callforpaper.org

Canada April 27-28, 2025

CONFERENCE WEBSITE: https://conf.researchr.org/home/icpc-2025

About the Conference

========== CALL FOR PAPERS ==========

33rd IEEE/ACM International Conference on Program Comprehension (ICPC)

04-27-2025
04-28-2025

Ottawa Canada

https://conf.researchr.org/home/icpc-2025

========== IMPORTANT DATES (Research Track) ==========

Fri 15 Nov 2024 - Abstract Deadline

Fri 22 Nov 2024 - Paper Deadline

Sun 12 Jan 2025 - Paper Notification

Wed 5 Feb 2025 - Camera Ready

Important Dates

APR 27

CONFERENCE DATE
April 27-28, 2025

========== TYPES OF SUBMISSION ==========

ICPC 2025 has four tracks:

* Research track (up to 10 pages + 2 additional pages for references).

* Early Research Achievements (ERA) (up to 4 pages plus one for references).
* Replications and Negative Results (RENE) (up to 10 pages plus one for references or 4 pages plus one for references).
* Tool demonstration

The 33rd IEEE/ACM International Conference on Program Comprehension (ICPC) is the premier venue for work in the area of software program comprehension.

It encompasses human activities for comprehending the software as well as methodologies and technologies for supporting such comprehension.

The research track provides a quality forum for researchers and practitioners from academia, industry, and government to present and discuss

new results in program comprehension. Topics of interest for all tracks include but are not limited to:

-Empirical evaluations of program comprehension tools, techniques, and approaches;

-Human aspects in program comprehension, including collaborative software engineering practices, gender considerations, information processing strategies, the role of emotions, emotional awareness, and more;

-Cognitive theories for program comprehension accompanied by different empirical strategies, including experiments, surveys, and case studies;

-Topics at the intersection between program comprehension and programming education;

-Individual, collaborative, distributed, and global program comprehension;

-Novel visualization or summarization techniques and interfaces to support program comprehension, including searching, browsing, and analyzing;

-Comprehension of specific types of software systems, such as open/closed source, mobile applications, spreadsheets, web-based systems, legacy systems, product lines, libraries, multi-threaded applications, and systems of systems;

-Comprehension in the context of diverse software process models and specific lifecycle activities, such as: maintenance, evolution, re-engineering, migration, security, auditing, and testing;

-Comprehension of software artifacts ranging from requirements documents to test cases and crash logs; from API documentation to models, meta-models and model transformation; and from Stack Overflow questions & answers to GitHub code review messages or video tutorials : all software artifacts and formal or informal documentation that software developers encounter when creating or evolving software;

-Comprehension and legal issues, such as due diligence, intellectual property and licensing, reverse engineering, and litigation; Issues and case studies in the transfer of program comprehension technology to industry;

-Automated tool support for program comprehension.