CSc 372, Fall 2001
Assignment 8
Due: Friday, December 7, 8:00 a.m.

Overview

This assignment consists of a number of essay questions that you ask to discuss various topics related to the languages we have studied and other languages you may know. You are to choose 75 points worth of questions and answer them.

Questions

1. Slide #5 in the Introduction enumerates several ways in which programming languages can be evaluated. Identify another way in which a language can be evaluated and evaluate two languages on that basis. One of the languages must be among the four we have studied. (20 points)

2. A program's source code can be partitioned into code that uses core language features and code that simply calls library routines. Languages can be characterized by what percentage of the code of typical programs falls into each category. Identify two languages whose relative percentages differ strongly and discuss some ramifications of those differences. A good answer on this question will likely require rough measurements of the usage of language elements and library routines in a handful of programs. (15 points)

3. Once a language designer has made a decision to include an operation such as string concatenation in a language the next decision is whether the operation should appear as an operator or a function. For example, ML's string concatenation operator, `^`, could have been a function, perhaps `concat(s1, s2)`, instead. What characteristics of an operation lead it to be more naturally expressed as operator instead of a function, or vice versa? What other factors come into play when deciding what operations to provide as operators? (15 points)

4. Imagine a person who has learned Prolog as their first programming language and is now trying to learn Java. Try to put yourself in their position and describe some of the difficulties they might encounter in trying to learn Java. What things about Java would they find to be awkward, inadequate, and/or primitive? What features of Java would they be glad to see? (15 points)

   For this problem, #4, ignore the object-oriented and graphical aspects of Java and focus on the core language and non-graphical libraries.

5. Do the previous problem but instead of imagining Prolog as a first language, consider ML, Icon, or Emacs Lisp instead. (10 points)

6. An issue that language designers struggle with is what routines to include in the library accompanying a language. Some languages have very large libraries and some have very

small libraries. What guidelines should language designers follow when designing a library? What are some characteristics of a good library? (15 points)

7. An aspect of Icon that was not discussed in class is the ability to write programs that make use of Prolog-like backtracking to search for solutions to a problem. Using Icon's backtracking capabilities implement an Icon version of the Prolog brick laying example, `place.pl`, or `connect.pl` from the spring 1997 semester. Discuss similarities and differences between the backtracking facilities provided by the two languages. (15 points)

8. Java uses zero-based indexing for arrays and for classes like `String` and `Vector`. Icon uses 1-based indexing for strings and lists. Emacs Lisp uses 0-based indexing for functions like `nth`, `elt`, and `substring`, but 1-based indexing for buffer contents. Present an argument in favor of either 0-based or 1-based indexing for array-like objects. Your argument should include examples that put your preference in a positive light and the alternative in a negative light. (15 points)

9. Slide #8 in the Introduction cites several factors that can influence the popularity of a language. Identify another factor that you believe influences the popularity of a language and describe the positive and/or negative effects it might produce. (5 points per factor, up to three factors)

10. It can be interesting to measure how long it takes to perform a particular computation in various languages. Write a pair of programs that perform the same computation in two different languages, one of which must be a language we studied, and report the execution times of each.

    The computation may be simple in nature. For example, you might write an ML function and a Prolog predicate to create a long list. You might then try timing them for lists differing in lengths by binary orders of magnitude.

    As another example, you might write programs in Icon and Java to compute Ackerman's function for given values. That's an interesting test of function call efficiency.

    The computations should run for perhaps 5-15 seconds—long enough that differences in measured time between the two programs are due to true differences in running time, not just variations due to imprecise measurement. You may report either CPU time or wall-clock time but if you choose the latter, be sure to run when the system is lightly loaded. Report times with 1/10 second resolution and report at least three times for each case being tested. Include a copy of program source code along with the data collected.

11. You've perhaps found elements of ML, Icon, Prolog, and Emacs Lisp that you'd like to see in Java. Identify one such element and show how it might be incorporated into Java. Be sure to include some examples of usage. (5 points per element, up to three elements)

12. Present a sample of code for a language that is not currently in widespread use and by explaining how the code works, show some interesting elements of the language. (15 points)

13. Write a question similar in nature to the questions in this assignment. (5 points per question, up to three questions)

14. Answer a question posed by yourself. To undertake this problem you must ask the instructor to assign a point value to a proposed question

Miscellaneous

You may discuss these questions in groups of any size and make written note of ideas that arise in those discussions. However, the text you submit for grading must be written entirely by yourself. In other words, group discussions can be used to produce ideas and elements of a final answer but organization and expression of those ideas must be your own work.

None of these questions have clear-cut answers—your task is to simply put forth an answer that intelligently addresses each question that you choose to answer. A well-written answer will receive full credit, regardless of whether the views expressed agree with those of the instructor.

Answers will be graded on an A/B/C/D/F scale where an "A" will be worth 100% of the possible points for the problem. A "B" will be worth 85% of the points. A "C" will be worth 70%, a "D" 50%, and an "F" will be worth no points. Intermediate scores, such as "B+" and "A-" may be assigned.

This is not a writing class but if spelling errors, poor grammar, fragmentary sentences, etc., make an answer hard to understand, points may be lost for that.

If answers are submitted for more than 75 points worth of questions, the answers will be graded in turn until 75 points worth of questions have been encountered.

To reach 75 points exactly you may designate one question to be scored for less than full value. For example, you might answer the first five questions, which total to 80 points, and indicate that you'd like question 2 scored for 10 points instead of 15.

The pool of questions for this assignment may grow between now and the due date.

Deliverables

You may submit your answers as a plain text file or as a PostScript file. Use the name `answers.txt` for a plain text file.

Use the name `answers.ps` if you choose to submit a PostScript file. **Be sure** to verify that the file you submit is printable by first sending it to a PostScript printer with `lpr`.

Use the tag `372_8` for `turnin`.