

Constructing a Scheme Test...

V. 1.1

Quick details:

- Due: Thurs, Oct. 15, 8:00a on the homework server
- File formats: Upload your document, which must be in either .docx or .pdf format. If you'll be writing it on Google Docs, simply use the File menu's Download option to download it in .docx or .pdf format and then upload it to the homework server.
- Document-header should have your name and period.
- Test-instructions: any special instructions to the students should be clearly stated at the top of the test.
- Length: your hypothetical students would be given 30 minutes to take the test, so try to construct it accordingly. 6-8 questions of different types would be fine.
- Coverage: Your questions should try to cover as many of the ideas as possible that we have worked with, not just syntax or correct spelling of the functions.
- Answer-key: The correct answer(s) to each question should follow the question in a different typeface or color. Any hints for grading should also be there.

General suggestions:

- Don't test the same skill/concept at the same level of difficulty more than once.
- Cultural and/or timely references are fine as long as: not political, not obscene, does not fly over Mr. Brooks's head (well, that narrows it down).
- One or two easy questions (or easy multi-part questions) are good to start with, so that your students do not panic (initially).
- Some other suggestions (tend toward the easier part of the test):
 - fix syntactically broken code
 - add to missing part
 - correct a function that works but gives the wrong answer
- Creative questions make the test more human (a not-well-defined idea) and will be rewarded.
- A challenge or bonus problem is a good idea, to help those students making inadvertent mistakes on previous problems. But it should be a challenge.

Samples: here are a few questions from previous Scheme tests:

Very simple:

Write the Scheme code to evaluate the following expressions:

a. $1.855 - (12 - 1.9)$

b. $20 + \sqrt{\frac{35}{3+1.22}}$

c. Avogadro's number (6.023×10^{23}) divided by (eight to the fifth power)

More involved:

Create the function *AndOr23(N)* which will be given a positive integer and:

- Will return $2N+1$ if N is evenly divisible by both 2 and 3
- Will return 0 otherwise

Evaluate the following Scheme expressions (give the same answer that Scheme would compute) (show some work and circle your answers):

a. `(/ (* 3 3 2) (- 26 11))`

b. `(define (Hermione N) (modulo (quotient N 100) 100))`
`(Hermione 155396)`

c. Using the definition of Hermione from the question above...
`(define (Voldy Fred George) (modulo (Hermione Fred) George))`
`(Voldy 8641325 11)`

Create the function *Xor(A,B)* that takes 2 Booleans (truth values) and returns a truth value. It will execute the logical exclusive-or function whose truth table is below. Try for as short a definition as possible.

A	B	Xor(A,B)
#t	#t	#f
#t	#f	#t
#f	#t	#t
#f	#f	#f

Extra credit (not easy):

Create the function *IsPrime_10_40(N)* which will be given an integer between 10 and 40 and will return #t if N is a prime number, otherwise #f. You may NOT just test each actual prime between 10 and 40 with a separate test (that would be a painful exercise in parentheses). Hint: what might the composite (non-prime) numbers between 10 and 40 have in common, in terms of prime factors?