

References for C-the-point

- [1] Ella Albrecht and Jens Grabowski. “Sometimes it’s just sloppiness-studying students’ programming errors and misconceptions”. In: *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. 2020, pp. 340–345.
<https://dl.acm.org/doi/pdf/10.1145/3328778.3366862>.
- [2] Nabeel Alzahrani and Frank Vahid. “Common logic errors for programming learners: A three-decade literature survey”. In: *2021 ASEE Annual Conference*. 2021.
<https://par.nsf.gov/servlets/purl/10329375>.
- [3] Yorah Bosse et al. “Catalogs of c and python antipatterns by cs1 students”. In: *arXiv preprint arXiv:2104.12542* (2021).
<https://arxiv.org/pdf/2104.12542>.
- [4] Michelle Craig and Andrew Petersen. “Student difficulties with pointer concepts in C”. In: *Proceedings of the Australasian Computer Science Week Multiconference*. 2016, pp. 1–10.
<https://dl.acm.org/doi/pdf/10.1145/2843043.2843348>.
- [5] Adwoa Donyina and Reiko Heckel. “Pedagogy of Teaching Pointers in the C Programming Language using Graph Transformations”. In: *arXiv preprint arXiv:2503.20469* (2025).
<https://arxiv.org/pdf/2503.20469>.
- [6] Matthew Heinsen Egan and Chris McDonald. “Program visualization and explanation for novice C programmers”. In: *Proceedings of the Sixteenth Australasian Computing Education Conference-Volume 148*. 2014, pp. 51–57.
<https://dl.acm.org/doi/epdf/10.5555/2667490.2667496>.
- [7] Andrew Ettles, Andrew Luxton-Reilly, and Paul Denny. “Common logic errors made by novice programmers”. In: *Proceedings of the 20th Australasian Computing Education Conference*. 2018, pp. 83–89.
<https://dl.acm.org/doi/pdf/10.1145/3160489.3160493>.
- [8] Matthew Heinsen Egan and Chris McDonald. “An evaluation of SeeC: a tool designed to assist novice C programmers with program understanding and debugging”. In: *Computer Science Education* 31.3 (2021), pp. 340–373.
<https://annas-archive.li/scidb/10.1080/08993408.2020.1777034/>.
- [9] Ryosuke Ishizue et al. “PVC. js: visualizing C programs on web browsers for novices”. In: *Heliyon* 6.4 (2020).
[https://www.cell.com/heliyon/fulltext/S2405-8440\(20\)30651-4](https://www.cell.com/heliyon/fulltext/S2405-8440(20)30651-4).
- [10] Lisa C Kaczmarczyk et al. “Identifying student misconceptions of programming”. In: *Proceedings of the 41st ACM technical symposium on Computer science education*. 2010, pp. 107–111.
<https://dl.acm.org/doi/epdf/10.1145/1734263.1734299>.

- [11] Utku Kose and Omer Deperlioglu. “Intelligent learning environments within blended learning for ensuring effective C programming course”. In: *arXiv preprint arXiv:1205.2670* (2012).
<http://arxiv.org/pdf/1205.2670>.
- [12] Kayvan Memarian et al. “Into the depths of C: elaborating the de facto standards”. In: *ACM SIGPLAN Notices* 51.6 (2016), pp. 1–15.
<https://dl.acm.org/doi/epdf/10.1145/2980983.2908081>.
- [13] Iain Milne and Glenn Rowe. “Ogre: Three-dimensional program visualization for novice programmers”. In: *Education and Information Technologies* 9.3 (2004), pp. 219–237.
<https://link.springer.com/article/10.1023/B:EAIT.0000042041.04999.17>.
- [14] Juha Sorva, Ville Karavirta, and Lauri Malmi. “A review of generic program visualization systems for introductory programming education”. In: *ACM Transactions on Computing Education (TOCE)* 13.4 (2013), pp. 1–64.
<https://dl.acm.org/doi/pdf/10.1145/2490822>.
- [15] Akhila Sri Manasa Venigalla, Chaitanya S Lakkundi, and Sridhar Chimalakonda. “PointerViz-towards visualizing pointers for novice programmers”. In: (2020).
<https://scholarspace.manoa.hawaii.edu/server/api/core/bitstreams/d6caf72c-4be5-409e-ab5d-709cbb9e713a/content>.
- [16] Koichi Yamashita et al. “Classroom practice for understanding pointers using learning support system for visualizing memory image and target domain world”. In: *Research and Practice in Technology Enhanced Learning* 12.1 (2017), p. 17.
<https://link.springer.com/content/pdf/10.1186/s41039-017-0058-4.pdf>.