The Design Recipe

Directions: Write a function `moving` that takes in the days and number of miles driven and returns the cost of renting a truck. The truck is $55 per day and each driven mile is 15¢.

Contract and Purpose Statement

Every contract has three parts...

\#   moving::   Number, Number   ->   Number

     function name   *domain*   *range*

\# Takes in a number of days and multiplies it by $45, then takes in a number of miles and multiplies it by $0.15, then adds the two products and returns the cost of moving

Examples

what does the function do?

Write some examples, then circle and label what changes...

examples:

     moving (1, 600   ) **is** (1 * 55) + (600 * 0.15)

     function name   *input(s)*   *what the function produces*

     moving (3, 1500   ) **is** (3 * 55) + (1500 * 0.15)

     function name   *input(s)*   *what the function produces*

end

Definition

Write the definition, giving variable names to all your input values...

fun   moving(days, miles):

     function name   *variable(s)*

   (days * 55) + (miles * 0.15)

             what the function does with those variable(s)

end

Directions: Write a function `lawn-area` that takes in the length and width of a rectangular lawn and returns its area.

Contract and Purpose Statement

Every contract has three parts...

\#   lawn-area::   Number, Number   ->   Number

     function name   *domain*   *range*

\# Takes in 2 numbers, length and width, and multiplies them and returns that value

what does the function do?

Examples

Write some examples, then circle and label what changes...

examples:

     lawn-area (10, 20   ) **is** 10 * 20

     function name   *input(s)*   *what the function produces*

     lawn-area (100, 300   ) **is** 100 * 300

     function name   *input(s)*   *what the function produces*

end

Definition

Write the definition, giving variable names to all your input values...

fun   lawn-area(length, width):

     function name   *variable(s)*

   length * width

             what the function does with those variable(s)

end