Simultaneously, the literature demonstrates that students have a difficult time with their initial experience with

programming, and OOP makes it even more challenging since its key concepts are abstract, highly interconnected, and can be easily lost in syntax and development environments (Calderon et al., 2023; Khudhur et al., 2023; Lian et al., 2022; Omer et al., 2021). These issues are particularly apparent in Java courses when students are expected to go beyond writing individual statements and start modeling problems with classes, objects, methods, inheritance, and encapsulation (Ahmed et al., 2026; Guo and Hu, 2024; Khudhur et al., 2023) Research synthesizing the literature on OOP finds that these challenges remain consistently present, with students struggling to understand classes, apply object orientation, understand methods, and model object-oriented relationships (Gutiérrez et al., 2022; Menolli and Strik, 2025) Research synthesizing the OOP literature identifies these issues as persistent, suggesting that the problem is not an isolated incident but structural to the way OOP is typically taught (Gutiérrez et al., 2022; Menolli and Strik, 2025) . It is because of this that a good introduction to Java pedagogy should start with the assumption that learning Java is not only a cognitive but also a practical and curricular issue but not a mere issue of syntax delivery (Tsichouridis et al., 2023; Wang et al., 2024) .

An effective object-oriented pedagogy of Java thus changes the focus of teacher-centred content delivery to learning designs that assist the students to build concepts through guided practice, interaction, and repetition. The two responses are rated as particularly significant in reviews of OOP teaching issues, including a focus on fundamental principles of programming and the introduction of the object-oriented paradigm at an early stage, as well as active learning with intrinsic rewards (Gutiérrez et al., 2022). This focus is important since the traditional approach to sequencing tends to introduce procedural programming initially and introduce OOP as an extension, which has been criticized as a pattern that perpetuates the myth that object orientation is only an added feature and not a modeling paradigm (Menolli & Strik, 2025). In practice, this implies that Java teaching should place classes and objects as basic abstractions in the foreground, and syntax in the background, with problem decomposition, design choices, and the behaviour of interacting software components (Guo and Hu, 2024; Gutiérrez et al., 2022) in the foreground and syntax in the background (Wang et al., 2024; Xing and Yanan, 2024) .Similar conclusions are made by outcome-based reforms, which note that overly abstract instruction, disconnected programming exercises, and one-shot evaluation are not very compatible with the practicality of Java and OOP instruction (Guo and Hu, 2024; Sun, 2023).

The other significant theme in the literature is that the teaching of Java OOP has representational and technological supports that render abstract concepts visible. It is repeatedly suggested to use visualization tools, analogies, UML diagrams, concept maps, and interactive environments since students tend to misinterpret the invisible connections between code structure and runtime behavior (Gutiérrez et al., 2022; Lian et al., 2022)). The concept-oriented tools seem to be especially helpful when they combine theoretical and practical knowledge instead of considering them as independent (Khudhur et al., 2023). In a classroom experiment, a conceptual representation approach increased students understanding of OOP concepts significantly and doubled the performance gains between pre-test and post-test with conventional instruction and also facilitated the identification of low-performing students earlier (Khudhur et al., 2023). Interactive Java-based tools evidence suggests the same: students with visualization-based support performed better on the assessment and demonstrated more efficient problem-solving and conceptual retention, but not all visualization interventions have demonstrated a decisive benefit over all traditional interventions (Ahmed et al., 2026; Lian et al., 2022). The systematic mapping of educational settings also indicates that successful OOP tools tend to be simple, provide interaction with objects, visualization, experimentation, informative feedback, and familiar

metaphors, but most of them are still not able to support large-scale conceptual design or active learning alone (Tsichouridis et al., 2023).

Java OOP Pedagogy is also influenced by delivery mode, diversity of learners and the design of assessment. The results of blended and collaborative environments show that learning outcomes are enhanced in courses that offer structured modules, video tutorials, discussion opportunities, rotation between lab and classroom work, and active learning with the support of ICT (M & S, 2023; Maspaeni et al., 2022). In a single study of blended learning, post-test scores improved to 80.40 and the percentage of students who met the minimum standard of achievement grew to 85% which shows that the design of delivery can significantly impact achievement in OOP courses (Maspaeni et al., 2022). The significance of delivery conditions was also emphasized in the COVID-19 era, when face-to-face teaching was more effective than virtual learning, in part due to the availability of laboratories, internet, and direct practice support that most distance learners did not have (Chimanga et al., 2021). In the case of novices with no prior programming experience, research suggests effective syllabus design, timely feedback, and various types of assessment, as well as support systems that address fear, passivity, and disproportionate preparation (Wang et al., 2024). This is also applicable to students who switch their first language to Java as a second programming language, where explicit transfer pedagogy positively influenced post-test performance on most concepts and was well-received by 88% of students as useful in preventing transfer errors (Tshukudu et al., 2021).

Collectively, the new literature conceptualizes the teaching of Java in object-oriented pedagogy as an issue of congruence between conceptual knowledge, experience, feedback, and support to learners. Constructive alignment, collaboration, and focus on motivation have been linked to student-centered approaches that have led to higher pass rates in Java OOP courses, with a 12.5% failure rate declining to 4.8% in one implementation (Haruna, 2023). Recent reforms also focus on individualized trajectories, project carriers, stepped teaching, and diversified classroom practices like scenario introduction, task-based learning, discussion, and group development to allow students to transition between basic abstractions and applied and innovative software work (Luo and Wu, 2025; Sun, 2023; Xing and Yanan, 2024). Evaluation is also changing accordingly, and automated grading and feedback systems provide near-real-time feedback and allow resubmission, but existing systems still tend to focus more on correctness than maintainability or readability (Messer et al., 2023). Introductory Java also seems to be an area where emerging AI assistance can be useful, as beginner learners write cleaner and less complex code when assisted, but again, the evidence is early and limited in scale. A lengthy preface to Teaching Java of Object-Oriented Pedagogy must thus place Java instruction not as a restricted coding drill, but as an evidence-based pedagogical venture that assists students to learn abstraction, design judgment, problem-solving skill, and long-term confidence by engaging in integrated, active, and adaptive teaching activities.

Research Question

RQ1: How has research on teaching Java evolved over time in terms of publication growth and scholarly development?
RQ2: Which publications have had the greatest scholarly impact on teaching Java, based on citation performance?
RQ3: Which countries have made the greatest contributions to teaching Java research in terms of publication productivity and citation impact?

International Journal of Education, Psychology and Counseling
IJEPC
EISSN : 0128-164X
Volume 11 Issue 65 (December 2026) PP. 01-19
RQ4: What are the dominant research themes, emerging topics, and thematic relationships in teaching Java research based on keyword co-occurrence analysis?
RQ5: What are the patterns and structures of international research collaboration in teaching Java, as revealed through country-level co-authorship networks?

Methodology

Bibliometric analysis is a systematic gathering, structuring and quantitative analysis of bibliographic data based on scholarly publications (Alves et al., 2021; Assyakur and Rosa, 2022; Verbeek et al., 2002). In addition to the descriptive indicators, including publication outlets, temporal publication patterns, and influential authors (Wu and Wu, 2017), bibliometric analysis involves sophisticated analysis methods, including document co-citation analysis, to reveal the intellectual organization and development of a research field. The systematic and intensive study of the literature necessitates an iterative and systematic process that involves the development of the right keywords, thorough database search and critical evaluation of the literature retrieved. This systematic methodology will help to create a valid bibliographic corpus and increase the validity and reproducibility of the obtained results (Fahimnia et al., 2015).

In the current research, special attention was paid to high-impact academic sources, as they could offer significant information about the theoretical background, intellectual evolution, and research directions of the sphere. The main bibliographic database used was Scopus because it has a broad multidisciplinary scope and has been used in bibliometric studies before (Al-Khoury et al., 2022; di Stefano et al., 2010; Khiste and Paithankar, 2017). In line with this, pertinent publications that were indexed in Scopus database of Elsevier and published between 1957 and August 2026 were retrieved and then underwent bibliometric analysis.

Data Search Strategy

The data collection process was designed in such a way that it formed a clear and reproducible bibliometric corpus on Java teaching. The search was performed in the Scopus database (article-title field) with the search expression TITLE ((teaching OR teach) AND java) and limited to the English language (see Figure 1). The alternatives in the usage of teaching and teach are methodologically grounded since the scholarly literature can use both morphological forms when referring to instructional practices, pedagogical interventions, or strategies of Java education. The article title search will offer a relatively narrow retrieval strategy to ensure that Java teaching is a primary theme of the publications retrieved, and not just a chance occurrence in the abstract or keywords. Simultaneously, storing all types of sources, types of documents, and areas of interest during the first retrieval level minimizes the chances of early elimination of valuable studies and maximizes the coverage of the bibliometric corpus. The wide time frame of 1957 to August 2026 also enables the analysis to reflect the historical growth and transformation of Java teaching research as opposed to setting an arbitrary publication date. However, since the research is supposed to describe a particular area of research, the further screening or selection criteria must be used to clearly differentiate between the relevant scholarly publications and the possible inappropriate types of documents to ensure the validity of the final dataset.

After the search strategy was predefined, 239 records were located to be subjected to bibliometric analysis. This size of corpus offers an adequately rich evidence base to study the trends of publications, the authors who have the most influence, the patterns on a source level, the structure of citations, and the intellectual evolution of the research on Java teaching. Notably, the resulting

number is to be viewed as a consequence of the intentional trade-off between recall and precision: the title-based search increases thematic specificity, and the broad Scopus coverage and inclusive source, document, and subject-area settings reduce the risk of missing relevant contributions. The resultant dataset thus offers a justifiable basis to the further bibliometric methods, which include citation analysis, co-authorship analysis, co-occurrence of keywords, bibliographic coupling and co-citation analysis. Besides, in the case when the final study is to examine peer-reviewed journal articles only, the figure must clearly reflect this limitation instead of stating Source Type: All and Document Type: All, which is critical to the consistency of the stated methodology with the actual Scopus filtering settings to guarantee the credibility and reproducibility of a Q1-level bibliometric study.

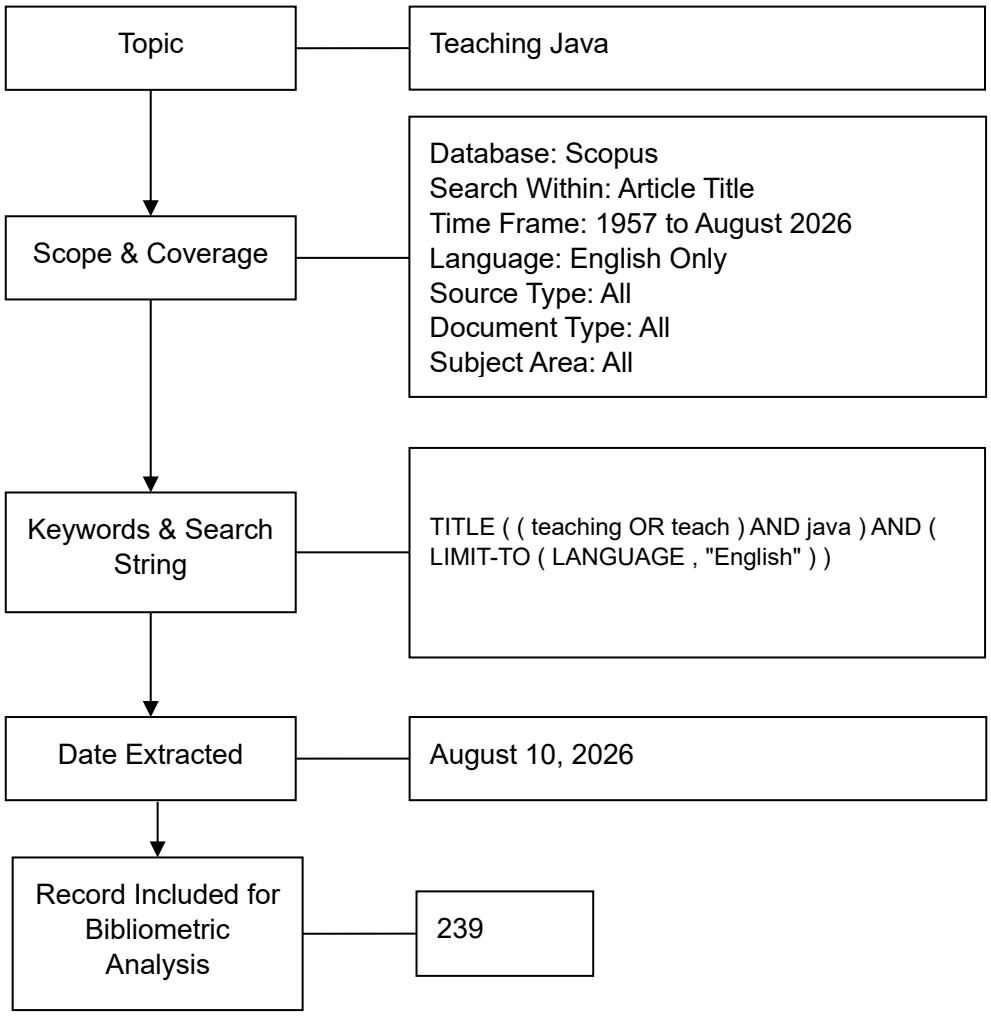


Figure 1. Flow Diagram of the Search Strategy

Data Analysis

VOSviewer is a proven bibliometrical analysis and visualization tool created by Nees Jan van Eck and Ludo Waltman of Leiden University, the Netherlands (van Eck & Waltman, 2010, 2017). It is specifically created to create, analyze, and visualize bibliometric networks, allowing researchers to explore the connections between scholarly entities, including authors, publications, journals, keywords, and citations. The software also offers various methods of analysis such as co-authorship, co-citation, bibliographic coupling, and keyword co-occurrence analysis, hence

International Journal of Education, Psychology and Counseling
IJEPC
EISSN : 0128-164X
Volume 11 Issue 65 (December 2026) PP. 01-19
helping to identify intellectual structures, thematic patterns, and collaborative relationships within a research field. One of the main strengths of VOSviewer is its ability to convert complex bibliographic relationships into meaningful visual displays using network and overlay visualizations and density visualizations. These visualizations allow the researchers to see clusters of closely related items, emerging research themes, influential contributors, and the structural development of a knowledge domain. Its interactive interface and the ability to flexibly map large bibliometric datasets also contribute to the exploration of large bibliometric datasets and enable the researcher to customize the analytical parameters and visualization settings to the purpose of the study.

In the current research, the bibliographic information (publication year, article title, author information, source journal, citation counts, and author keywords) was searched using the Scopus database in Excel format. The dataset included the publications published between 1957 and August 2026 and analyzed with VOSviewer version 1.6.21. The program was used to create bibliometric maps and find clusters and connections between the chosen publications, authors, and keywords. The VOSviewer usage is especially suitable in the current study since the aim is not limited to descriptive bibliographic statistics but the analysis of the intellectual and thematic framework of the research on the teaching of Java. VOS mapping and clustering allow placing publications and other bibliometric units in terms of their relatedness, which allows identifying the prevailing research themes and their relationships. The visualization, in turn, offers an analytical viewpoint that supplements traditional descriptive bibliometric indicators and allows interpreting the development of the research field and its organization in a more systematic way.
VOSviewer methodologically relies on VOS (Visualization of Similarities) approach which is conceptually similar to Multidimensional Scaling (MDS) but with mathematical differences in the formulation and normalization process (van Eck and Waltman, 2010). Both methods are intended to model the relationships between items in a low-dimensional space, but VOS puts special emphasis on the location of strongly related items close to each other and preserves a meaningful representation of the overall similarity structure. The property renders the method especially appropriate to bibliometric network data, such as co-occurrence and citation relations. VOSviewer uses normalization processes like association strength (AS) to measure the relatedness of two items taking into consideration the difference in occurrence frequencies.(van Eck & Waltman, 2010). The association-strength measure can be expressed as:

$$AS_{ij} = \frac{C_{ij}}{w_i w_j}$$

where c_{ij} represents the number of co-occurrences between items i and j, while w_i and w_j denote the total occurrence frequencies of items i and j, respectively. This normalization is important because raw co-occurrence frequencies may disproportionately favor highly frequent items. By applying association strength, VOSviewer provides a more balanced representation of the relatedness among bibliometric entities, thereby supporting the identification of meaningful clusters and patterns within the research landscape (van Eck & Waltman, 2010; Van Eck & Waltman, 2007).

Result and Discussion

This section discusses the findings according to the research questions, highlighting key results, emerging trends, and their implications within the study context.

International Journal of Education, Psychology and Counseling
IJEPC
EISSN : 0128-164X
Volume 11 Issue 65 (December 2026) PP. 01-19

RQ1: How Has Research on Teaching Java Evolved Over Time in Terms of Publication Growth and Scholarly Development?

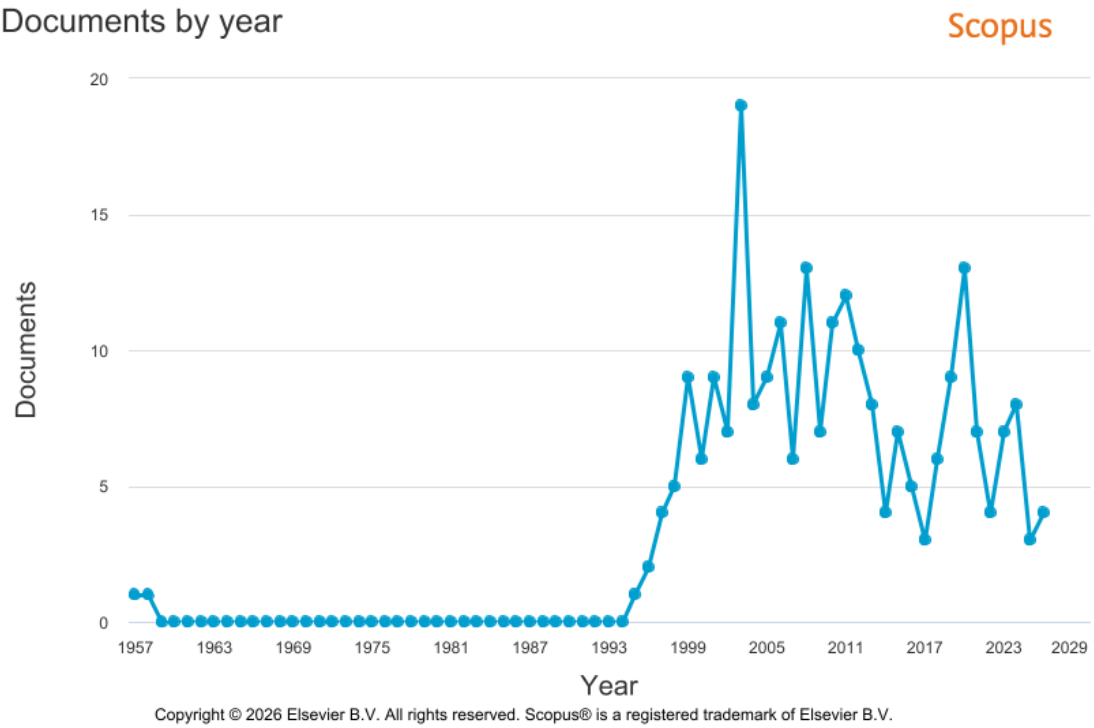


Figure 2: Trend of Research in Teaching Java by Years

Table 1: Percentage of Publication by Year

Year	Number of Publication	Percentage (%)
2026	4	1.67
2025	3	1.26
2024	8	3.35
2023	7	2.93
2022	4	1.67
2021	7	2.93
2020	13	5.44
2019	9	3.77
2018	6	2.51
2017	3	1.26
2016	5	2.09
2015	7	2.93

Figure 2 shows the temporal change in the study of Java programming, the variation in the activity of publications during the study period and Table 1 shows the distribution of the publications by year, which shows the temporal change in the research activity in the Java programming field. The

trend in the number of publications on Teaching Java between 2015 and August 2026 shows a comparatively low yet variable level of research output, and no steady rising or declining pattern. The highest number of publications was recorded in 2020 (13 publications; 5.44%), followed by 2019 (9; 3.77%), 2024 (8; 3.35%), and 2023 (7; 2.93%). The comparatively high publication rates of 2019-2020 could be indicative of a growing interest in enhancing programming education with Java-based pedagogies especially as higher education institutions moved more towards technology-enhanced, online, and student-centred learning methods. The steep rise in 2020 can also be attributed to the disruption occasioned by the COVID-19 pandemic, which hastened the shift towards traditional face-to-face programming teaching to online and blended learning. This fact might have prompted scholars to explore other methods of teaching Java, such as virtual learning environments, automated assessment, collaborative programming, and other online teaching tools. Conversely, the decrease to 4 publications in 2022 (1.67%) and 3 in 2025 (1.26) indicates that the research focus might have changed to the general teaching of programming, new programming languages, artificial intelligence, and technology-enhanced learning.

The fact that the volume of publications has decreased relatively since the 2020 peak is not necessarily an indicator of the decreased significance of Java as a teaching language. Instead, it can reflect a maturation and diversification of research in programming education, in which Java is being more and more viewed as a context of implementation in larger pedagogical research, and not as the focus of research. This interpretation is supported by the recovery in 20232024 (7 and 8 publications, respectively) and indicates a resurgence of interest in Java teaching, possibly due to studies on active learning, automated feedback, programming assessment, collaborative learning, and new educational technologies. In the meantime, the 2026 amount of four publications (1.67%) should be viewed with caution since the data is only up to August 2026, which is not a full publication year. On the whole, this trend shows that Teaching Java research has been fairly stable, but intermittent, with the publication activity being more affected by shifts in educational technology, instructional issues and the larger trends in programming-education than by a consistent growth in Java-specific research. The 2020 peak thus seems to be a special year of increased research activity, and the following swings indicate that Java remains a useful pedagogical language but is becoming more and more integrated into broader discourses on programming education.

RQ2: Which Publications Have Had the Greatest Scholarly Impact on Teaching Java, Based on Citation Performance?

Table 2. Top 10 Highly Cited Publications on Teaching Java

No	Authors	Title	Year	Source title	Cited by
1	Kölling M.; Rosenberg J.(Kölling & Rosenberg, 2001)	Guidelines for teaching object orientation with Java	2001	Proceedings of the Conference on Integrating Technology into Computer Science Education, ITiCSE	94
2	Sage D.; Unser M.(Sage & Unser, 2003)	Teaching Image-Processing Programming in Java	2003	IEEE Signal Processing Magazine	83

3	Barnes D.J.(Barnes, 2002)	Teaching introductory Java through LEGO MINDSTORMS models	2002	SIGCSE Bulletin (Association for Computing Machinery, Special Interest Group on Computer Science Education)	70
4	Becker B.W.(Becker, 2001)	Teaching CS1 with Karel the Robot in Java	2001	SIGCSE Bulletin (Association for Computing Machinery, Special Interest Group on Computer Science Education)	68
5	Sciore E.(Sciore, 2007)	SimpleDB: A simple java-based multiuser syst for teaching database internals	2007	SIGCSE 2007: 38th SIGCSE Technical Symposium on Computer Science Education	33
6	Riley D.(Riley, 2012)	Using mobile phone programming to teach Java and advanced programming to computer scientists	2012	SIGCSE'12 - Proceedings of the 43rd ACM Technical Symposium on Computer Science Education	30
7	Hauswirth M.; Adamoli A.(Hauswirth & Adamoli, 2013)	Teaching java programming with the Informa clicker system	2013	Science of Computer Programming	27
8	Rasala R.; Raab J.; Proulx V.K.(Rasala et al., 2001)	Java power tools: Model software for teaching object-oriented design	2001	SIGCSE Bulletin (Association for Computing Machinery, Special Interest Group on Computer Science Education)	27
9	Emurian H.H.; Holden H.K.; Abarbanel R.A.(Emurian et al., 2008)	Managing programmed instruction and collaborative peer tutoring in the classroom: Applications in teaching Java™	2008	Computers in Human Behavior	22
10	Burger K.R.(Burger, 2003)	Teaching two-dimensional array concepts in Java with image processing examples	2003	SIGCSE Bulletin (Association for Computing Machinery, Special Interest Group on Computer Science Education)	21

International Journal of Education, Psychology and Counseling
IJEPC
EISSN : 0128-164X
Volume 11 Issue 65 (December 2026) PP. 01-19

Table 2 gives the most-cited publications in Teaching Java research. The 10 highest-cited authors in this dataset have a total 475 citations with a strong concentration of influence at the highest rank. The four leading papers alone have 315 citations (66.3 of the total), with the first two papers, Kölling and Rosenberg (94 citations) and Sage and Unser (83 citations) leading. One of the main arguments to explain this distribution is the time-in-field effect: the publications published in the early 2000s (2001 - 2003) have 363 citations (76.4%), and the publications published after 2007 have a maximum of 33 citations. This time lead overlapped with the fast uptake of Java as the standard introductory computer science (CS1/CS2) language in introductory computer science courses globally in the early 2000s. The early scholars who released foundational instructional frameworks in this period of transition laid down baseline methodologies that were regularly referred to by other educators and researchers in the next 20 years.

In addition to the time dynamics, the other major force behind the citation volume is the pedagogical strategy promoted by the authors, namely, the application of visual, tangible, or microworld abstractions to reduce the steep learning curve of Java among beginners. High-impact articles became popular, substituting the dry syntax drills with more interesting real-world mental models, including physical robotics with LEGO MINDSTORMS (Barnes, 70 citations), visual microworlds with Karel the Robot (Becker, 68 citations), and graphical applications like image processing (Sage & Unser, 83 citations). Moreover, the concentration of venues is high: 7 of 10 publications are by ACM SIGCSE-affiliated venues (including SIGCSE Bulletins, Symposia, and ITiCSE). This validates the finding that the large numbers of citation in computer science education can be attributed to committed pedagogical communities in which classroom-tested software tools, guidelines, and curriculum designs are quickly adopted, tested, and cross-referenced.

RQ3: Which Countries Have Made the Greatest Contributions to Teaching Java Research in Terms of Publication Productivity and Citation Impact?

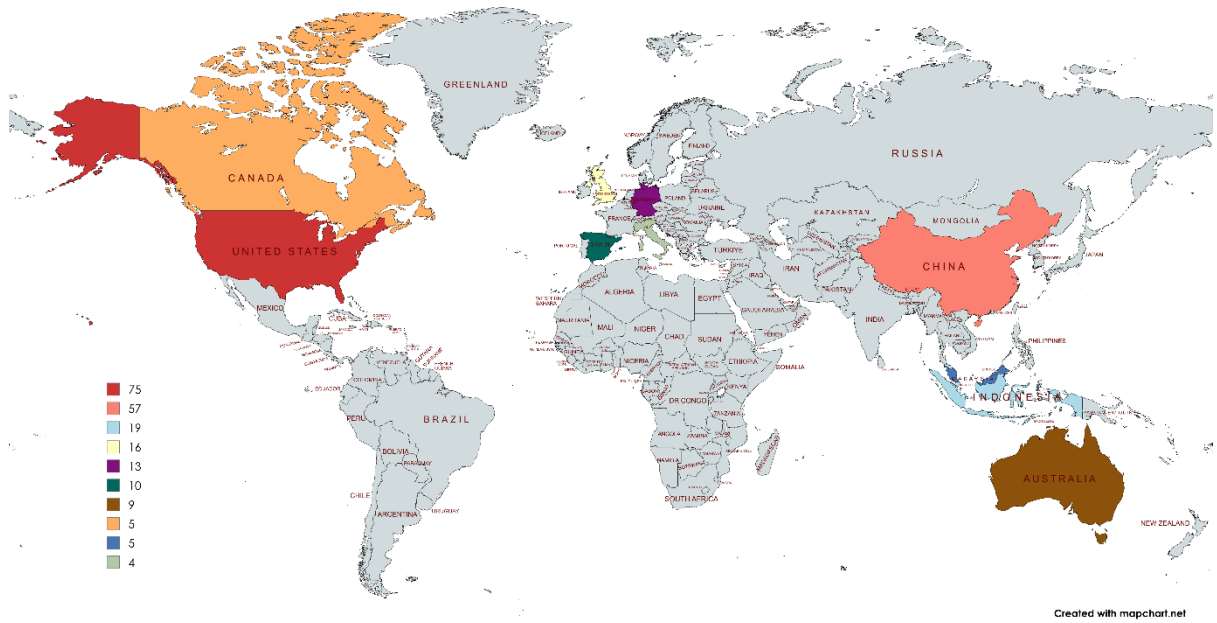


Figure 3: Country That Contributes the Publication in Teaching Java

International Journal of Education, Psychology and Counseling
IJEPC
EISSN : 0128-164X
Volume 11 Issue 65 (December 2026) PP. 01-19

The geographical distribution of publications on Teaching Java as shown in Figure 3 show that there are high concentrations in few countries with United States having 75 publications (31.38) and China having 57 publications (23.85). These two countries combined constitute 55.23 percent of the publications included in the dataset, which means that the study of Teaching Java is very much concentrated in the large research producing countries. The hegemony of the United States can be explained by its huge higher-education and computer science research community, significant investment in educational technology, and long-term use of Java in introductory programming and software engineering education. The second rank of China can also be attributed to its high higher-education sector, high number of students of computer science and increasing focus on programming and computing education. The fact that Indonesia (19; 7.95) and the United Kingdom (16; 6.69) contribute significantly less indicates a second group of active contributors. The relatively high ranking is especially significant and could be linked to the prevalence of Java in computer science and information technology programmes in the growing higher-education sector in Indonesia. In general, the distribution indicates that the productivity of publication depends not only on the relevance of Java as a programming language but also on the size of the academic community, the capacity to conduct research, the adoption of educational technology, and the institutional focus on computing education.

The rest of the countries - Germany (13; 5.44%), Spain (10; 4.18%), Australia (9; 3.77%), Canada (5; 2.09%), Malaysia (5; 2.09%), and Italy (4; 1.67%); have a smaller contribution, showing that the research on Teaching Java is geographically extensive but unevenly distributed. The fact that some European nations and English-speaking research systems are among the largest contributors implies that Java teaching research has grown in well-established computer science education communities, and that Malaysia, with a fifth-place tie with Canada (2.09%), has a significant but relatively minor contribution. Notably, the reduced number of publications of these countries may not always be the reduced interest or reduced research capacity, but also can be due to the size of the population, the number of universities where computing programmes are taught, the national research priorities, the patterns of collaboration, and the tendency of the researchers to publish articles under the larger themes like programming education, computer science education, or even computational thinking, rather than specifically use the term Teaching Java as the focus of the research. In this way, the country distribution is a testament to a research landscape dominated by the United States and China, but with contributions spread throughout Asia, Europe, North America, and Oceania, though with significant variation in research output.

International Journal of Education, Psychology and Counseling
IJEPC
EISSN : 0128-164X
Volume 11 Issue 65 (December 2026) PP. 01-19

RQ4: What Are the Dominant Research Themes, Emerging Topics, And Thematic Relationships in Teaching Java Research Based on Keyword Co-Occurrence Analysis?

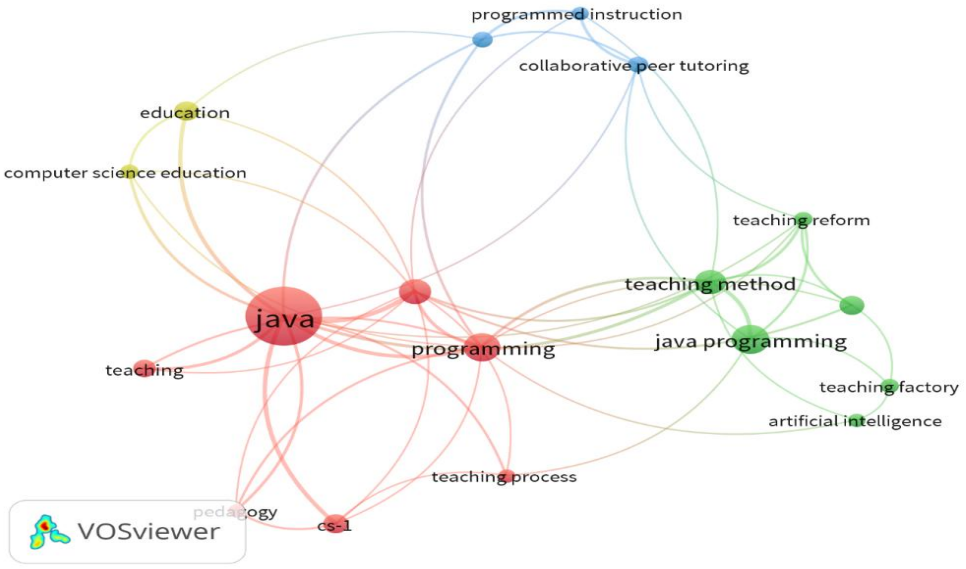


Figure 4: Network Visualization Map of Keywords' Co-Occurrence

Figure 4 shows the co-occurrence network of key words created with the help of VOSviewer and demonstrates the conceptual framework and the prevailing areas of research focus in the literature on Java programming education. The most obvious one is the keyword of Java, as it has the most occurrences (56) and the total link strength (TLS) of 45, which means that Java is not only a regularly studied concept, but also closely related to other significant concepts in the field of research. This dominance is also supported by the presence of such terms as Java programming (18 occurrences; TLS = 10) and Java language (9 occurrences; TLS = 7), which proves that Java is a core technological setting of programming education research. In the meantime, the terms programming (17 occurrences; TLS = 24) and object-oriented programming (14 occurrences; TLS = 23) have relatively high link strengths, which implies that the studies with Java are closely related to the basics of programming and the principles of object-oriented programming. The comparatively high connectivity of these keywords suggests that the literature is not limited to Java as a programming language but has a significant emphasis on how its object-oriented nature can be used to help students acquire fundamental knowledge and skills in programming.

Figure 4 further indicates a strong pedagogical orientation, with "teaching method" recording 13 occurrences and a TLS of 20, followed by "teaching" (8; TLS = 7), "pedagogy" (5; TLS = 8), and "teaching process" (5; TLS = 5). This pattern suggests that research on Java programming has increasingly emphasized *how programming is taught*, rather than focusing exclusively on technical content. The presence of "cs-1" (9; TLS = 11), "introductory programming" (7; TLS = 10), and "computer science education" (6; TLS = 7) further supports the argument that Java is predominantly examined within foundational programming courses and early-stage computer science education. In addition, keywords such as "collaborative peer tutoring" (7; TLS = 8), "teaching reform" (5; TLS = 10), and "artificial intelligence" (5; TLS = 3) indicate emerging interest in collaborative, innovative, and technology-enhanced pedagogical approaches. It is also interesting to note that the relatively low frequency of artificial intelligence in comparison to Java

International Journal of Education, Psychology and Counseling
IJEPC
EISSN : 0128-164X
Volume 11 Issue 65 (December 2026) PP. 01-19

and teaching-related keywords indicates that AI is a developing, but not a dominant research focus. In general, the network in Figure 2 indicates that the field is mainly organized around the interplay of Java programming, basic programming education, and pedagogical innovation, with an increasing focus on collaborative learning, teaching reform, and AI-assisted learning practices.

RQ5: What Are the Patterns and Structures of International Research Collaboration in Teaching Java, As Revealed Through Country-Level Co-Authorship Networks?

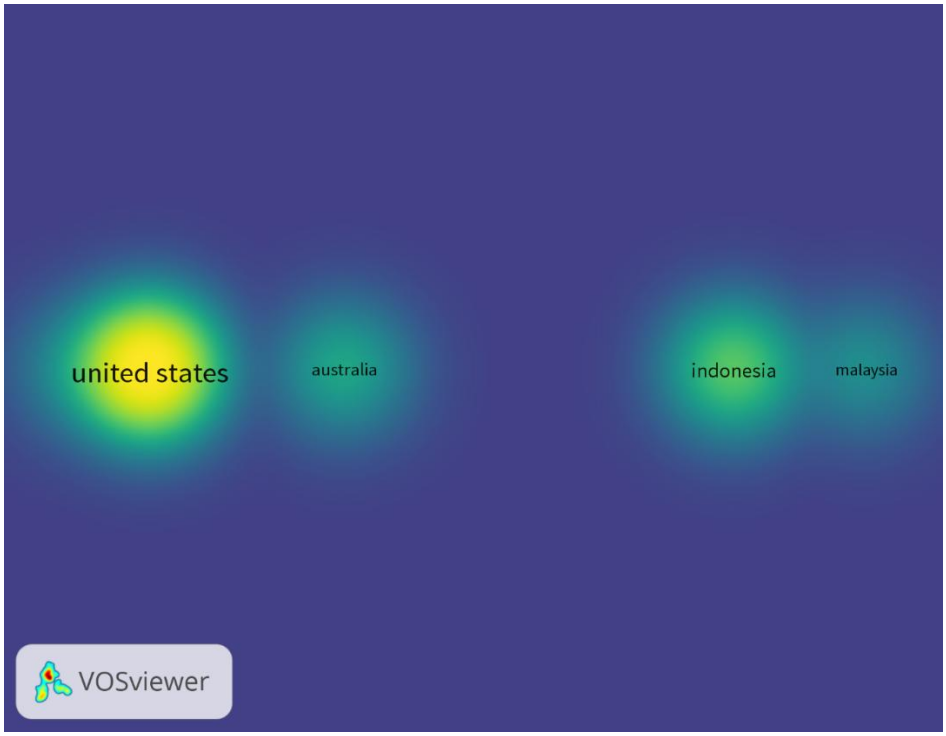


Figure 5: Co-Author Occurrence Visualization

Figure 5 shows that the United States and China have a significant proportion of the absolute volume of research output, but with different profiles of academic impact. The United States is on the top with 75 documents and 514 citations, which is a high citation-per-document ratio (about 6.85 citations/paper), but China is second in the list with 57 documents and only 71 citations (1.25 citations/paper). Australia also has an impressive scientific impact per capita with 209 citations in only 9 documents (23.2 citations/paper), surpassing even the intellectual reach of larger producers such as China and Indonesia. This difference is mostly motivated by the disparity in academic maturity, publication traditions, and language reach; western institutions (US, UK, Australia) have traditionally been the leaders in curriculum standardizations (including ACM/IEEE recommendations) and have venues of high impact computing education, allowing them to adopt earlier and accumulate citations faster.

Although the number of documents is high, Figure 2 shows that the total scores of the link strength are significantly low in nearly all major countries, which indicates a clear absence of cross-border international cooperation in this particular area of research. China and Spain have a total link strength of 0 and the United States has a link strength of just 2, which means that the majority of the research output is produced by domestic co-author groups in closed national borders. The UK (5), Germany (5), Indonesia (4) and Malaysia (3) have a bit more tendency towards relative

collaboration. The main rationale of this disjointed network of collaboration is that computer science education research is often associated with local curriculum changes, institution-specific course applications, and localized instructional practices, which inherently prefer local institutional collaborations to international co-authorship.

Conclusion

This bibliometric study examined the evolution, conceptual framework, and patterns of collaboration in Java programming pedagogy based on 239 publication records obtained via Scopus. In answering the key research questions, the findings suggest that the volume of publications was episodic with a significant peak in 2020 and then stabilized at the baseline levels. The literature that is highly cited is mostly found in the early 2000s, motivated by foundational visual toolkits and introductory frameworks designed to fit CS1 courses. Geographically, the production of research is very concentrated in the United States and China, but Western institutions tend to have higher citation impacts per document. The clusters of keyword co-occurrence demonstrate that the studies are centered around the main principles of object-oriented, teaching methods, and the introductory problems in programming, and artificial intelligence is a new direction of interest. Moreover, country-level co-authorship mapping reveals limited cross-border cooperation, which is an orientation towards local curriculum projects rather than international collaborative research.

These results add to the literature at large by offering a systematic and quantitative mapping of the way Java education research changed over time. To be applied in practice, the analysis reveals a distinct movement towards active, student-centered learning designs, active code visualization and automated assessment systems as opposed to passive lecture delivery. These insights can be used by educators and curriculum designers to ensure that introductory courses are based on established pedagogical strategies. The main limitation of this research is that it only used the title-field search of Scopus, which might not include the papers that address Java instruction mostly in the abstract or keywords. This limitation can be resolved by future studies by including complementary databases like Web of Science or IEEE Xplore, expanding search options to full-text areas, and further examination of the emerging technologies like generative AI in programming education. The results of this study, in general, indicate the usefulness of bibliometric mapping in determining the trends in pedagogy and offers a baseline to base future research in computer science education.

Acknowledgements: The authors would like to express their sincere gratitude to Universiti Teknologi MARA (UiTM) Cawangan Pahang, Kampus Raub for providing the opportunity and support to undertake research activities through the approved leave to attend the research course. This support provided valuable time and an appropriate environment for enhancing research knowledge, developing the manuscript, and completing the present study. The authors sincerely appreciate the institution's commitment to supporting academic development and research productivity, which contributed significantly to the successful completion of this research and the resulting publication.

Funding Statement: No Funding

Conflict of Interest Statement: The authors declare that there is no conflict of interest regarding the publication of this paper. All authors have contributed to this work and approved the final version of the manuscript for submission to the journal.

Ethics Statement: This study did not involve any human participants, animals, or sensitive data requiring ethical approval. The authors confirm that the research was conducted in accordance with accepted academic integrity and ethical publishing standards.

Author Contribution Statement: All authors contributed significantly to the development of this manuscript. Mohd Rizal Razak and Nor Zalina Ismail was responsible for the conceptualization, methodology, and overall supervision of the study. Nor ‘aqilah Misman, Mohd Azim Zainal and Khairunnisa Abdul Kadir handled data collection, analysis, and interpretation of results. Ahmad Subhi Zolkafly and Mohamad Affendi Abdul Malek contributed to the literature review, drafting, and critical revision of the manuscript. All authors read and approved the final version of the manuscript prior to submission.

International Journal of Education, Psychology and Counseling
IJEPC
EISSN : 0128-164X
Volume 11 Issue 65 (December 2026) PP. 01-19

References

Ahmed, S., Albarkoli, M., Alfitouri, A., Ahmed, K., & Alawan, M. (2026). Assessing the Impact of the LearnOOP Tool on Enhancing the Understanding of Object-Oriented Programming-An Experimental Study Across Three Semesters. *2026 IEEE 5th International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA)*, 1–8. https://doi.org/10.1109/mi-sta68962.2026.11511253

Al-Khoury, A., Hussein, S. A., Abdulwhab, M., Aljuboori, Z. M., Haddad, H., Ali, M. A., Abed, I. A., & Flayyih, H. H. (2022). Intellectual Capital History and Trends: A Bibliometric Analysis Using Scopus Database. *Sustainability (Switzerland)*, *14*(18). https://doi.org/10.3390/su141811615

Alves, J. L., Borges, I. B., & De Nadae, J. (2021). Sustainability in complex projects of civil construction: Bibliometric and bibliographic review. *Gestao e Producao*, *28*(4). https://doi.org/10.1590/1806-9649-2020v28e5389

Assyakur, D. S., & Rosa, E. M. (2022). Spiritual Leadership in Healthcare: A Bibliometric Analysis. *Jurnal Aisyah : Jurnal Ilmu Kesehatan*, *7*(2). https://doi.org/10.30604/jika.v7i2.914

Barnes, D. J. (2002). Teaching introductory Java through LEGO MINDSTORMS models. *SIGCSE Bulletin (Association for Computing Machinery, Special Interest Group on Computer Science Education)*, 147–151. https://doi.org/10.1145/563517.563397

Becker, B. W. (2001). Teaching CS1 with Karel the Robot in Java. *SIGCSE Bulletin (Association for Computing Machinery, Special Interest Group on Computer Science Education)*, 50–54. https://doi.org/10.1145/366413.364536

Burger, K. R. (2003). Teaching two-dimensional array concepts in Java with image processing examples. *SIGCSE Bulletin (Association for Computing Machinery, Special Interest Group on Computer Science Education)*, 205–209. https://doi.org/10.1145/792548.611970

Calderon, I., Silva, W., & Feitosa, E. (2023). Active Learning Methodologies for Teaching Programming in Undergraduate Courses: A Systematic Mapping Study. *Informatics Educ.*, *23*, 279–322. https://doi.org/10.15388/infedu.2024.11

Chimanga, K., Siluyele, N., Langton, S., & Nkonde, E. (2021). A Comparative Study of Students Performance in Object Oriented Programming (OOP) Using Java Before and During COVID-19 (A Case of Kapasa Makasa Campus). *American Journal of Networks and Communications*. https://doi.org/10.11648/j.ajnc.20211001.11

di Stefano, G., Peteraf, M., & Veronay, G. (2010). Dynamic capabilities deconstructed: A bibliographic investigation into the origins, development, and future directions of the research domain. *Industrial and Corporate Change*, *19*(4), 1187–1204. https://doi.org/10.1093/icc/dtq027

Emurian, H. H., Holden, H. K., & Abarbanel, R. A. (2008). Managing programmed instruction and collaborative peer tutoring in the classroom: Applications in teaching Java™. *Computers in Human Behavior*, *24*(2), 576–614. https://doi.org/10.1016/j.chb.2007.02.007

Fahimnia, B., Sarkis, J., & Davarzani, H. (2015). Green supply chain management: A review and bibliometric analysis. In *International Journal of Production Economics* (Vol. 162, pp. 101–114). https://doi.org/10.1016/j.ijpe.2015.01.003

Guo, X., & Hu, X. (2024). Teaching Reform Based on OBE Object-Oriented Programming Language (Java) Course. *Proceedings of the 3rd International Conference on Internet Technology and Educational Informatization, ITEI 2023, November 24–26, 2023, Zhengzhou, China*. https://doi.org/10.4108/eai.24-11-2023.2343639

Gutiérrez, L., Guerrero, C., & López-Ospina, H. (2022). Ranking of problems and solutions in the teaching and learning of object-oriented programming. *Education and Information Technologies*, *27*, 7205–7239. https://doi.org/10.1007/s10639-022-10929-5

Haruna, A. A. (2023). Towards enhancing teaching and learning computer programming in Saudi Arabia. *International Journal of ADVANCED AND APPLIED SCIENCES*. https://doi.org/10.21833/ijaas.2023.03.024

Hauswirth, M., & Adamoli, A. (2013). Teaching java programming with the Informa clicker system. *Science of Computer Programming*, *78*(5), 499–520. https://doi.org/10.1016/j.scico.2011.06.006

Khiste, G. P., & Paithankar, R. R. (2017). Analysis of Bibliometric term in Scopus. *International Research Journal*, *01*(32), 78–83.

Khudhur, N., Nurmaya, Hayashi, Y., & Hirashima, T. (2023). Conceptual Representation of the Source-Code to Support the Learning of Object-Oriented Programming Concepts. *International Journal of Information and Education Technology*. https://doi.org/10.18178/ijiet.2023.13.12.1999

Kölling, M., & Rosenberg, J. (2001). Guidelines for teaching object orientation with Java. *Proceedings of the Conference on Integrating Technology into Computer Science Education, ITiCSE*, 33–36. https://doi.org/10.1145/507758.377461

Lian, V., Varoy, E., & Giacaman, N. (2022). Learning Object-Oriented Programming Concepts Through Visual Analogies. *IEEE Transactions on Learning Technologies*, *15*, 78–92. https://doi.org/10.1109/tlt.2022.3154805

Luo, L., & Wu, Q. (2025). Optimization of Course Content and Exploration of Teaching Pathways for the JavaEE Framework Technology Course Oriented Toward Innovative Talent Cultivation. *Education Insights*. https://doi.org/10.70088/5r1feb17

M, M., & S, S. (2023). From Observation to Active Learning: A Collaborative Learning Approach for Object Oriented Programming Course. *Journal of Engineering Education Transformations*. https://doi.org/10.16920/jeet/2023/v36i4/23125

Maspaeni, Muslim, S., & Ibrahim, N. (2022). Improving Course Learning Outcomes Object-Oriented Programming through Blended Learning. *International Journal of Information and Education Technology*. https://doi.org/10.18178/ijiet.2022.12.8.1674

Menolli, A., & Strik, B. (2025). *Educational Insights from Code: Mapping Learning Challenges in Object-Oriented Programming through Code-Based Evidence*. 544–554. https://doi.org/10.48550/arxiv.2507.17743

Messer, M., Brown, N., Kölling, M., & Shi, M. (2023). Automated Grading and Feedback Tools for Programming Education: A Systematic Review. *ACM Transactions on Computing Education*, *24*, 1–43. https://doi.org/10.1145/3636515

Omer, U., Farooq, M., & Abid, A. (2021). Introductory programming course: review and future implications. *PeerJ Computer Science*, *7*. https://doi.org/10.7717/peerj-cs.647

Rasala, R., Raab, J., & Proulx, V. K. (2001). Java power tools: Model software for teaching object-oriented design. *SIGCSE Bulletin (Association for Computing Machinery, Special Interest Group on Computer Science Education)*, 297–301. https://doi.org/10.1145/366413.364606

Riley, D. (2012). Using mobile phone programming to teach Java and advanced programming to computer scientists. *SIGCSE'12 - Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, 541–546. https://doi.org/10.1145/2157136.2157292

Sage, D., & Unser, M. (2003). Teaching Image-Processing Programming in Java. *IEEE Signal Processing Magazine*, *20*(6), 43–52. https://doi.org/10.1109/MSP.2003.1253553

Sciore, E. (2007). SimpleDB: A simple java-based multiuser syst for teaching database internals. *SIGCSE 2007: 38th SIGCSE Technical Symposium on Computer Science Education*, 561–565. https://doi.org/10.1145/1227310.1227498

Sun, Y. (2023). Research on Teaching Reform of Object-oriented Programming under OBE Concept. *Proceedings of the 4th International Conference on Education, Knowledge and Information Management, ICEKIM 2023, May 26–28, 2023, Nanjing, China*. https://doi.org/10.4108/eai.26-5-2023.2337297

Tshukudu, E., Cutts, Q., & Foster, M. E. (2021). Evaluating a Pedagogy for Improving Conceptual Transfer and Understanding in a Second Programming Language Learning Context. In *Proceedings of the 21st Koli Calling International Conference on Computing Education Research*. https://doi.org/10.1145/3488042.3488050

Tsichouridis, A., Xinogalos, S., & Ampatzoglou, A. (2023). Educational Programming Environments for Enhancing Conceptual Design in the Object-Oriented Paradigm: A Systematic Mapping Study. *Journal of Educational Computing Research*, *62*, 290–322. https://doi.org/10.1177/07356331231203251

van Eck, N. J., & Waltman, L. (2010). Software survey: VOSviewer, a computer program for bibliometric mapping. *Scientometrics*, *84*(2), 523–538. https://doi.org/10.1007/s11192-009-0146-3

van Eck, N. J., & Waltman, L. (2017). Citation-based clustering of publications using CitNetExplorer and VOSviewer. *Scientometrics*, *111*(2), 1053–1070. https://doi.org/10.1007/s11192-017-2300-7

Van Eck, N. J., & Waltman, L. (2007). Bibliometric mapping of the computational intelligence field. *International Journal of Uncertainty, Fuzziness and Knowldege-Based Systems*, *15*(5), 625–645. https://doi.org/10.1142/S0218488507004911

Verbeek, A., Debackere, K., Luwel, M., & Zimmermann, E. (2002). Measuring progress and evolution in science and technology - I: The multiple uses of bibliometric indicators. *International Journal of Management Reviews*, *4*(2), 179–211. https://doi.org/10.1111/1468-2370.00083

Wang, Q., Zheng, L., & Hong, R. (2024). Exploration on the Teaching Model of Java Programming and Practice for Students with No Programming Background. *Advances in Educational Technology and Psychology*. https://doi.org/10.23977/aetp.2024.080629

Wu, Y. C. J., & Wu, T. (2017). A decade of entrepreneurship education in the Asia Pacific for future directions in theory and practice. In *Management Decision* (Vol. 55, Number 7, pp. 1333–1350). https://doi.org/10.1108/MD-05-2017-0518

Xing, X., & Yanan, Z. (2024). Application and practice of personalized education based on big data analysis in Java object-oriented program design. *MATEC Web of Conferences*. https://doi.org/10.1051/matecconf/202439501041