IEEE Software solicits contributions to a special issue on New Directions in Programming Languages to be published in 2014. Computer programming as we know it today has a fairly long history. Fortran was one of the first programming languages created and is still used today. Why, though, isn’t all programming today done in Fortran and other languages of its era? One major reason is the diversification of those who want, or need to program. Consequently, programming languages have evolved to satisfy a diverse range of requirements for many different groups of programmers: from professionals working for large software companies, through biologists, chemists and physicists, to graphics designers and teenagers writing applications for Android or iPhone. Furthermore, everyone wants their software to run fast and modern programming languages must offer programmers the ability to maximize utilization of the underlying hardware, which today is often multi-core, many-core, or even heterogeneous.

In this special edition of IEEE Software, “What’s New in Programming Languages?” we are looking for contributions describing how modern programming languages cope with the challenges posed by varied requirements from different programmer groups, combined with the rapid evolution of hardware platforms.

All submissions must take the form of case studies of language use, design and/or implementation. Negative experiences (regarding features or implementations that did not work well in practice) are welcome, as are positive results. Well-suited submissions will show how cross-discipline issues (for example, software/hardware interactions; implications of programming languages for other disciplines) have been addressed in practice.

As long as they meet the requirements above, submissions may take on topics including, but not limited to:

- novel programming language features
- multiparadigm programming languages, such as Scala, C# or F#
- dynamic/scripting programming languages designed for productivity, such as JavaScript, Lua or Python
- domain-specific languages (DSLs)
- concurrent and parallel programming languages, such as Cilk, Clojure, Erlang or languages from the PGAS family
- programming language support for heterogeneity
- programming language construct and features aimed at simplifying today’s software development, for example, transactional memory or deterministic parallelism

Each article should clearly outline the problem to be addressed, the solution or the findings, (at least) a proof of concept, and the options for transferring the solution/findings into practice.

Questions?
For more information about the focus, contact the guest editors:
- Laurence Tratt (laurie@tratt.net)
- Adam Welc (adamwwelc@gmail.com)

Submission guidelines
Manuscripts must not exceed 4,700 words including figures and tables, which count for 200 words each. Submissions in excess of these limits may be rejected without refereeing. The articles we deem within the theme and scope will be peer-reviewed and are subject to editing for magazine style, clarity, organization, and space. We reserve the right to edit the title of all submissions. Be sure to include the name of the theme or special issue you are submitting for.

Articles should have a practical orientation and be written in a style accessible to practitioners. Overly complex, purely research-oriented or theoretical treatments are not appropriate. Articles should be novel. IEEE Software does not republish material published previously in other venues, including other periodicals and formal conference/workshop proceedings, whether previous publication was in print or in electronic form.

Full author guidelines:
www.computer.org/software/author.htm
Submission details:
software@computer.org
Submit an article:
https://mc.manuscriptcentral.com/sw-cs