

Design Recipe Exercise Answer Key

target-leap

1. First example should be (EXAMPLE (target-leap 100) (* 100 2))
2. Second example should be (EXAMPLE (target-leap 40) (* 40 2))
3. In the definition, the name of the function should be "target-leap"
4. The variable name in the definition should be consistent (either x or x-coor).
5. The body of the function definition should be (* x-coor 2)

state-tax

1. The domain for the function should be "string number," to account for both the state and the price of the item
2. The function name in both examples should be "state-tax"
3. The example inputs ("Delaware" and "Georgia") should be strings
4. Examples should include a numerical price instead of the variable name "price"
5. The examples should use "*" not "+"

offscreen?

1. The Purpose Statement should include "return true is the coordinate is less than -50 and greater than 690"
2. Both examples should show the work, not just the answer:
 - a. (EXAMPLE (offscreen? 60) (or (< 60 -50) (> 60 690)))
 - b. (EXAMPLE (offscreen? 800) (or (< 8000 -50) (> 800 690)))
3. The function definition should use the function "or" instead of "and"

long-name?

1. Both examples should use the function "string-length," not "string=?"
2. The examples should check if the name is longer than 20 characters, not 10
3. The function name in the definition should be "long-name?"
4. The body of the function should be (< (string-length name) 20).

late-to-class?

1. Both examples should include 4 numbers as inputs
2. In the first example, "<" should be used in place of ">"

flower-name

1. The second example should be (EXAMPLE (flower-name "purple") "tulip")

equal-length?

1. The function body should be (= (string-length string1) (string-length string2))

enough-cash?

1. The domain of the function should be a number (representing the price), not a string.
2. The two examples should give numbers as an input and test if they are less than 1.50. For instance, (EXAMPLE (enough-cash? 4.50) (< 4.50 1.50))
3. The variable name in the function body can be "item", but a more accurate name would be price or cost.

enough-carpet?

1. The range of the function should be a boolean
2. The example inputs should not be in parentheses
3. Both the examples and the function definition should use "<=" instead of "<"

circle-area

1. The function body should be (* (sqr (/ diameter 2)) (/ 22 7))

check-total

1. The examples should use the function name "check-total" instead of "total"
2. Both Examples have too many parentheses for the function call:
(EXAMPLE ((total 20))...) should be (EXAMPLE (total 20)...)
3. The * function must come before its inputs in the examples:
(EXAMPLE (+ (* 0.20 56.67) 56.67))
4. The function body should have the * and + functions reversed:
(+ (* 0.20 food-total) food-total)

scale-image

1. The purpose statement doesn't specify what strings matter, or how much to scale by
The Examples do not use the scale function at all, and instead change the parameters of the image. The first Example should be:
(EXAMPLE (scale-image (circle 5 "solid" "red") "bigger")
 (scale 2 (circle 5 "solid" "red")))
2. The function name in the second example is incorrect