

## Third essay: functional programming

As announced in class, your next homework assignment will be an essay over two articles, “Why Functional Programming Matters” and “Can functional programming be liberated from the von Neumann style?”. You will write a (roughly) 2 page opinion essay, where I would like you to comment on functional programming in general, particularly as it relates to your experience. Is this a useful paradigm, or are you glad that the structured paradigm dominated from the 70’s on in modern programming language design?

Also, beyond this, you’ve now seen two functional languages: Haskell is pretty purely functional, and Lisp is in the same family (if a bit less pure). After that, you saw functional programming in newer versions of C++ and Python, both of which began as imperative languages but more recently adopted functional features. This reflects a recent interest on functional programming, and what trends it can bring to software engineers and programmers.

A few relevant things to consider and perhaps address or answer in your essay: Has learning a functional language changed your programming style or views? How useful do you see the features in the other languages you saw with the substitutes? Do you feel functional languages are better or worse? Even if you don’t continue to use a functional language, is learning one a valuable experience in some way?

Please begin the essay with a VERY clear thesis statement (as this is an opinion piece) that concisely describes your opinion, and then use examples or information to back up your view. You are welcome to draw from our own use of Haskell and other languages covered in the class, or comment on other aspects of functional programming that you have experienced as well. I’m happy to see either love or hate towards this paradigm; my purpose is to get you to critically evaluate it and convey that opinion in a convincing manner!

For this paper, please be sure to actually read and use two papers given as reading, since your job is either to agree or disagree with their premise; because of this, you must reference both directly and in a meaningful way in your essay for full credit, including citations. You aren’t required to use other sources, although it may be helpful to do so; in particular, there is a lot of good (and sometimes highly amusing) commentary about this paradigm on various blogs and in various books. (Just google “I love functional programming” and then try “I hate functional programming” - tons of angry or odd people providing entertainment on various blogs and tech sites.) You are welcome to reference or quote any of these works, but please cite all sources that you do use or draw from in any way, as always.

Note: If you are registered for honors credit or the graduate section, you must write a longer version of this paper (5 pages total instead of 2), incorporating more of the material from the Lisp, C++ and Python bits that the sub covered, and also discussing how useful you see these features being.