

Review questions for final

Functional programming

1. What is a side effect, and why do functional languages not have them?
2. Give an example of how control flow is different in functional languages (particularly in Haskell). What type of statements in standard programming languages are not allowed in functional languages?
3. What is a higher order function? What is a first class object?
4. How is I/O accommodated in functional programming languages, since it is pretty much purely based on side effects?
5. What is a functor in Haskell?
6. How are types different in Haskell? Describe its type classes, and how they are different from object oriented classes.
7. Be prepared to code Haskell functions, at the level of one of our homework assignments.

Prolog

8. List a few applications of prolog, or things that it can do well.
9. What is unification, and how does Prolog attempt to do it?
10. What is a functor in Prolog?
11. How is a variable represented in Prolog? How are clauses formed?
12. Does the ordering of the clauses in a database matter in Prolog? Why or why not?
13. What is the cut (!) in prolog?

Concurrency

14. Why have parallel algorithm and multiprocessor support become so important in the last 20 years? (Give at least 2 reasons.)
15. What is the coherence problem in multiprocessor caches?
16. What is the difference between mutual exclusion and condition synchronization?
17. What is a race condition? What is a context switch?
18. What are the six principal operations that programming languages use to create new threads? Be able to explain roughly what each is, and how it works.
19. Why don't message passing programs require explicit synchronization?
20. What is a barrier, and what types of programs use them?
21. What is a semaphore? Describe what operations are supported.

COBOL

22. Name one place COBOL is still in use, and give several reasons.
23. What are the principle strengths and weaknesses of COBOL?
24. How are COBOL programs organized?

Scripting Languages

25. List the principal features or characteristics that scripting languages tend to exhibit? How are these different from more “conventional” languages?

26. List a few modern scripting tools, and tell how they implement specifically the characteristics you listed in the previous problem.

27. What are the two principal ancestors of modern scripting languages?

28. What are regular expressions and extended regular expressions, and which languages use them in particular?

29. What are some of the major languages used for mathematical computing, and what specific features do they incorporate that are useful?