```
(require spd/tags)

;; Put the code up in Dr R.  Put questions up beside that.
;; note first question is LAST, then fix all others

;; Given the following code:

;; Constants:
(define SPEED 10)

;; Data definitions:
(@htdd Ball)
(define-struct ball (x y))
;; Ball is (make-ball Integer Integer)
;; interp. a ball with screen coordinates

(define B1 (make-ball 10 20))

(@dd-template-rules compound) ;2 fields

(define (fn-for-ball b)
  (... (ball-x b)   ;Integer
       (ball-x b))) ;Integer ;!!! ball-x should be ball-y

;; Functions:
(@htdf slide)
(@signature Ball -> Ball)
;; move ball down screen by SPEED
(check-expect (slide (make-ball 10 10))
              (make-ball 10 (+ 10 SPEED)))

;(define (slide b) b) ;stub

(@template-origin Ball)

(@template
 (define (slide b)
   (... (ball-x b)
        (ball-x b))))

(define (slide b)
  (make-ball (ball-x b)
             (+ (ball-x b) SPEED)))
```

```
;; QUESTION (120 seconds)
;; The LAST error is (first means the one that appears
;; first in the program text):
;;
;; A. In purpose for slide.
;; B. In the check-expects for slide.
;; C. In the @template-origin tag for slide.
;; D. In the @template for slide.
;; E. In the function definition for slide.
```

```
;; QUESTION [30 seconds]

;; The now first error is in:
;;
;; A. In the Constants.
;; B. In the type comment for Ball.
;; C. In purpose for slide.
;; D. In the check-expects for slide.
;; E. In the @template-origin tag for slide.
```

```
;; ERRORS
;;
;; Template has ball-x twice
;; @template has ball-x twice
;; Function has ball-x twice
;;
;; one test only, repeated values
```