```
(require spd/tags)

;; QUESTION [30 seconds]
;;
;; Which template would you use as the basis for a function
;; with this signature?
;;
(@signature Intersection -> TrafficLight)

;; A. The template from Intersection.
;; B. The template from TrafficLight.
;; C. An entirely new template would have to be created.
;; QUESTION [60 seconds]
;;
;; What is the next step in the evaluation of the following expression?

(cond [(> 3 4) "charlie"]
      [(= 6 6) "bravo"]
      [else    "alpha"])

;; A.
"bravo"

;; B.
[(= 6 6) "bravo"]


;; C.
(cond [false "charlie"]
      [(= 6 6) "bravo"]
      [else    "alpha"])


;; D.
(cond [(= 6 6) "bravo"]
      [else    "alpha"])
```

```
;; QUESTION [30 seconds]
;;
;; What is the next step in the evaluation of the following expression?

(cond [false "A"]
      [else "B"])

;; A. (cond ["A"]
;;          [else "B"])
;; B. (cond [else "B"])
;; C. "A"
;; D. "B"
;; QUESTION [90 seconds]
;;
;; There are two errors in the following function design. A single action
;; you are supposed to take when using the recipe (on a computer) would have
;; uncovered both of these errors.
;;

(@htdf circle-area)
(@signature Number -> Number)
;; Given radius compute circle area
(check-expect (circle-area 2) (* 3.14159 (sqr 2)))
(check-expect (circle-area 3 4) 2)

;(define (circle-area r) 0 ;stub

(@template-origin Number)

(define (circle-area r)
  (* 3.14159 (sqr r)))



;; The first error is (first means the one that appears first in the
;; program text):
;;
;; A. In the purpose.
;; B. In the signature.
;; C. In the stub.
;; D. In the check-expects.
;; QUESTION [20 seconds]
;;
;; There are two errors in the following function design. A single action
;; you are supposed to take when using the recipe (on a computer) would have
;; uncovered both of these errors.
;;
(@htdf circle-area)
(@signature Number -> Number)
;; Given radius compute circle area
(check-expect (circle-area 2) (* 3.14159 (sqr 2)))
(check-expect (circle-area 3 4) 2)

;(define (circle-area r) 0 ;stub

(@template-origin Number)

(define (circle-area r)
  (* 3.14159 (sqr r)))


;; The second error is:
;;
;; A. In the purpose.
;; B. In the signature.
;; C. In the stub.
;; D. In the check-expects.
;; QUESTION [30 seconds]
;;
;; There are two errors in the following function design. A single action
;; you are supposed to take when using the recipe (on a computer) would have
;; uncovered both of these errors.
;;
;; This design suggests that a key step was missed during the design process.
;; What was it?
(@htdf circle-area)
(@signature Number -> Number)
;; Given radius compute circle area
(check-expect (circle-area 2) (* 3.14159 (sqr 2)))
(check-expect (circle-area 3 4) 2)

;(define (circle-area r) 0 ;stub

(@template-origin Number)

(define (circle-area r)
  (* 3.14159 (sqr r)))


;; A. Checking that the signature matches the problem statement.
;; B. Checking that the purpose matches the signature.
;; C. Running the check-expects using the stub.
;; D. Commenting out the stub.
```