Directions: Write a function that takes the target's x-coordinate and makes a player leap by returning an x-coordinate that is double the original x-coordinate.

Contract and Purpose Statement

Every contract has three parts…

; `target-leap` : `Number` -> `Number`

function name | *domain* | *range*

; Takes the x-coordinate and returns a new one, multiplied by 2.

what does the function do?

Examples

Write some examples, then circle and label what changes…

(EXAMPLE (`target-leap` `100`) `200`)

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

(EXAMPLE (`target-leap` `40`) `200`)

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

Definition

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

(define (`leap` `x-coor`)

function name | *variable(s)*

(* x 5))

what the function does with those variable(s)

Directions: Write a function, offscreen?, which returns true is Sam the butterfly's x-coordinate is less than -50 or greater than 690.

Contract and Purpose Statement

Every contract has three parts…

; `offscreen?` : `Number` -> `Boolean`

function name *domain* *range*

; Given an x-coordinate, returns true if the coordinate is less than -50

what does the function do?

Examples

Write some examples, then circle and label what changes…

(EXAMPLE (`offscreen?` `60`) `true`)

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

(EXAMPLE (`offscreen?` `800`) `false`)

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

Definition

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

(define (`off-screen` `x-coord`)

function name *variable(s)*

```
(and (< x-coord -50) (> x-coord 690))                                    )
```

what the function does with those variable(s)

Directions: All students are given five (5) pencils at the beginning of the school year. Write a function called calc-pencils that takes in the number of students in the school and calculates the number of pencils needed for that school.

Contract and Purpose Statement

Every contract has three parts…

; calc-pencils : Number -> Number

function name *domain* *range*

; Takes a number of students and gives the number of pencils

what does the function do?

Examples

Write some examples, then circle and label what changes…

(EXAMPLE (calc-pencils 100) (* 100 5))

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

(EXAMPLE (calc-pencils 40) (* 40 6))

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

Definition

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

(define (calculate-pencils p)

function name *variable(s)*

(* p 5))

what the function does with those variable(s)

Directions: Write a function that returns the area of a circle given its diameter.

Contract and Purpose Statement

Every contract has three parts…

; circle-area : Number -> Number

function name | *domain* | *range*

; Given the diameter, multiply pi by radius squared to get the area

what does the function do?

Examples

Write some examples, then circle and label what changes…

(EXAMPLE (circle-area 10) (* (sqr (/ 10 2)) pi))

(EXAMPLE (circle-area 50) (* (sqr (/ 50 2)) pi))

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

Definition

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

(define (area diameter)

function name | *variable(s)*

(* (sqr diameter) pi)

what the function does with those variable(s)

Directions: It is customary to tip 20% on a bill at a restaurant. Write a function that takes the total cost of the food and returns the new total including tip.

Contract and Purpose Statement

Every contract has three parts…

; check-total : Number -> Number

function name *domain* *range*

; Returns the total of a check with 20% of the cost added

what does the function do?

Examples

Write some examples, then circle and label what changes…

(EXAMPLE (total 20) 20 + (0.2 * 20))

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

(EXAMPLE (total 56.67) 56.67 + (0.2 * 56.67))

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

Definition

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

(define (check-total food-total)

function name *variable(s)*

(* (+ 0.2 food-total) food-total))

what the function does with those variable(s)

Directions: You have 100 square feet of carpet to put down in your room. Write a function that takes in the length and width of your room and returns true if you have enough carpet and false if you don't.

Contract and Purpose Statement

Every contract has three parts…

; enough-carpet?: Number Number -> Number

function name *domain* *range*

; Given length and width of a room, is the area <= 100 sq feet?

what does the function do?

Examples

Write some examples, then circle and label what changes…

(EXAMPLE (enough-carpet? (10 15)) (< (* 10 15) 100))

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

(EXAMPLE (enough-carpet? (9 10)) (< (* 9 10) 100))

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

Definition

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

(define (enough-carpet? length width)

function name *variable(s)*

(< (* length width) 100))

what the function does with those variable(s)

Directions: You go to the store with $1.50 in your pocket. Write a function that takes in the price of an item and returns true if you have enough money to buy the item and false if you do not.

Contract and Purpose Statement

Every contract has three parts…

; `enough-cash?` : `String` -> `Boolean`

function name | *domain* | *range*

; Check to see if the item costs less than 1.50

what does the function do?

Examples

Write some examples, then circle and label what changes…

(EXAMPLE (`enough-cash?` `2.5`) `(>= 1.5 2.5)`)

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

(EXAMPLE (`enough-cash?` `9.0`) `(< gum 150)`)

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

Definition

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

(define (`enough-cash?` `item`)

function name | *variable(s)*

```
(<= item 1.5)
```
)

what the function does with those variable(s)

Directions: Write a function that takes in two strings and returns true if their lengths are equal and false otherwise.

Contract and Purpose Statement

Every contract has three parts…

```
; equal-length? : String String -> Boolean
```

function name | *domain* | *range*

```
; Given two strings, check if they are the same length
```

what does the function do?

Examples

Write some examples, then circle and label what changes…

```
(EXAMPLE (equal-length? "yes" "no"))
  (= (string-length "yes") (string-length "no")))
```

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

```
(EXAMPLE (equal-length? "dog" "cat"))
  (= (string-length "dog") (string-length "cat")))
```

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

Definition

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

```
(define (equal-length? string1 string2)
  (=))
```

function name | *variable(s)* | *what the function does with those variable(s)*

Directions: You are putting together a list of flowers in your garden based on their color. You have red roses, purple tulips, and yellow daisies. Write a function that takes in the color of a flower and returns the name of the flower.

Contract and Purpose Statement

Every contract has three parts…

```
; flower-name : String -> String
```

function name | *domain* | *range*

```
; Takes the name of the flower and returns its color
```

what does the function do?

Examples

Write some examples, then circle and label what changes…

```
(EXAMPLE (flower-name "red") "rose")
(EXAMPLE (flower-name "tulip") "purple")
(EXAMPLE (flower-name "yellow") "daisy")
```

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

Definition

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

```
(define (flower-name color)
  (cond
    [(string=? color "red") "rose"]
    [(string=? color "purple") "tulip"]
    [(string=? color "yellow") "daisy"]
    [else "That flower isn't in the garden!"]
  ))
```

function name | *variable(s)*

what the function does with those variable(s)

Directions: Names that are longer than 20 characters are considered long names. Write a function that takes in a person's name and returns true if it is a long name and false if it is not.

Contract and Purpose Statement

Every contract has three parts…

```
; long-name?   :   String   ->   Boolean
```

function name | *domain* | *range*

```
; Check if a name is longer than 20 characters
```

what does the function do?

Examples

Write some examples, then circle and label what changes…

```
(EXAMPLE (long-name?   "John Joseph Jingleheimer Schmidt")
```

function name | *input(s)*

```
  (> (string=? "John Joseph Jingleheimer Schmidt")
  10)
(EXAMPLE (long-name?   "Jos Jaime Juarez")  10)
```

what the function produces

```
  (> (string=? "Jos Jaime Juarez")
  10)
)
```

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

Definition

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

```
(define (long?   name  )
```

function name | *variable(s)*

```
  (< name 20)                                                    )
```

what the function does with those variable(s)

Directions: Write a function that takes an image and a string, representing what to scale the image by. The function should return a smaller image if the string is 'smaller' and a bigger image if the string is 'bigger'.

Contract and Purpose Statement

Every contract has three parts…

```
; scale-image : Image String -> image
```

function name | *domain* | *range*

```
; Make the image bigger or smaller, depending on the given string
```

what does the function do?

Examples

Write some examples, then circle and label what changes…

```
(EXAMPLE (scale-image (circle 5 "solid" "red") "bigger")
         (circle 10 "solid" "red"))
(EXAMPLE (scale-image (triangle 20 "solid" "blue") "smaller")
         (triangle 10 "solid" "blue"))
```

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

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

Definition

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

```
(define (scale-image original-image scale-factor)
  (cond
    [(string=? scale-factor "bigger") (scale 2 original-image)]
    [(string=? scale-factor "smaller")
     (scale 0.5 original-image)]
    [else original-image]
  ))
```

function name | *variable(s)* | *what the function does with those variable(s)*

Directions: Some states have different tax rates. New York is 8%, Pennsylvania is 3%, and Delaware is 0%. All other states are 5%. Write a function that takes in the price of an item and returns how much the tax will be on the item.

Contract and Purpose Statement

Every contract has three parts…

```
; state-tax : String -> Number
```
function name / *domain* / *range*

```
; Given the state and an item's price, return the tax on that item
```
what does the function do?

Examples

Write some examples, then circle and label what changes…

```
(EXAMPLE (tax Delaware) (+ 0.0 price))
```
function name / *input(s)* / *what the function produces*

```
(EXAMPLE (tax Georgia) (+ 0.05 price))
```
function name / *input(s)* / *what the function produces*

Definition

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

```
(define (state-tax state price)
```
function name / *variable(s)*

```
  (cond
    [(string=? state "Pennsylvania")  (* 0.03 price)]
    [(string=? state "New York")      (* 0.08 price)]
    [(string=? state "Delaware")      (* 0.0 price)]
    [else                             (* 0.05 price)]
  ))
```
what the function does with those variable(s)

Directions: You will be late to class if you have to walk more than 25 pixels to get there. Write a function that takes in your x-coordinate and y-coordinate and the x-coordinate and y-coordinate of the classroom and returns true if you will be late to class and false if you will be on time.

Contract and Purpose Statement

Every contract has three parts…

; late-to-class?: Number Number Number Number -> Boolean

function name | *domain* | *range*

; Takes the coorindates of my location and a classroom and returns true if the distance is more than 25 and false if it is less than 25.

what does the function do?

Examples

Write some examples, then circle and label what changes…

(EXAMPLE (late-to-class? 40 55) (> 25 (distance 40 55)))

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

(EXAMPLE (late-to-class? 40 55) (< 25 (distance 40 55)))

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

Definition

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

(define (late-to-class? student-x student-y school-x school-y)

function name | *variable(s)*

(< 25 (distance student-x student-y))

what the function does with those variable(s)