; Quiz 1: Commentary
; ===================

; For the first few weeks, a number of things affect many students:
; • enrolling late and needing time to catch up on practice
; • pre-conceptions from any previous exposure to programming
; • learning the importance of attention to detail in a CS (or Math or Science course), and that it requires active practice (not just passive watching of videos and lectures).
; • trying to infer meaning, as opposed to following our relatively small set of uniform rules in all situations

; We adjusted the marks to be out of 8, capped at 8, but still reported out of 10. Check you mark on MarkUs.
; Use your original mark seriously as feedback about what to work on.
; Pay attention to the TA's annotations even on questions you received full marks for.
; Blog about what any mistakes meant for you, and what you're now doing to make those mistakes (and kinds of mistakes) less likely to happen.

; Here's the adjustment function we used.
; adjust : number -> number
(define (adjust mark)
    (min (*/ (/ 10 8) mark) 10))

; Commentary on the individual questions
; ----------------------------------------
(+ 3 2 1) ; Generally well done.

(string-append "h" "mmm?"
; Most students rushed out of the quiz in five minutes.
; Did you rush and miss the question mark character, or did you see it and ignore it based on a misunderstanding of strings?
; Did you use quotes to show the result? Although we didn't take marks off for that, it's an important reminder of the datatype (vs being a variable name, or a number if the string happens to just contain digits). Some students quoted images and numbers in the second quiz, so some students not quoting the result here could be a sign of a deeper misunderstanding.
(< 1 2) ; Generally well done.
; Some students wrote True or TRUE, although we didn't take marks off for that. But note that case matters in DrRacket.
(define abc (string-append "h" "mmm?")
(string-length abc)
; The variable abc refers to the string "hmmm?", which contains five
; characters. We still gave you full marks if you counted based on an
; incorrect answer to (string-append "h" "mmm?").

(string-length 12345) ; 12345 is not a string, so this is an error.
(/ 2 1/3) ; Division of numbers is fine.

(< (< 1 2) 3) ; Doing the intermediate steps reveals the error.
; (< true 3) : true is not a number.

(+ "abc" "def") ; "abc" and "def" are not numbers, so this is an error.

; If you are used to other programming languages which have a lot of
; special case rules to memorize: we don't (some other languages do
; string-append when using + on strings, others will look at whether
; the strings contain digits and will treat them as numbers, some
; depend on whether it's a string and a number, etc).

(require picturing-programs)

(beside ○ □)
; ○□ : Avoid space between them, to emphasize it's one image.
; Some students put double-quotes around it: don't, it's not a string.

(above (beside ○ □) △)
; You worked out (beside ○ □), so now put that result above △: ○ □. 
; If you put your incorrect answer to (beside ○ □) above the triangle,
; you got full marks unless it made the question too different.

(above ○ □) ; □

(beside (above ○ □))
(above △ ★)
; Intermediate steps help (and notice you just did (above ○ □)):

(beside □ (above △ ★))
(beside △ □ ★)
(beside △ □ ★)