FIRST QUIZ

Please read these instructions carefully; they will apply for all our quizzes, but we won’t repeat them every time. You have 10 minutes from the start of class to complete this quiz.

Please read all the problems closely. If you have any questions on what a problem means, don’t hesitate to ask us. Don’t get bogged down on any one problem; if you have trouble on a problem, go on to the next one. Unless a problem specifically asks you to consider errors, you should assume that each problem is correct and solvable, and ask us if you believe otherwise.

Please write your answers clearly—we can’t give you credit if we can’t decipher what you’ve written. We’ll give partial credit for partially correct answers, so writing something is better than writing nothing. But no question requires an answer longer than two sentences, so don’t just write everything you know and hope that the right answer will be included somewhere; we will deduct points for needlessly long answers. Good luck!

Problem 1 (5 points)

Each of the following statements claims to be a policy, procedure, good advice, or other characteristic of Informatics 41, but each is inaccurate, misguided, or wrongheaded in some way. Please change each statement (as little as necessary) to make it an accurate statement on the same topic.

(a) Every student needs to purchase a paper copy of the Picturing Programs text and the How to Design Programs text.

(b) You’ll get the fastest response to course-related questions by sending electronic mail to three Email addresses: the instructor’s, the TA’s, and the reader’s.

(c) In pair programming, two programmers split up the work so they can finish twice as fast.

(d) If you find some course concepts difficult or confusing, or if you see people in the lab who work a lot faster than you do, you should give up and take another class.

(e) Informatics 41 has nothing to do with computer science; there’s a separate Computer Science department and major for that.

(continued on reverse)
Problem 2 (3 points)

Evaluate each of the following Scheme expressions. That is, what value does DrRacket display in the interactions window when you enter the expression or click Run? (We may not have said it explicitly, but besides + for addition, Scheme uses – for subtraction, * for multiplication, and / for division. Recall too that the function `string-length` takes a string as its input and returns the number of characters in the string; for example, `(string-length "Hello")` returns 5.)

(a)  `(+ 24 3)`  

(b)  `(* 20 (- 7 3))`  

(c)  `(/ (* 20 5) (+ 1 3))`  

(d)  `(+ (string-length "Autumn") 4)`

Problem 3 (2 points)

After evaluating the three Scheme expressions below, DrRacket displays one number. What is it?

```scheme
(define BREAD "rye")
(define FRUIT "peach")
(+ (string-length FRUIT) (string-length BREAD))
```