7th Symposium on the History and Philosophy of Programming

Call for Papers

Commission for the History and Philosophy of Computing HaPoC
<https://hapoc.org/>

What is critical about programming?

13th of November 2026 - Universität Basel, Switzerland

We are pleased to announce the call for papers for the 7th symposium on the history and philosophy of programming (HaPoP) on November 13th, 2026. We invite contributions on the history and philosophy of programming broadly understood, whether conceptual concerns or practical matters, field-specific examinations or cross-disciplinary comparisons, as well as, but not limited to, epistemic, aesthetic and ethical reflections.

In the spirit of interdisciplinary inquiry into programming, this symposium will focus on "criticality in programming", and asking broadly "what is critical about programming?". Recalling the shared etymological root of "critical" with "crisis", we start from appreciating a critical moment, a crisis, as that which requires a decision, and thus sharp, unambiguous judgment, to dismiss any doubt. A crisis requires a critical assessment of a situation, and subsequent action, considering programming as both a reflexive and decisive action.

The initial crisis of programming, the so-called "software crisis" in the 1960s, was addressed through hardening of practices (Dijkstra 2007). Critically assessing the state of programming practices resulted in resolving a crisis by changing how programming is done. How was the engineering crisis conceived of in the first place? Then, what kinds of soft uncertainties were taken out to solve such a crisis? What kinds of methodologies (e.g. code reviews, proofs, tests, perpetual beta) and arrangements (e.g. specialisation and professionalisation) compose this hardening, and how were judgments passed as to whether the software engineering crisis had been averted? More recently, the development of tooling based on large language

models might constitute a new critical moment. How should they be assessed when employed in the building of critical systems? What kind of crisis might it represent to existing conceptions and histories of programming?

Within computer science, critical programming usually refers to the design, implementation, modeling, testing, execution, maintenance and repair of software in so-called critical systems. Programming thus makes automated decisions about potential crises, also where software directly causes a loss of human life (Baase 2008, Bächle and Bareis 2025). So how and when was the criticality of a system defined? How is criticality itself articulated along mission-critical, safety-critical, business-critical, and performance-critical? How do adjacent notions of risk, safety and harm at the center of modernity (Beck 1992) (dis)qualify certain kinds of system from being considered critical (November et al. 2010)? And how are such risks addressed through renewed programming practices, from mathematical proofs and extensive test suites to programming styles and design processes (Pingree et al. 2002)?

We can also consider programming not as the object, but as a means of conducting critique. Working on the fringes of software engineering, independent game developers, hackers, designers, artists have been using and creating computer programs in order to shift perspectives on domains beyond programming *per se* (Cox and McLean 2013; Soon and Cox 2021; Lavigne 2023). How can movements such as free software, hacktivism or net.art be understood historically, and how have they shifted over time? What kind of addition to the broader practice of criticism is programming? When is a use of programming to criticize other domains itself uncritical, and when does it involve being self-critical as well (Sollfrank and Soon 2021)? And when and how is programming used critically against critical systems (Skare 2016)? The intent here is to reflect on how programming can help uncover existing modes of thought in non-programming domains.

Finally, the critical outside of computer science refers not so much to the status as to the process. From Immanuel Kant to Michel Foucault, the critical stance or *critique* aims at determining the limitations of one's knowledge and abilities, whether intrinsically linked to a thinking subject (Kant 1998) or extrinsically rooted in societal structures (Foucault [1978] 2024; Butler 2002), culture, and the material world. Critically examining programming involves tracing and revealing the ways in which programming is limited in itself (beyond the oft-cited halting problem forever haunting programming (Turing 1937)), or in which it limits the abilities of the domains and users upon which it operates. Here, we take programming, and programming artefacts, as important factors in contemporary culture, and thus

valid objects of critique (Fraser 1985). What is critiqued about programming, where do those critiques stem from, and how are their respective merits evaluated? What are genealogies of critical stances towards programming, such as critical technical practice (Agre 1997) or permacomputing (Mansoux et al. 2023), and what have been the results of these critiques? Which lessons can be taken from the history of the open source movement which, over half a century, transformed from a critique to an industrial and IT governance mainstream? How can programming (and computer science as a field) learn from other disciplines to develop its own critique(s) (Petricek 2025)? One of the stakes here is then the extent to which programming can provide a critical re-examination of itself using its own theoretical and methodological baggage.

We therefore invite abstracts on the history and philosophy of programming, including, but not restricted to, the following topics:

- Histories and definitions of criticality in software
- Histories of critiques of programming
- Methods of programming critical systems
- Relationships between safety, error, logic and harm
- Critical movements within programming
- Contested programming practices, structures and concepts
- Epistemology of modeling and simulation
- Successes and failures of critical arguments conducted as computer programs
- Artistic projects and approaches to programming
- Hacking as a critical activity
- Programming as socio-political interventions

Abstracts of around 500 words (including references) should be sent to basel-2026@hapoc.org by 2026-07-31. Final decisions will be announced by 2026-08-31.

We particularly invite graduate students and early career researchers from under-represented communities and disadvantaged backgrounds. Travel funding will be available for graduate students and researchers without permanent employment. Requests for travel funding of up to 250€ should be directed to treasurer@hapoc.org, after notification of acceptance. Final decisions on the allocation of travel funds will be made 2026-09-15.

Organisation Team

Pierre Depaz (University of Basel), Yannick Nepomuk Fritz (University of Basel) and Mace Ojala (Ruhr-University Bochum).

Program Committee

Arianna Borelli (RWTH Aachen), Simon Castellan (INRIA), Baptiste Chenus (EHESS), Ranjodh Singh Dhaliwal (University of Basel), Juan-Luis Gastaldi (ETH Zürich), Nadja Gaudillière-Jami (Royal Danish Academy of Fine Arts), Gabriele Gramelsberger (RWTH Aachen), Amira Moeding (University of Cambridge), Sophie Quinton (INRIA), Mate Szabo (University of Southern California) and the organisation team.

References

Agre, Philip. 1997. “Toward a Critical Technical Practice.” In *Bridging the Great Divide: Social Science, Technical Systems, and Cooperative Work*, edited by Geoffrey C. Bowker, Lee Gasser, Susan Leigh Star, and Bill Turner. Erlbaum.

Beck, Ulrich.1992. *Risk Society: Towards a New Modernity*. Sage Publications.

Baase, Sara. 2008. *A Gift of Fire : Social, Legal, and Ethical Issues for Computing and the Internet*. Upper Saddle River, NJ : Pearson Prentice Hall.

Butler, Judith. 2002. “What Is Critique? An Essay on Foucault’s Virtue.” In *The Political: Readings in Continental Philosophy*, edited by David Ingram. Basil Blackwell.

Bächle, Thomas Christian and Jascha Bareis (eds.) 2025. *The Realities of Autonomous Weapons*. Bristol University Press.

Cox, Geoff, and Christopher Alex McLean. 2013. *Speaking Code: Coding as Aesthetic and Political Expression*. MIT Press.

Dijkstra, Edsger W. 2007. “The Humble Programmer.” In *ACM Turing Award Lectures*. Association for Computing Machinery.

Foucault, Michel. (1978) 2024. “What Is Critique?” In *“What Is Critique?” And “The Culture of the Self.”* University of Chicago Press.

Fraser, Nancy. 1985. “What’s Critical about Critical Theory? The Case of Habermas and Gender.” *New German Critique*, no. 35: 97–131. https://doi.org/10.2307/488202.

Kant, Immanuel. 1998. *Critique of Pure Reason*. Edited by Paul Guyer and Allen W. Wood. The Cambridge Edition of the Works of Immanuel Kant. Cambridge University Press.

Lavigne, Sam. 2023. “Scrapism: A Manifesto.” *Critical AI* 1 (1–2).

Mansoux, Aymeric, Brendan Howell, Dušan Barok, Ville-Matias Heikkilä. 2023. "Permacomputing Aesthetics: Potential and Limits of Constraints in Computational Art, Design and Culture". *Computing within Limits*.

November, Valérie, Eduardo Camacho-Hübner, and Bruno Latour. 2010. “Entering a Risky Territory: Space in the Age of Digital Navigation.” *Environment and Planning D: Society and Space* 28 (4): 581–99. https://doi.org/10.1068/d10409.

Petricek, Tomas. 2025. “Critical Architecture/Software Theory.” Tomasp.Net. https://tomas.net/architecture.

Pingree, P. J., E. Mikk, G. J. Holzmann, M. H. Smith, and D. Dams. 2002. “Validation of Mission Critical Software Design and Implementation Using Model Checking [Spacecraft].” *Proceedings. The 21st Digital Avionics Systems Conference* 1: 6A4-6A4.

Skare, Erik. 2016. *Digital Jihad: Palestinian Resistance in the Digital Era*. Bloomsbury Publishing.

Sollfrank, Cornelia, and Winnie Soon. 2021. *Fix my Code*. EECLECTIC.

Soon, Winnie and Geoff Cox. *Aesthetic Programming. A Handbook of Software Studies*. Open Humanities Press. https://www.aesthetic-programming.net/

Turing, Alan. 1937. "On Computable Numbers, with an Application to the Entscheidungsproblem". *Proceedings of the London Mathematical Society.*