

Essay 1

Introduction and overview: One large part of a professional job in computer science is communication at many levels, from technical discussions with peers to non-technical exposition to employers or other people outside of the field. In order to foster and improve this skill, this course will include several essays. Each essay in this class will be technical in nature, but about a quarter of your grade on each essay will also include grammar and professional writing. In addition to addressing the question asked, please also try to learn something new and to enjoy conveying this information! This ability to communicate technical ideas in an interesting and understandable tone is one of the one of the strongest skills you will use a future career.

Assignment details: For this assignment, you will write a 1-2 page paper on one of the following topics. I will expect at least 3 sources beyond your textbook, with proper citations for relevant facts from those references. Of course, for this assignment, you are welcome to use web or book resources of any type; just be sure to use reputable sources and include them in your bibliography.

Please note that since all of these topics have at least some component that is opinion-based, I expect a clear thesis statement which summarizes your opinion or finding, which must then be supported by relevant facts and reasonable arguments in the body of your essay.

(Also: If you have some other programming language related topic that you are dying to write about, feel free to come and discuss with me, since I'm open to similar types of essay topics which are relevant to the course. Just be sure to get prior approval before writing!)

- In recent years, there has been a resurgence of interest in functional programming as a paradigm. (We'll spend quite a bit of time learning Haskell later in the semester, so you'll hear more about this.) Go read a bit about functional programming as a paradigm if you aren't familiar with it, as well as the ways it is being incorporated into more "mainstream" objected oriented paradigms. What are some of the pros and cons of this paradigm? Do you think it's just a fad, or represents a major or important shift in language development?
- Consider the programming language with which you are most familiar, and list three things about it that you find frustrating or counterintuitive and wish had been differently designed. Now look into its history a bit: why was it was designed the way it was? How would you fix these issues if you had a chance to do it over? Would there be any negative consequences, for example in terms of compiler complexity or program execution speed, or do you think other languages show these changes are work it? Overall, do you think it is worth redesigning it, or are there good reasons to keep things they way they are?
- Familiarize yourself with the history of Java and C# using online (or other) sources, including the conflict between Sun and Microsoft over Java standardization. Some have claimed that C#, at least partially, is Microsoft's attempt to kill Java. Defend or refute this claim, with relevant facts to back up your opinion.
- C has a reputation for being a relatively "unsafe" high-level language. Why is this the case? Is this a feature, or a major design flaw? What utilities exist to enforce safer practices in C, and are these sufficient "patches" to C's security vulnerabilities, or does the language deserve its reputation?