

CIS 2.55 Final Exam

Each question is 5 points. You get 2 points for leaving a question blank. You get no points for a completely wrong answer. You get partial credit for a partially correct answer.

1. List some basic Perl data types. What does Perl stand for?
2. Given `@values`, write Perl code to find the mean and variance.
3. Given `@values`, write Perl code to reverse it (*do not* use any built in functions).
4. Declare an array of hashes of arrays of hashes (Don't write out pages and pages of values, just supply a few sample values for it).
5. For an html file `index.html`, write code to display just the text (without any tags).
6. Name some differences between Perl and C/C++. What can you do in Perl that you cannot do in C++? What can you do in C++ that you cannot do in Perl?
7. Note some of the strength of Perl. How is it different from traditional languages like Lisp? Haskell? Prolog? What does Perl have that C does not? Why would you use Perl as opposed to some other language?
8. Create code to build a Binary Search Tree, and add numbers: 5, 10, 7, 4, 2, 6, 9 to it. Tip: use hashes to represent nodes. You only need 'insert' subroutine.
9. Create code to traverse the above tree such that the elements are printed out sorted.
10. For an html file `index.html`, write code to replace every `<` with `[`, and every `>` with `]`, in effect replacing every html tag with `[tag]`.
11. You've just downloaded a file called `blah.html`. Write a Perl script to extract every e-mail from that file and place it in `emails.txt`, one e-mail per line.
12. You're not done with `blah.html`. Remove all the comments; those are the ones that start with `<!--` and end with `-->`. And then save every single link (the 'a' tag) in a file called `links.txt`.
13. For this question, you have access to the unix `mail` program. You have a text file of e-mails: `emails.txt` (one e-mail per line). You also have a `message.txt` file. Write a Perl script to send `message.txt` to every email in the `emails.txt` file.
14. Describe the process involved in creating a Perl object. How is it different from other languages?
15. Calculate the sum of squares from 1 to 1000. Tip: be very very lazy (don't use a loop). (2 points extra credit: same thing for sum of cubes, again, without using a loop.)
16. Write a subroutine called `swap` that would swap the first two arguments. Create a reference to subroutine `swap` and call it.
17. Write your own sorting routine (you may use 'sort' internally) that would accept an array, and a reference to a comparison function. The routine would return the sorted array.
18. What exactly does `bless` do? Explain.
19. Write code for `Person.pm`, an object that would have a name, and age—meaning that it will have set/get methods for name and age.
20. Write a *breadth-first* traversal of a binary tree (note that this is not the same as question 9).