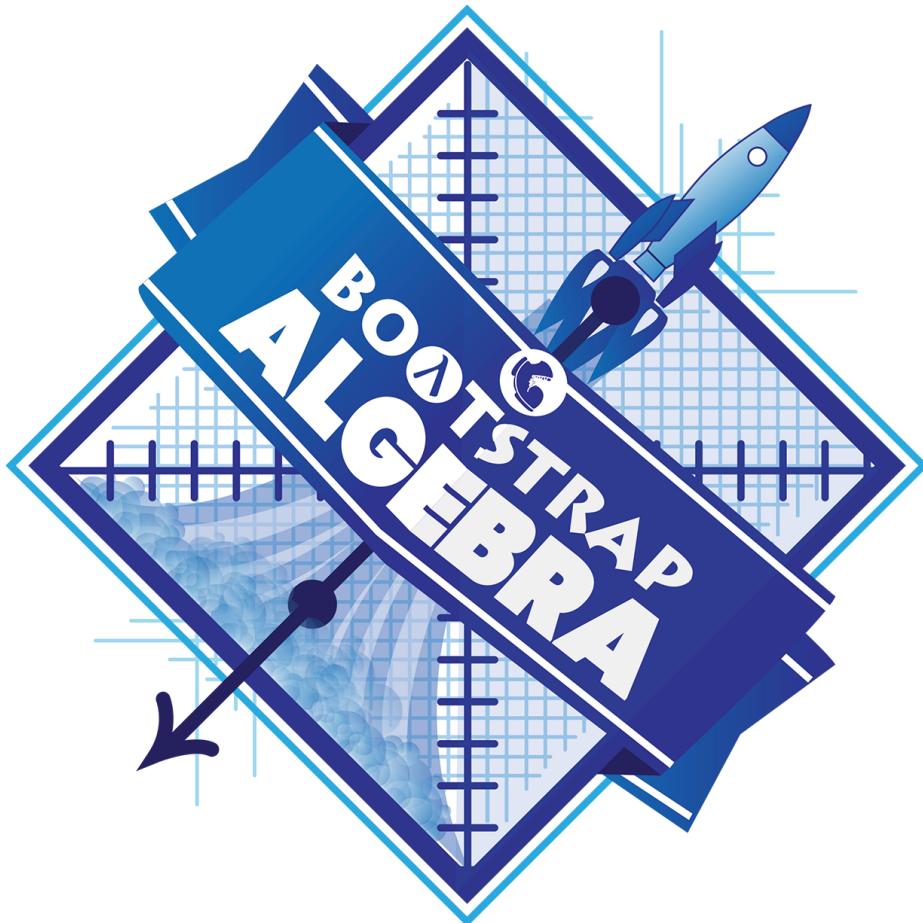


Name: \_\_\_\_\_



## Student Workbook

Class: \_\_\_\_\_



Workbook v2.7

Brought to you the Bootstrap team:

- Emmanuel Schanzer
- Kathi Fisler
- Shriram Krishnamurthi
- Emma Youndtsmith
- Rosanna Sobota

Visual Design: Colleen Murphy

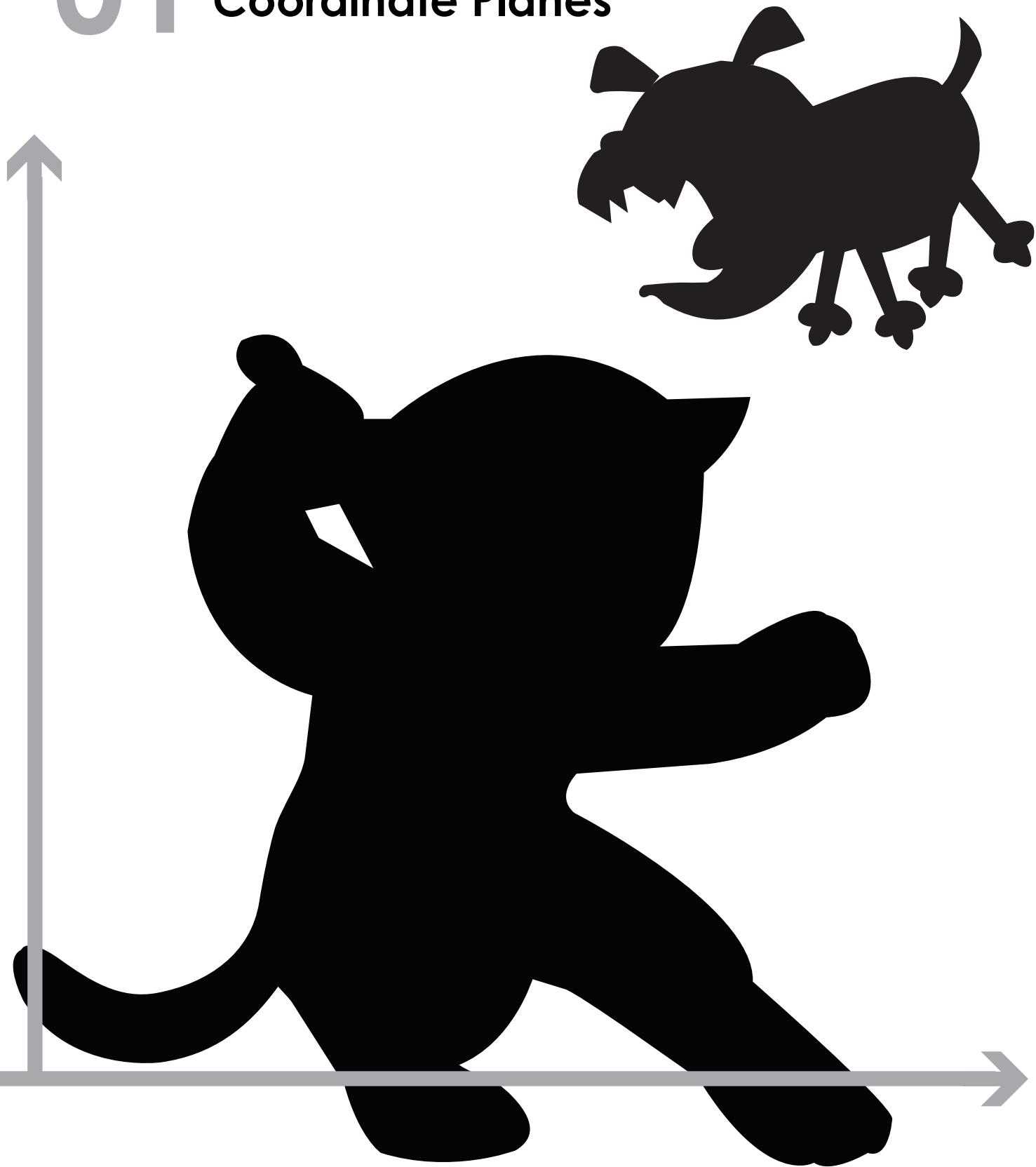
---

Bootstrap is licensed under a Creative Commons 3.0 Unported License. Based on a work from [www.BootstrapWorld.org](http://www.BootstrapWorld.org). Permissions beyond the scope of this license may be available at [schanzer@BootstrapWorld.org](mailto:schanzer@BootstrapWorld.org).

# Bootstrap Units

|           |   |           |                          |
|-----------|---|-----------|--------------------------|
| <b>01</b> | Videogames<br>and<br>Coordinate<br>Planes | <b>06</b> | Comparing<br>Functions   |
| <b>02</b> | Contracts,<br>Strings, and<br>Images      | <b>07</b> | Conditional<br>Branching |
| <b>03</b> | Intro to<br>Definitions                   | <b>08</b> | Collision<br>Detection   |
| <b>04</b> | Design Recipe                             | <b>09</b> | Prepping for<br>Launch   |
| <b>05</b> | Game<br>Animation                         | <b>10</b> | Additional<br>Material   |

# 01 Videogames and Coordinate Planes



# Lesson 1

Reverse-Engineering: How does NinjaCat work?

## Finding Coordinates



The coordinates for the PLAYER (NinjaCat) are:  $(\underline{\hspace{2cm}}, \underline{\hspace{2cm}})$   
x-coordinate      y-coordinate

The coordinates for the DANGER (Dog) are:  $(\underline{\hspace{2cm}}, \underline{\hspace{2cm}})$

The coordinates for the TARGET (Ruby) are:  $(\underline{\hspace{2cm}}, \underline{\hspace{2cm}})$

# Our Videogame

Created by (write your names): \_\_\_\_\_

## Background

Our game takes place in: \_\_\_\_\_  
(space? the desert? a mall?)

## The Player

*The player is a \_\_\_\_\_.*

The player moves only up and down.

## The Target

*Your player GAINS points when they hit the target.*

*The Target is a \_\_\_\_\_.*

The Target moves only to the left and right.

## The Danger

*Your player LOSES points when they hit the danger.*

*The Danger is a \_\_\_\_\_.*

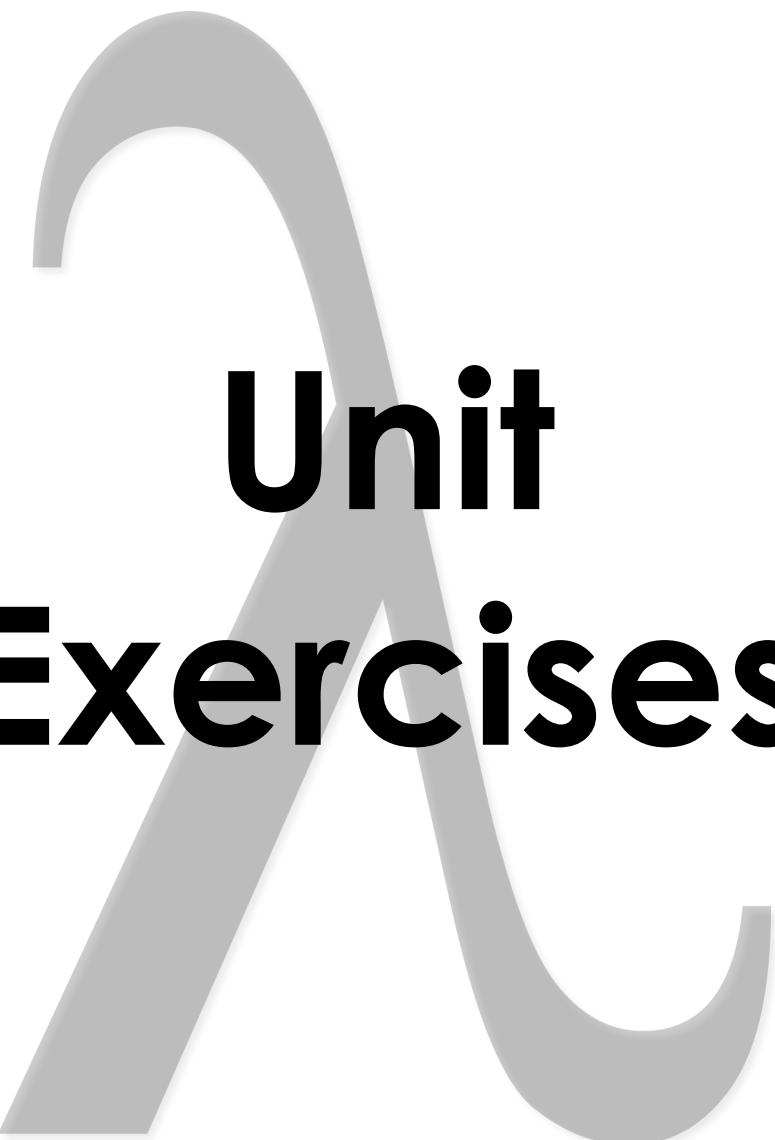
The Danger moves only to the left and right.

## Circle of Evaluation Practice

Time: 5 minutes

Don't forget to use the computer's symbols for things like multiply and divide!

| Math                        | Circle of Evaluation | Racket Code |
|-----------------------------|----------------------|-------------|
| $5 \times 10$               |                      |             |
| $8 + (5 \times 10)$         |                      |             |
| $(8 + 2) - (5 \times 10)$   |                      |             |
| $\frac{5 \times 10}{8 - 2}$ |                      |             |



# **Unit Exercises**

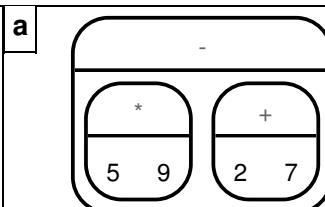
Name:

Date:

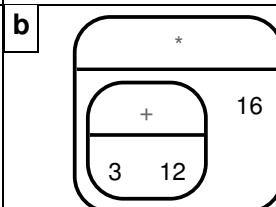
# Exercise: Completing Partial Circles of Evaluation from Arithmetic Expressions

**Directions:** For each arithmetic expression on the left, finish the Circle of Evaluation on the right by filling in the blanks.

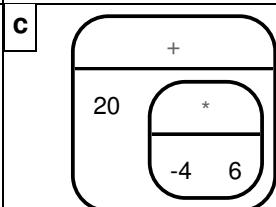
1  $(5 * 9) - (2 + 7)$



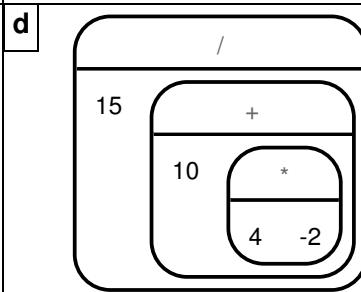
2  $(3 + 12) * 16$



3  $20 + (-4 * 6)$



4  $\frac{15}{10 + (4 * -2)}$



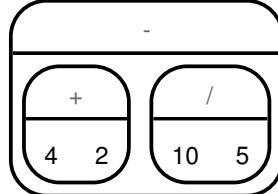
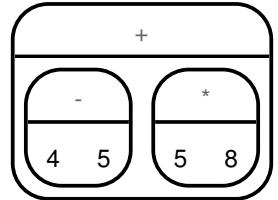
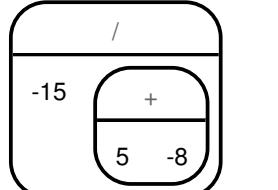
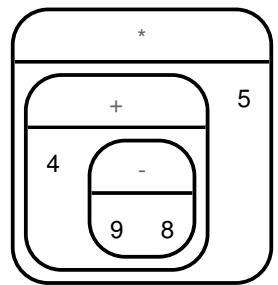
Bootstrap:Algebra by Emmanuel Schanzer, Emma Youndtsmith, Kathi Fisler, and Shriram Krishnamurthi is licensed under a Creative Commons 4.0 Unported License. Based on a work at [www.BootstrapWorld.org](http://www.BootstrapWorld.org). Permissions beyond the scope of this license may be available by contacting [schanzer@BootstrapWorld.org](mailto:schanzer@BootstrapWorld.org).

Name: \_\_\_\_\_

Date: \_\_\_\_\_

# Exercise: Completing Partial Circles of Evaluation from Arithmetic Expressions

**Directions:** For each arithmetic expression on the left, finish the Circle of Evaluation on the right by filling in the blanks.

|   |   |  |
|---|---|--|
| 1 | a |    |
| 2 | b |    |
| 3 | c |   |
| 4 | d |  |



Bootstrap:Algebra by Emmanuel Schanzer, Emma Youndtsmith, Kathi Fisler, and Shriram Krishnamurthi is licensed under a Creative Commons 4.0 Unported License. Based on a work at [www.BootstrapWorld.org](http://www.BootstrapWorld.org). Permissions beyond the scope of this license may be available by contacting [schanzer@BootstrapWorld.org](mailto:schanzer@BootstrapWorld.org).

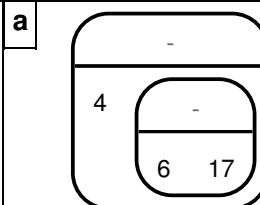
Name: \_\_\_\_\_

Date: \_\_\_\_\_

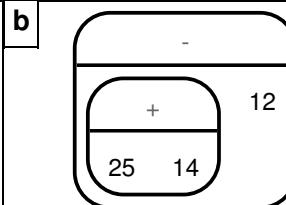
# Exercise: Creating Circles of Evaluation from Arithmetic Expressions

**Directions:** For each math expression on the left, draw its Circle of Evaluation on the right.

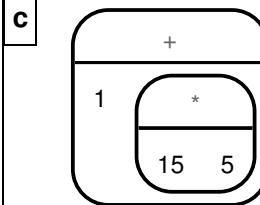
1  $4 - (6 - 17)$



2  $(25 + 14) - 12$



3  $1 + (15 * 5)$



Bootstrap:Algebra by Emmanuel Schanzer, Emma Youndtsmith, Kathi Fisler, and Shriram Krishnamurthi is licensed under a Creative Commons 4.0 Unported License. Based on a work at [www.BootstrapWorld.org](http://www.BootstrapWorld.org). Permissions beyond the scope of this license may be available by contacting [schanzer@BootstrapWorld.org](mailto:schanzer@BootstrapWorld.org).

Name: \_\_\_\_\_

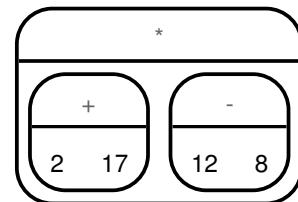
Date: \_\_\_\_\_

# Exercise: Creating Circles of Evaluation from Arithmetic Expressions

**Directions:** For each math expression on the left, draw its Circle of Evaluation on the right.

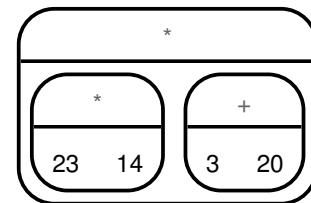
1

a



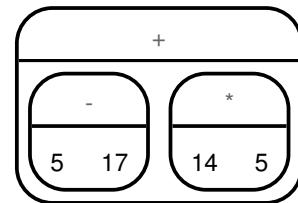
2

b



3

c



Bootstrap:Algebra by Emmanuel Schanzer, Emma Youndtsmith, Kathi Fisler, and Shriram Krishnamurthi is licensed under a Creative Commons 4.0 Unported License. Based on a work at [www.BootstrapWorld.org](http://www.BootstrapWorld.org). Permissions beyond the scope of this license may be available by contacting [schanzer@BootstrapWorld.org](mailto:schanzer@BootstrapWorld.org).

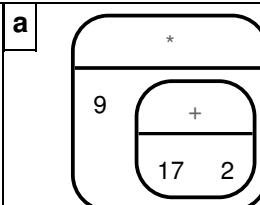
Name:

Date:

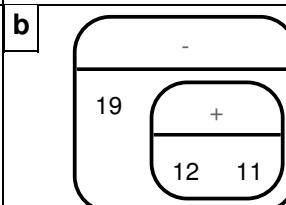
# Exercise: Creating Circles of Evaluation from Arithmetic Expressions

**Directions:** For each math expression on the left, draw its Circle of Evaluation on the right.

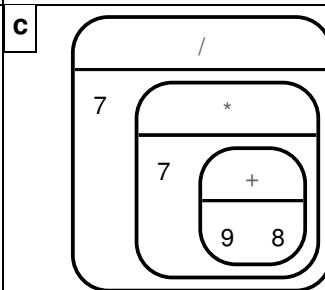
1  $9 * (17 + 2)$



2  $19 - (12 + 11)$



3  $\frac{7}{7 * (9+8)}$



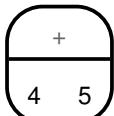
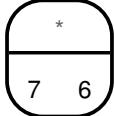
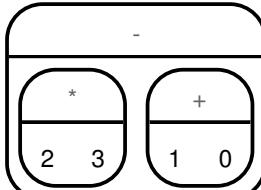
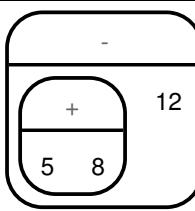
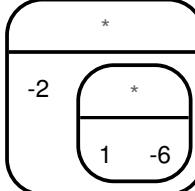
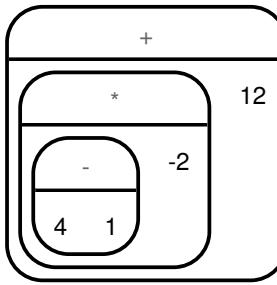
Bootstrap:Algebra by Emmanuel Schanzer, Emma Youndtsmith, Kathi Fisler, and Shriram Krishnamurthi is licensed under a Creative Commons 4.0 Unported License. Based on a work at [www.BootstrapWorld.org](http://www.BootstrapWorld.org). Permissions beyond the scope of this license may be available by contacting [schanzer@BootstrapWorld.org](mailto:schanzer@BootstrapWorld.org).

Name: \_\_\_\_\_

Date: \_\_\_\_\_

# Exercise: Converting Circles of Evaluation to Arithmetic Expressions

**Directions:** For each Circle of Evaluation on left, write the Arithmetic expression on the right:

|   |                                |
|---|--------------------------------|
| <b>1</b><br>   | <b>a</b> $4 + 5$               |
| <b>2</b><br>   | <b>b</b> $7 * 6$               |
| <b>3</b><br>   | <b>c</b> $(2 * 3) - (1 + 0)$   |
| <b>4</b><br>  | <b>d</b> $(5 + 8) - 12$        |
| <b>5</b><br> | <b>e</b> $-2 * (1 * -6)$       |
| <b>6</b><br> | <b>f</b> $((4 - 1) * -2) + 12$ |



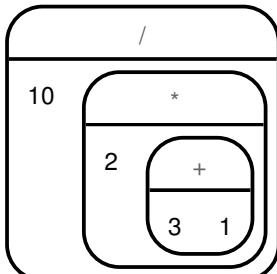
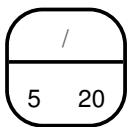
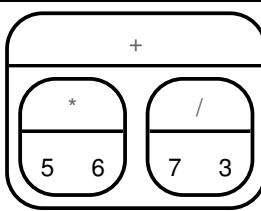
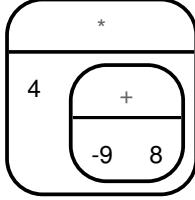
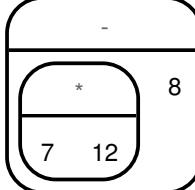
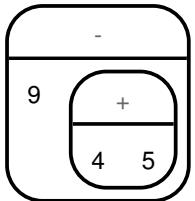
Bootstrap:Algebra by Emmanuel Schanzer, Emma Youndtsmith, Kathi Fisler, and Shriram Krishnamurthi is licensed under a Creative Commons 4.0 Unported License. Based on a work at [www.BootstrapWorld.org](http://www.BootstrapWorld.org). Permissions beyond the scope of this license may be available by contacting [schanzer@BootstrapWorld.org](mailto:schanzer@BootstrapWorld.org).

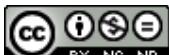
Name: \_\_\_\_\_

Date: \_\_\_\_\_

# Exercise: Converting Circles of Evaluation to Arithmetic Expressions

**Directions:** For each Circle of Evaluation on left, write the Arithmetic expression on the right:

|   |                                  |
|---|----------------------------------|
| <b>1</b><br>   | <b>a</b> $\frac{10}{2 * (3+1)}$  |
| <b>2</b><br>   | <b>b</b> $\frac{5}{20}$          |
| <b>3</b><br>  | <b>c</b> $(5 * 6) + \frac{7}{3}$ |
| <b>4</b><br> | <b>d</b> $4 * (-9 + 8)$          |
| <b>5</b><br> | <b>e</b> $(7 * 12) - 8$          |
| <b>6</b><br> | <b>f</b> $9 - (4 + 5)$           |



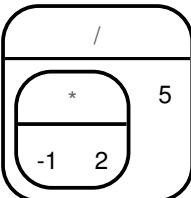
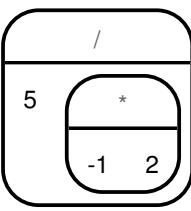
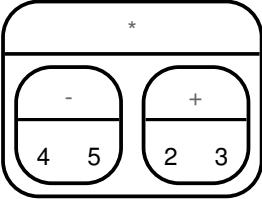
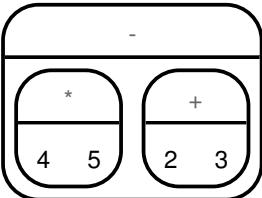
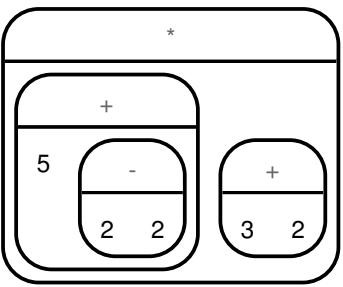
Bootstrap:Algebra by Emmanuel Schanzer, Emma Youndtsmith, Kathi Fisler, and Shriram Krishnamurthi is licensed under a Creative Commons 4.0 Unported License. Based on a work at [www.BootstrapWorld.org](http://www.BootstrapWorld.org). Permissions beyond the scope of this license may be available by contacting [schanzer@BootstrapWorld.org](mailto:schanzer@BootstrapWorld.org).

Name: \_\_\_\_\_

Date: \_\_\_\_\_

# Exercise: Matching Circles of Evaluation and Arithmetic Expressions

**Directions:** Draw a line from each Circle of Evaluation in the left to its corresponding arithmetic expression on the right:

|   |   |
|---|---|
| <b>1</b><br>   | <b>b</b><br>$\begin{array}{r} -1 * 2 \\ \hline 5 \end{array}$ |
| <b>2</b><br>   | <b>e</b><br>$\begin{array}{r} 5 \\ \hline -1 * 2 \end{array}$ |
| <b>3</b><br>  | <b>c</b><br>$(4 - 5) * (2 + 3)$                               |
| <b>4</b><br> | <b>a</b><br>$(4 * 5) - (2 + 3)$                               |
| <b>5</b><br> | <b>d</b><br>$(5 + (2 - 2)) * (3 + 2)$                         |



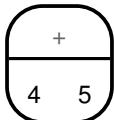
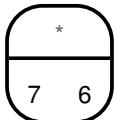
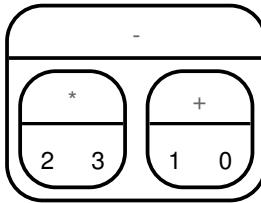
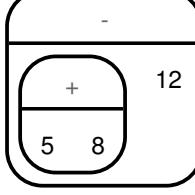
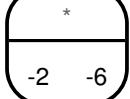
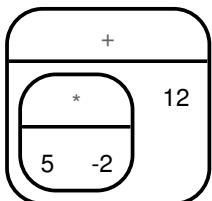
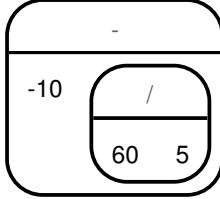
Bootstrap:Algebra by Emmanuel Schanzer, Emma Youndtsmith, Kathi Fisler, and Shriram Krishnamurthi is licensed under a Creative Commons 4.0 Unported License. Based on a work at [www.BootstrapWorld.org](http://www.BootstrapWorld.org). Permissions beyond the scope of this license may be available by contacting [schanzer@BootstrapWorld.org](mailto:schanzer@BootstrapWorld.org).

Name: \_\_\_\_\_

Date: \_\_\_\_\_

# Exercise: Evaluating Circles of Evaluation

**Directions:** For each Circle of Evaluation on the left, evaluate the Circle and write down the ANSWER (a number) on the right:

|   |              |
|---|--------------|
| <b>1</b><br>   | <b>a</b> 9   |
| <b>2</b><br>   | <b>b</b> 42  |
| <b>3</b><br>   | <b>c</b> 5   |
| <b>4</b><br>  | <b>d</b> 1   |
| <b>5</b><br> | <b>e</b> 12  |
| <b>6</b><br> | <b>f</b> 2   |
| <b>7</b><br> | <b>g</b> -22 |



Bootstrap:Algebra by Emmanuel Schanzer, Emma Youndtsmith, Kathi Fisler, and Shriram Krishnamurthi is licensed under a Creative Commons 4.0 Unported License. Based on a work at [www.BootstrapWorld.org](http://www.BootstrapWorld.org). Permissions beyond the scope of this license may be available by contacting [schanzer@BootstrapWorld.org](mailto:schanzer@BootstrapWorld.org).

Name: \_\_\_\_\_

Date: \_\_\_\_\_

# Exercise: Evaluating Circles of Evaluation

**Directions:** For each Circle of Evaluation on the left, evaluate the Circle and write down the ANSWER (a number) on the right:

|   |  |      |
|---|--|------|
| 1 |  | a 9  |
| 2 |  | b 42 |
| 3 |  | c 5  |
| 4 |  | d 1  |
| 5 |  | e 12 |
| 6 |  | f 2  |

|    |          |      |
|----|----------|------|
| 7  | <b>g</b> | 2    |
| 8  | <b>h</b> | 18   |
| 9  | <b>i</b> | -3   |
| 10 | <b>j</b> | 14   |
| 11 | <b>k</b> | 33   |
| 12 | <b>l</b> | -112 |
| 13 | <b>m</b> | -22  |
| 14 | <b>n</b> | 20   |



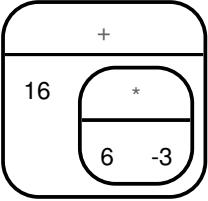
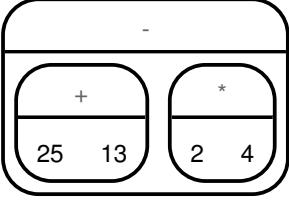
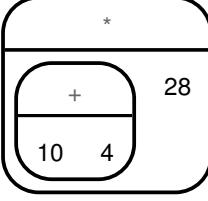
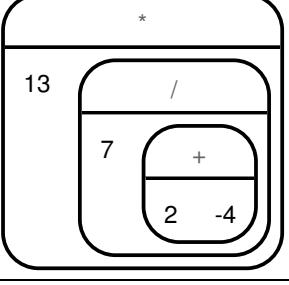
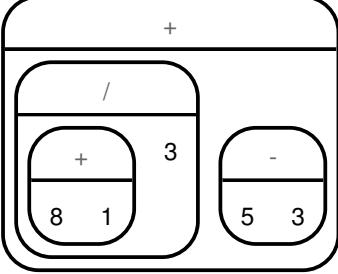
Bootstrap:Algebra by Emmanuel Schanzer, Emma Youndtsmith, Kathi Fisler, and Shriram Krishnamurthi is licensed under a Creative Commons 4.0 Unported License. Based on a work at [www.BootstrapWorld.org](http://www.BootstrapWorld.org). Permissions beyond the scope of this license may be available by contacting [schanzer@BootstrapWorld.org](mailto:schanzer@BootstrapWorld.org).

Name:

Date:

# Exercise: Completing Partial Code from Circles of Evaluation

**Directions:** Each Circle of Evaluation on the left has been partially-converted to code on the right. Finish the code by filling in the gaps, so that it matches the Circle.

|   |   |
|---|---|
| <b>1</b><br>   | <b>a</b><br>$(+ 16 (* 6 -3))$           |
| <b>2</b><br>   | <b>b</b><br>$(- (+ 25 13) (* 2 4))$     |
| <b>3</b><br>  | <b>c</b><br>$(* (+ 10 4) 28)$           |
| <b>4</b><br> | <b>d</b><br>$(* 13 (/ 7 (+ 2 -4)))$     |
| <b>5</b><br> | <b>e</b><br>$(+ (/ (+ 8 1) 3) (- 5 3))$ |



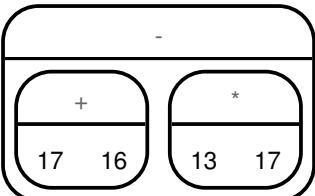
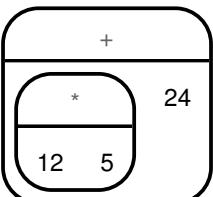
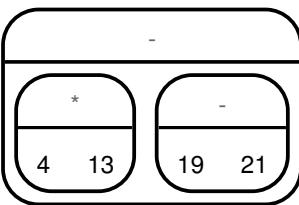
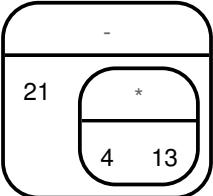
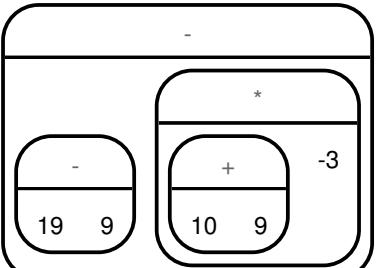
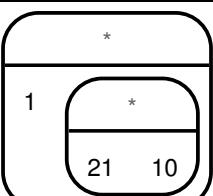
Bootstrap:Algebra by Emmanuel Schanzer, Emma Youndtsmith, Kathi Fisler, and Shriram Krishnamurthi is licensed under a Creative Commons 4.0 Unported License. Based on a work at [www.BootstrapWorld.org](http://www.BootstrapWorld.org). Permissions beyond the scope of this license may be available by contacting [schanzer@BootstrapWorld.org](mailto:schanzer@BootstrapWorld.org).

Name:

Date:

# Exercise: Converting Circles of Evaluation to Code

**Directions:** For each Circle of Evaluation on the left-hand side, write the code for the Circle on the right-hand side:

|   |  |
|---|--|
| <b>1</b><br>   | <b>a</b><br>$(- (+ 17 16) (* 13 17))$      |
| <b>2</b><br>   | <b>b</b><br>$(+ (* 12 5) 24)$              |
| <b>3</b><br>  | <b>c</b><br>$(- (* 4 13) (- 19 21))$       |
| <b>4</b><br> | <b>d</b><br>$(- 21 (* 4 13))$              |
| <b>5</b><br> | <b>e</b><br>$(- (- 19 9) (* (+ 10 9) -3))$ |
| <b>6</b><br> | <b>f</b><br>$(* 1 (* 21 10))$              |



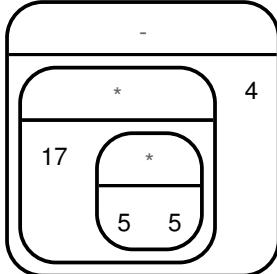
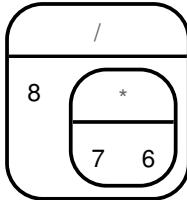
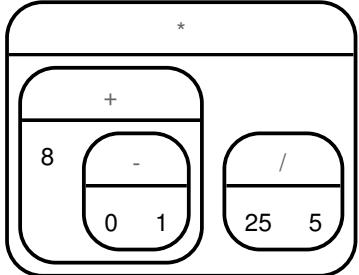
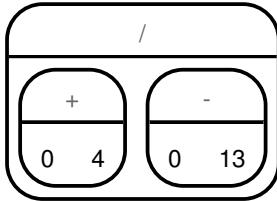
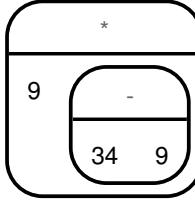
Bootstrap:Algebra by Emmanuel Schanzer, Emma Youndtsmith, Kathi Fisler, and Shriram Krishnamurthi is licensed under a Creative Commons 4.0 Unported License. Based on a work at [www.BootstrapWorld.org](http://www.BootstrapWorld.org). Permissions beyond the scope of this license may be available by contacting [schanzer@BootstrapWorld.org](mailto:schanzer@BootstrapWorld.org).

Name:

Date:

# Exercise: Converting Circles of Evaluation to Code

**Directions:** For each Circle of Evaluation on the left-hand side, write the code for the Circle on the right-hand side:

|   |  |
|---|--|
| <b>1</b><br>   | <b>a</b><br>$(- (* 17 (* 5 5)) 4)$       |
| <b>2</b><br>   | <b>b</b><br>$(/ 8 (* 7 6))$              |
| <b>3</b><br>  | <b>c</b><br>$(* (+ 8 (- 0 1)) (/ 25 5))$ |
| <b>4</b><br> | <b>d</b><br>$(/ (+ 0 4) (- 0 13))$       |
| <b>5</b><br> | <b>e</b><br>$(* 9 (- 34 9))$             |



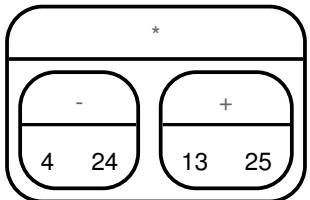
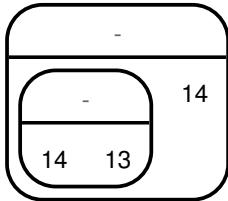
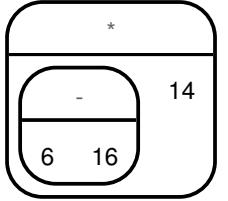
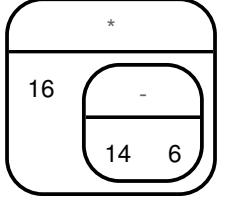
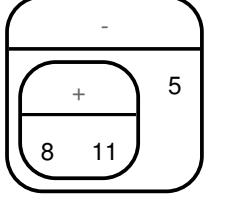
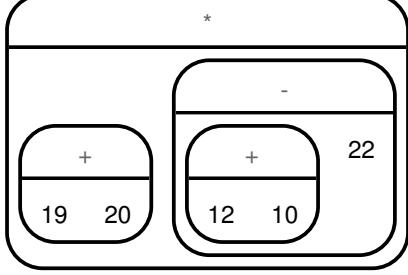
Bootstrap:Algebra by Emmanuel Schanzer, Emma Youndtsmith, Kathi Fisler, and Shriram Krishnamurthi is licensed under a Creative Commons 4.0 Unported License. Based on a work at [www.BootstrapWorld.org](http://www.BootstrapWorld.org). Permissions beyond the scope of this license may be available by contacting [schanzer@BootstrapWorld.org](mailto:schanzer@BootstrapWorld.org).

Name:

Date:

# Exercise: Matching Circles of Evaluation and Code

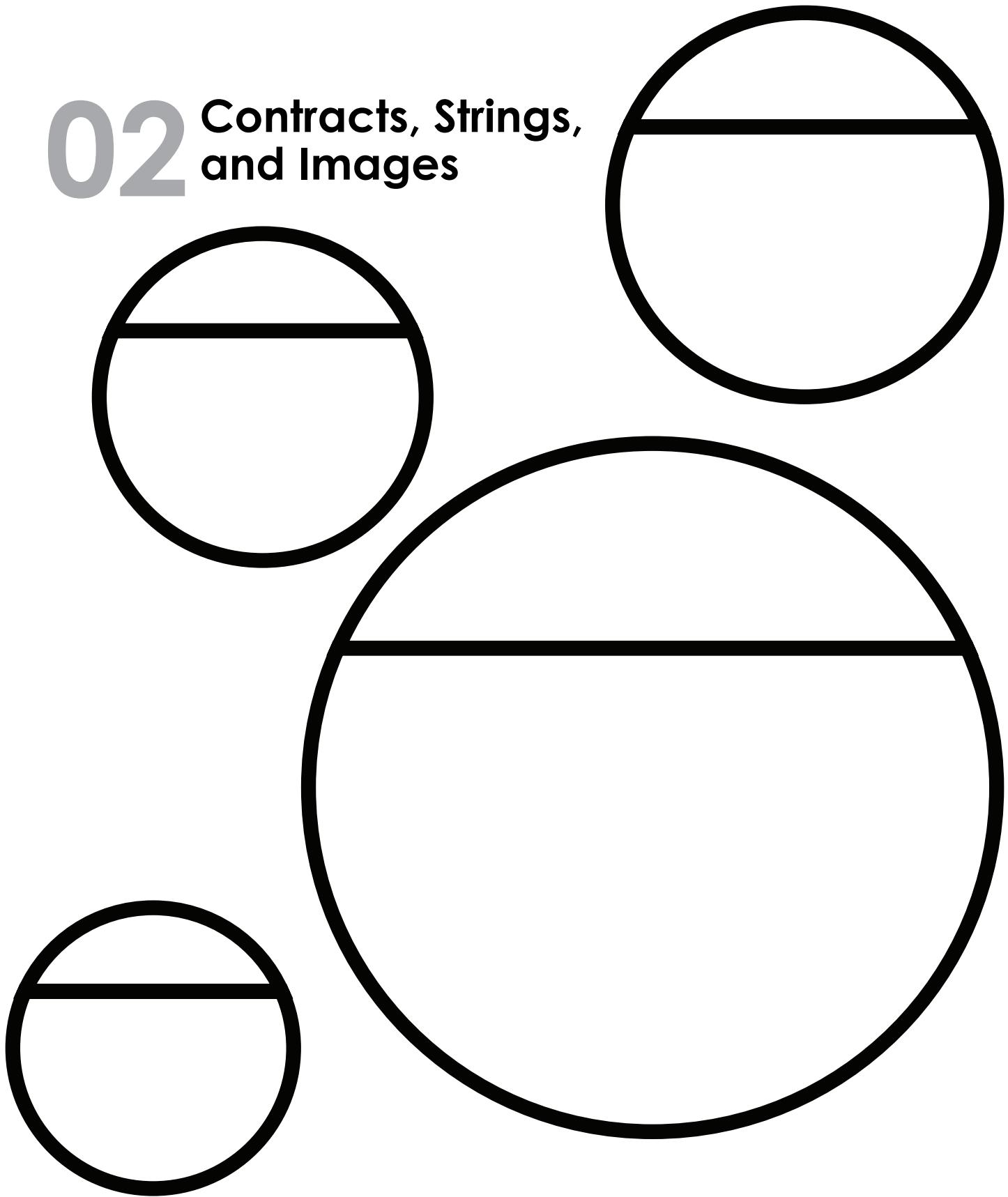
**Directions:** Draw a line from each Circle of Evaluation on the left to the corresponding code on the right.

|   |   |   |  |
|---|---|---|--|
| 1 |    | e | $(* \ (- \ 4 \ 24) \ (+ \ 13 \ 25))$             |
| 2 |    | b | $(- \ (- \ 14 \ 13) \ 14)$                       |
| 3 |   | d | $(* \ (- \ 6 \ 16) \ 14)$                        |
| 4 |  | c | $(* \ 16 \ (- \ 14 \ 6))$                        |
| 5 |  | a | $(- \ (+ \ 8 \ 11) \ 5)$                         |
| 6 |  | f | $(* \ (+ \ 19 \ 20) \ (- \ (+ \ 12 \ 10) \ 22))$ |



Bootstrap:Algebra by Emmanuel Schanzer, Emma Youndtsmith, Kathi Fisler, and Shriram Krishnamurthi is licensed under a Creative Commons 4.0 Unported License. Based on a work at [www.BootstrapWorld.org](http://www.BootstrapWorld.org). Permissions beyond the scope of this license may be available by contacting [schanzer@BootstrapWorld.org](mailto:schanzer@BootstrapWorld.org).

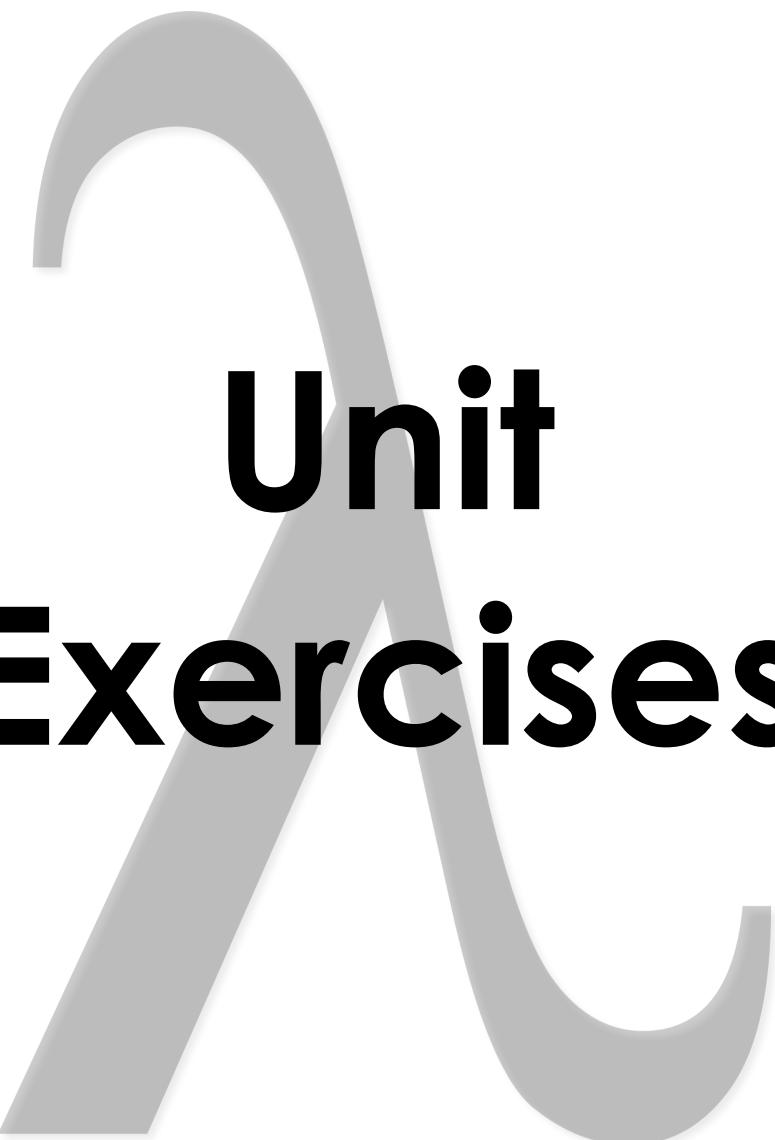
## 02 Contracts, Strings, and Images



## Circles Competition

Time: 5 minutes

|             | Math                | Round 1 -Circle of Evaluation | Round 2 - Racket Code |
|-------------|---------------------|-------------------------------|-----------------------|
| Challenge A | $(3 * 7) - (1 + 2)$ |                               |                       |
| Challenge B | $3 - (1 + 2)$       |                               |                       |
| Challenge C | $3 - (1 + (5 * 6))$ |                               |                       |
| Challenge D | $(1 + (5 * 6)) - 3$ |                               |                       |



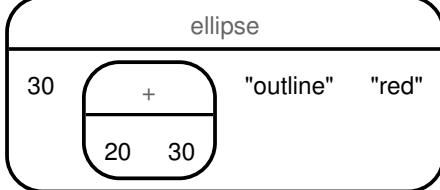
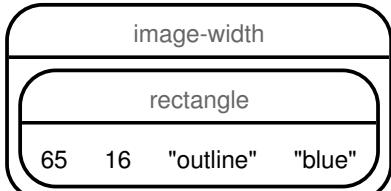
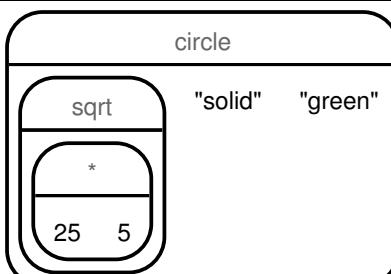
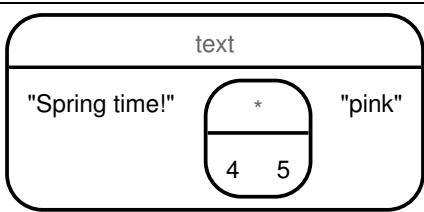
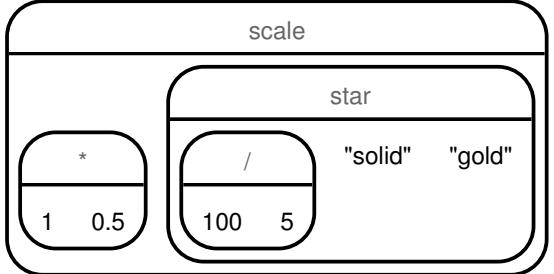
# **Unit Exercises**

Name:

Date:

# Exercise: Converting Circles of Evaluation to Code

**Directions:** For each Circle of Evaluation on the left, write down the code on the right:

|   |   |   |  |
|---|---|---|--|
| 1 |   | a | (ellipse 30 (+ 20 30) "outline" "red")               |
| 2 |  | b | (image-width (rectangle 65 16 "outline" "blue"))     |
| 3 |    | c | (circle (sqrt (* 25 5)) "solid" "green")             |
| 4 |                    | d | (text "Spring time!" (* 4 5) "pink")                 |
| 5 |    | e | (scale (* 1 0.5)<br>(star (/ 100 5) "solid" "gold")) |



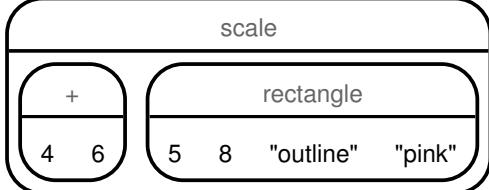
Bootstrap:Algebra by Emmanuel Schanzer, Emma Youndtsmith, Kathi Fisler, and Shriram Krishnamurthi is licensed under a Creative Commons 4.0 Unported License. Based on a work at [www.BootstrapWorld.org](http://www.BootstrapWorld.org). Permissions beyond the scope of this license may be available by contacting [schanzer@BootstrapWorld.org](mailto:schanzer@BootstrapWorld.org).

Name:

Date:

# Exercise: Converting Circles of Evaluation to Code

**Directions:** For each Circle of Evaluation on the left, write down the code on the right:

|   |   |   |
|---|---|---|
| 1 |   | a<br>(rotate 20<br>(triangle (- 45 6) "solid" "gold"))            |
| 2 |  | b<br>(scale (+ 4 6)<br>(rectangle 5 8 "outline" "pink"))          |
| 3 |   | c<br>(rectangle (/ 20 2) 30 "solid" "black") )                    |
| 4 |   | d<br>(rotate 255<br>(scale 3<br>(ellipse 30 10 "solid" "gray")) ) |
| 5 |   | e<br>(+ 15<br>(image-height<br>(circle 16 "outline" "teal")) )    |



Bootstrap:Algebra by Emmanuel Schanzer, Emma Youndtsmith, Kathi Fisler, and Shriram Krishnamurthi is licensed under a Creative Commons 4.0 Unported License. Based on a work at [www.BootstrapWorld.org](http://www.BootstrapWorld.org). Permissions beyond the scope of this license may be available by contacting [schanzer@BootstrapWorld.org](mailto:schanzer@BootstrapWorld.org).

Name:

Date:

# Exercise: Identifying Parts of Expressions

**Directions:** Answer each of the following questions in the space provided:

|   |   |   |   |
|---|---|---|---|
| 1 | How many arguments does the circle function take in (circle (* 3 10) "outline" "black") ?         | a | 3   |
| 2 | How many arguments does the string-append function take in (string-append "Happy" "Halloween") ?  | b | 2   |
| 3 | What is the name of the function being used in (* 4 5) ?  | c | *   |
| 4 | What is the name of the function being used in (string-length "Math is fun!") ?                   | d | string-length   |
| 5 | What is the name of the outermost function being used in (rotate 45 (star 15 "solid" "orange")) ? | e | rotate (star is also a function, but not the one in the overall expression) |
| 6 | Is "outline" the name of a function or an argument in (triangle 48 "outline" "pink") ?            | f | an argument   |
| 7 | How many arguments does rotate expect in (rotate 45 (star 15 "solid" "orange")) ?                 | g | 2   |
| 8 | What is the first argument to the rotate function in (rotate 45 (star 15 "solid" "orange")) ?     | h | 45  |
| 9 | What is the third argument to the star function in (rotate 45 (star 15 "solid" "orange")) ?       | i | "orange"  |



Bootstrap:Algebra by Emmanuel Schanzer, Emma Youndtsmith, Kathi Fisler, and Shriram Krishnamurthi is licensed under a Creative Commons 4.0 Unported License. Based on a work at [www.BootstrapWorld.org](http://www.BootstrapWorld.org). Permissions beyond the scope of this license may be available by contacting [schanzer@BootstrapWorld.org](mailto:schanzer@BootstrapWorld.org).

Name:

Date:

# Exercise: Identifying Parts of Expressions

**Directions:** Answer each of the following questions in the space provided:

|   |   |   |  |
|---|---|---|--|
| 1 | What is the name of the innermost function being used in (+ (string-length "broccoli") 8) ?       | a | string-length  |
| 2 | What is the domain of the outermost function being used in (scale 2 (circle 40 "solid" "blue")) ? | b | number image   |
| 3 | What is the domain of the innermost function being used in (scale 2 (circle 40 "solid" "blue")) ? | c | number string string   |
| 4 | How many arguments does the + function take in (+ (string-length "broccoli") 8) ?                 | d | 2  |
| 5 | What is the range of the function string-length?  | e | number   |
| 6 | Is text a string, a function, or an image?  | f | function   |
| 7 | Is the range of text a string or an image?  | g | image  |
| 8 | What is the first argument to the circle function in (scale 2 (circle 40 "solid" "blue")) ?       | h | 40   |
| 9 | Do the rectangle and the square function have the same domain?                                    | i | No. The domain of rectangle is number number string string and the domain of square is number string string. |



Bootstrap:Algebra by Emmanuel Schanzer, Emma Youndtsmith, Kathi Fisler, and Shriram Krishnamurthi is licensed under a Creative Commons 4.0 Unported License. Based on a work at [www.BootstrapWorld.org](http://www.BootstrapWorld.org). Permissions beyond the scope of this license may be available by contacting [schanzer@BootstrapWorld.org](mailto:schanzer@BootstrapWorld.org).

Name:

Date:

# Exercise: Matching Expressions and Contracts

**Directions:** Can you guess the contract for a function just by seeing how it's used? Try to match each Expression in the left column with a Contract in the right column that describes the function being used. Some expressions or contracts might not have a match.

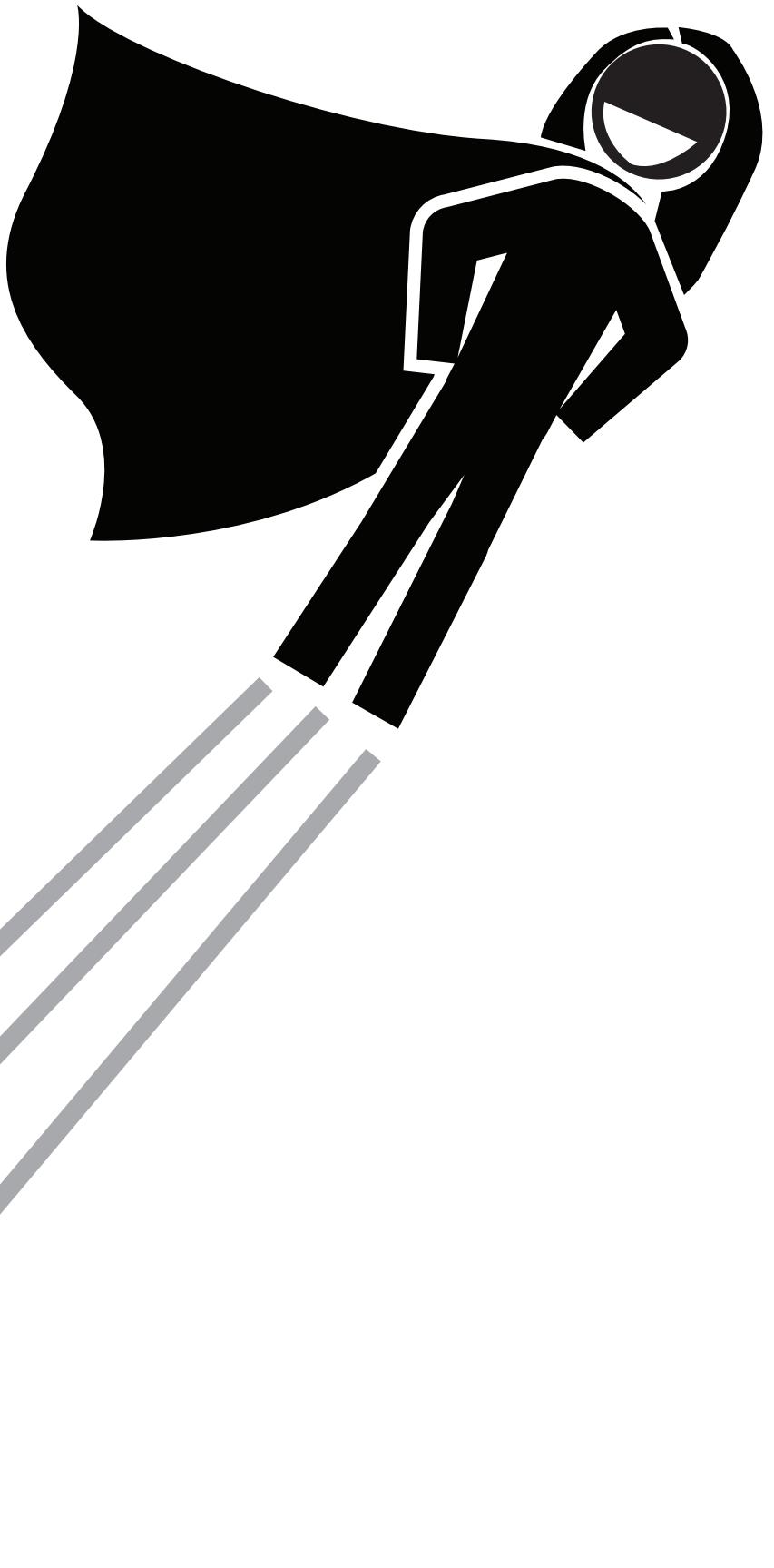
|   |   |   |
|---|---|---|
| 1 | (big-star "purple")                     | No matching answer  |
| 2 | (big-star 25 "gold")                    | a<br>; big-star : number string -> image                      |
| 3 | (message-cost 15)                       | c<br>; message-cost : number -> number                        |
| 4 | (message-cost "where are you?")         | b<br>; message-cost : string -> number                        |
| 5 | (show-score "Red Sox" 6 "Cardinals" 4)  | f<br>; show-score : string number string number<br>; -> image |
| 6 | (show-score "Red Sox" "Cardinals" 5 5)  | No matching answer  |
| 7 | (find-winner "Red Sox" 4 "Cardinals" 6) | No matching answer  |
| 8 | (pick-flavor "vanilla" "oreo" "mint")   | i<br>; pick-flavor : string string string -> image            |
| 9 | (prefer-flavor "lemon" "lime")          | g<br>; prefer-flavor : string string -> number                |



Bootstrap:Algebra by Emmanuel Schanzer, Emma Youndtsmith, Kathi Fisler, and Shriram Krishnamurthi is licensed under a Creative Commons 4.0 Unported License. Based on a work at [www.BootstrapWorld.org](http://www.BootstrapWorld.org). Permissions beyond the scope of this license may be available by contacting [schanzer@BootstrapWorld.org](mailto:schanzer@BootstrapWorld.org).

# 03

## Intro to Definitions



## Fast Functions

; \_\_\_\_\_ : \_\_\_\_\_ -> \_\_\_\_\_  
name domain range

(EXAMPLE (\_\_\_\_\_ \_\_\_\_\_) \_\_\_\_\_)

(EXAMPLE (\_\_\_\_\_ \_\_\_\_\_) \_\_\_\_\_)

(define (\_\_\_\_\_ \_\_\_\_\_) \_\_\_\_\_)

; \_\_\_\_\_ : \_\_\_\_\_ -> \_\_\_\_\_  
name domain range

(EXAMPLE (\_\_\_\_\_ \_\_\_\_\_) \_\_\_\_\_)

(EXAMPLE (\_\_\_\_\_ \_\_\_\_\_) \_\_\_\_\_)

(define (\_\_\_\_\_ \_\_\_\_\_) \_\_\_\_\_)

; \_\_\_\_\_ : \_\_\_\_\_ -> \_\_\_\_\_  
name domain range

(EXAMPLE (\_\_\_\_\_ \_\_\_\_\_) \_\_\_\_\_)

(EXAMPLE (\_\_\_\_\_ \_\_\_\_\_) \_\_\_\_\_)

(define (\_\_\_\_\_ \_\_\_\_\_) \_\_\_\_\_)

; \_\_\_\_\_ : \_\_\_\_\_ -> \_\_\_\_\_  
name domain range

(EXAMPLE (\_\_\_\_\_ \_\_\_\_\_) \_\_\_\_\_)

(EXAMPLE (\_\_\_\_\_ \_\_\_\_\_) \_\_\_\_\_)

(define (\_\_\_\_\_ \_\_\_\_\_) \_\_\_\_\_)

## Fast Functions

; \_\_\_\_\_ : \_\_\_\_\_ -> \_\_\_\_\_  
name domain range

(EXAMPLE (\_\_\_\_\_ \_\_\_\_\_) \_\_\_\_\_)

(EXAMPLE (\_\_\_\_\_ \_\_\_\_\_) \_\_\_\_\_)

(define (\_\_\_\_\_ \_\_\_\_\_) \_\_\_\_\_)

; \_\_\_\_\_ : \_\_\_\_\_ -> \_\_\_\_\_  
name domain range

(EXAMPLE (\_\_\_\_\_ \_\_\_\_\_) \_\_\_\_\_)

(EXAMPLE (\_\_\_\_\_ \_\_\_\_\_) \_\_\_\_\_)

(define (\_\_\_\_\_ \_\_\_\_\_) \_\_\_\_\_)

; \_\_\_\_\_ : \_\_\_\_\_ -> \_\_\_\_\_  
name domain range

(EXAMPLE (\_\_\_\_\_ \_\_\_\_\_) \_\_\_\_\_)

(EXAMPLE (\_\_\_\_\_ \_\_\_\_\_) \_\_\_\_\_)

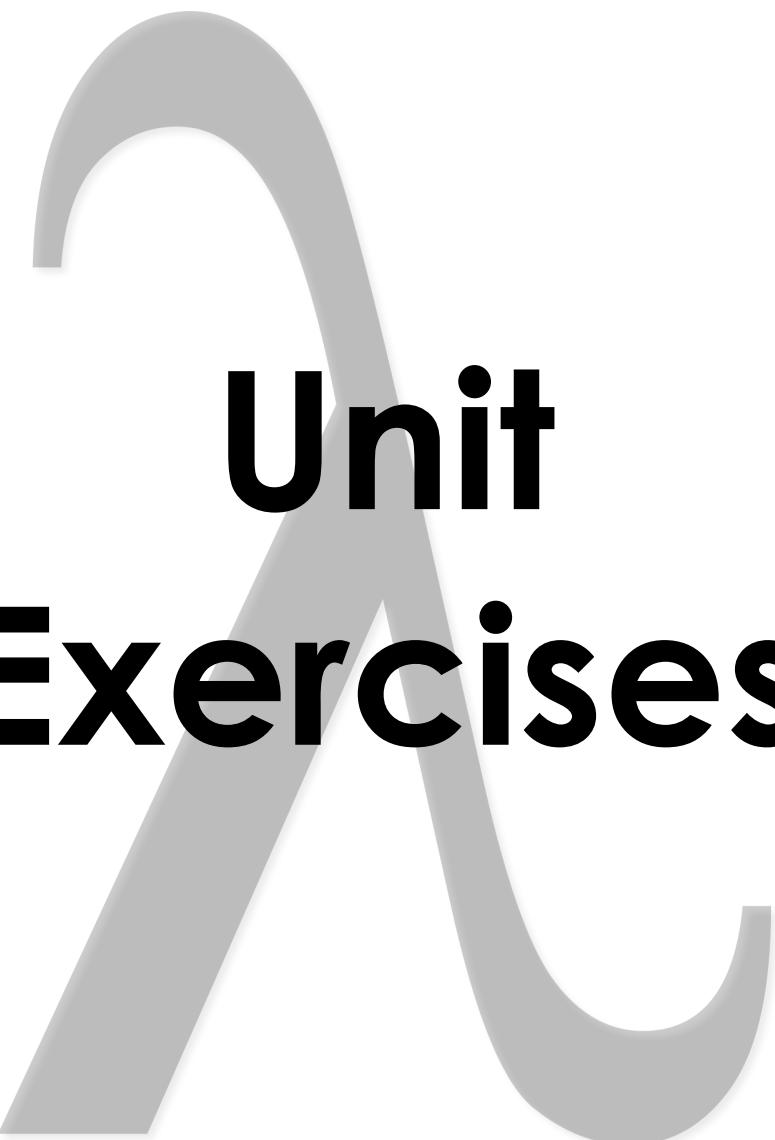
(define (\_\_\_\_\_ \_\_\_\_\_) \_\_\_\_\_)

; \_\_\_\_\_ : \_\_\_\_\_ -> \_\_\_\_\_  
name domain range

(EXAMPLE (\_\_\_\_\_ \_\_\_\_\_) \_\_\_\_\_)

(EXAMPLE (\_\_\_\_\_ \_\_\_\_\_) \_\_\_\_\_)

(define (\_\_\_\_\_ \_\_\_\_\_) \_\_\_\_\_)



# **Unit Exercises**

Name:

Date:

# Exercise: Matching Examples and Function Definitions

**Directions:** Match the EXAMPLEs on the left with the definitions on the right. NOTE: Some definitions may not match any examples!

|   |   |   |  |
|---|---|---|--|
| 1 | (EXAMPLE (mystery "red")<br>(circle 10 "solid" "red"))<br>(EXAMPLE (mystery "orange")<br>(circle 10 "solid" "orange"))  | f | (define (mystery c)<br>(circle 10 "solid" c))            |
| 2 | (EXAMPLE (mystery 30)<br>(circle 30 "outline" "red"))<br>(EXAMPLE (mystery 10)<br>(circle 10 "outline" "red"))  | a | (define (mystery r)<br>(circle r "outline" "red"))       |
| 3 | (EXAMPLE (mystery 30)<br>(rectangle 30 10 "outline" "red"))<br>(EXAMPLE (mystery 10)<br>(rectangle 10 10 "outline" "red"))  | c | (define (mystery w)<br>(rectangle w 10 "outline" "red")) |
| 4 | (EXAMPLE (mystery 30)<br>(rectangle 20 30 "solid" "red"))<br>(EXAMPLE (mystery 20)<br>(rectangle 20 20 "solid" "red"))<br>(EXAMPLE (mystery 10)<br>(rectangle 20 10 "solid" "red")) | g | (define (mystery h)<br>(rectangle 20 h "solid" "red"))   |
| 5 | (EXAMPLE (mystery 2)<br>(+ 2 2))<br>(EXAMPLE (mystery 4)<br>(+ 4 4))<br>(EXAMPLE (mystery 5)<br>(+ 5 5))  | d | (define (mystery num)<br>(+ num num))                    |



Bootstrap:Algebra by Emmanuel Schanzer, Emma Youndtsmith, Kathi Fisler, and Shriram Krishnamurthi is licensed under a Creative Commons 4.0 Unported License. Based on a work at [www.BootstrapWorld.org](http://www.BootstrapWorld.org). Permissions beyond the scope of this license may be available by contacting [schanzer@BootstrapWorld.org](mailto:schanzer@BootstrapWorld.org).

# Exercise: Creating Contracts From Examples

**Directions:** For each of the following collections of Examples, write the contract for the function above the examples.

1. ; big-triangle : number string -> image

```
(EXAMPLE (big-triangle 100 "red")
         (triangle 100 "solid" "red"))
(EXAMPLE (big-triangle 100 "orange")
         (triangle 100 "solid" "orange"))
```

2. ; purple-square : number -> image

```
(EXAMPLE (purple-square 15)
         (rectangle 15 15 "outline" "purple"))
(EXAMPLE (purple-square 6)
         (rectangle 6 6 "outline" "purple"))
```

3. ; banner : string -> image

```
(EXAMPLE (banner "Game Today!")
         (text "Game Today!" 50 "red"))
(EXAMPLE (banner "Go Team!")
         (text "Go Team!" 50 "red"))
(EXAMPLE (banner "Exit")
         (text "Exit" 50 "red"))
```

4. ; twinkle : number string -> image

```
(EXAMPLE (twinkle 5 "outline")
         (star 5 "outline" "yellow"))
(EXAMPLE (twinkle 15 "solid")
         (star 15 "solid" "yellow"))
(EXAMPLE (twinkle 36 "outline")
         (star 36 "outline" "yellow"))
```

5. ; half : number -> number

```
(EXAMPLE (half 2)
        (/ 2 2))
(EXAMPLE (half 4)
        (/ 4 2))
(EXAMPLE (half 9)
        (/ 9 2)))
```

6. ; how-long? : string -> number

```
(EXAMPLE (how-long? "Soccer") 6)
(EXAMPLE (how-long? "Cupcakes") 8)
(EXAMPLE (how-long? "Hi Mom") 6)
(EXAMPLE (how-long? "Cheese pizza") 12)
```



Bootstrap:Algebra by Emmanuel Schanzer, Emma Youndtsmith, Kathi Fisler, and Shriram Krishnamurthi is licensed under a Creative Commons 4.0 Unported License. Based on a work at [www.BootstrapWorld.org](http://www.BootstrapWorld.org). Permissions beyond the scope of this license may be available by contacting [schanzer@BootstrapWorld.org](mailto:schanzer@BootstrapWorld.org).

Name:

Date:

# Exercise: Creating Contracts From Examples

**Directions:** For each of the following collections of Examples, write the contract for the function above the examples.

1. ; third-root : number -> number

```
(EXAMPLE (third-root 16)
         (expt 16 1/3))
(EXAMPLE (third-root 125)
         (expt 125 1/3))
(EXAMPLE (third-root 625)
         (expt 625 1/3))
```

2. ; sentence-length : string number -> number

```
(EXAMPLE (sentence-length "hello" 5)
         (+ (string-length "hello") 5))
(EXAMPLE (sentence-length "please" 6)
         (+ (string-length "please") 6))
(EXAMPLE (sentence-length "goodbye" 10)
         (+ (string-length "please") 10))
```

3. ; upside-down : image -> image

```
(EXAMPLE (upside-down (triangle 50 "solid" "yellow"))
         (rotate 180 (triangle 50 "solid" "yellow")))
(EXAMPLE (upside-down (star 150 "outline" "blue"))
         (rotate 180 (star 150 "outline" "blue")))
```

4. ; product-squared : number number -> number

```
(EXAMPLE (product-squared 5 6)
         (sqr (* 5 6)))
(EXAMPLE (product-squared 10 17)
         (sqr (* 10 17)))
(EXAMPLE (product-squared 2 5)
         (sqr (* 2 5))))
```

5. ; half-image : string -> image

```
(EXAMPLE (half-image "www.images.com/cat_image")
         (scale .5 (bitmap/url "www.images.com/cat_image")))
(EXAMPLE (half-image "www.images.com/dog_image")
         (scale .5 (bitmap/url "www.images.com/dog_image")))
```

6. ; triple : number -> number

```
(EXAMPLE (triple 4)
         (* 4 3))
(EXAMPLE (triple 10)
         (* 10 3))
(EXAMPLE (triple 3)
         (* 3 3)))
```



Bootstrap:Algebra by Emmanuel Schanzer, Emma Youndtsmith, Kathi Fisler, and Shriram Krishnamurthi is licensed under a Creative Commons 4.0 Unported License. Based on a work at [www.BootstrapWorld.org](http://www.BootstrapWorld.org). Permissions beyond the scope of this license may be available by contacting [schanzer@BootstrapWorld.org](mailto:schanzer@BootstrapWorld.org).

# 04 Design Recipe

1 Contract

2 Example

3 Definition



## Word Problem: rocket-height

**Directions:** A rocket blasts off, traveling at 7 meters per second. Write a function called 'rocket-height' that takes in the number of seconds that have passed since the rocket took off, and which produces the height of the rocket at that time.

# **Contract and Purpose Statement**

## *Every contract has three parts...*

|                 |   |        |   |        |
|-----------------|---|--------|---|--------|
| ; rocket-height | : | number | → | number |
| function name   |   | domain |   | range  |

; Produce height of rocket after given number of seconds

*what does the function do?*

## Examples

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

|           |                      |                 |   |                                   |   |
|-----------|----------------------|-----------------|---|-----------------------------------|---|
| (EXAMPLE( | <u>rocket-height</u> | 0               | ) | (* 0 7)                           | ) |
|           | <i>function name</i> | <i>input(s)</i> |   | <i>what the function produces</i> |   |
| (EXAMPLE( | <u>rocket-height</u> | 10              | ) | (* 10 7)                          | ) |
|           | <i>function name</i> | <i>input(s)</i> |   | <i>what the function produces</i> |   |

## Definition

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

## Word Problem: lawn-area

**Directions:** Use the Design Recipe to write a function 'lawn-area', which takes in the width and length of a lawn, and returns the area of the lawn. (Don't forget: area = length \* width!)

### Contract and Purpose Statement

Every contract has three parts...

|   |  |   |        |        |   |        |
|---|--|---|--------|--------|---|--------|
| ; | <u>lawn-area</u>                                 | : | number | number | → | number |
|   |  |   |        | domain |   | range  |
| ; | Produce area of lawn with given length and width |   |        |        |   |        |

*what does the function do?*

### Examples

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

|           |                  |          |   |                            |   |
|-----------|------------------|----------|---|----------------------------|---|
| (EXAMPLE( | <u>lawn-area</u> | 10 20    | ) | (* 10 20)                  | ) |
|           |                  | input(s) |   | what the function produces |   |
| (EXAMPLE( | <u>lawn-area</u> | 50 5     | ) | (* 50 5)                   | ) |

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

### Definition

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

|                         |                  |              |   |
|-------------------------|------------------|--------------|---|
| (define(                | <u>lawn-area</u> | length width | ) |
|                         | function name    | variables    |   |
| <u>(* length width)</u> |                  |              | ) |

*what the function does with those variables*

# Word Problem: red-square

**Directions:** Use the Design Recipe to write a function 'red-square', which takes in a number (the length of each side of the square) and outputs a solid red rectangle whose length and width are the same size.

## Contract and Purpose Statement

Every contract has three parts...

|   |   |   |        |   |       |
|---|---|---|--------|---|-------|
| ; | red-square  | : | number | → | image |
|   |   |   | domain |   | range |
| ; | Produce rectangle with given number as width and height |   |        |   |       |
|   | what does the function do?                              |   |        |   |       |

## Examples

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

|           |               |          |   |                                 |   |
|-----------|---------------|----------|---|---------------------------------|---|
| (EXAMPLE( | red-square    | 4        | ) | (rectangle 4 4 "solid" "red")   | ) |
|           | function name | input(s) |   | what the function produces      |   |
| (EXAMPLE( | red-square    | 12       | ) | (rectangle 12 12 "solid" "red") | ) |
|           | function name | input(s) |   | what the function produces      |   |

## Definition

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

|   |               |           |   |
|---|---------------|-----------|---|
| (define(                                    | red-square    | size      | ) |
|   | function name | variables |   |
| (rectangle size size "solid" "red")         |               |           | ) |
| what the function does with those variables |               |           |   |



# **Unit Exercises**

Name: \_\_\_\_\_

Date: \_\_\_\_\_

# Exercise: Do Examples Have the Same Contracts?

**Directions:** For each pairs of Examples below, decide whether the two examples have the same contract. If they do, fill in the contract in the space provided. If not, write a few words explaining why.

|   |   |
|---|---|
| <b>1</b><br><pre>(EXAMPLE (mystery 30           (* 30 50)) (EXAMPLE (mystery 10           (text "Welcome!" 10 "darkgreen")))</pre>  | <b>a</b> no match (the range types are different)                   |
| <b>2</b><br><pre>(EXAMPLE (mystery 30 40           (- 40 (* 2 30))) (EXAMPLE (mystery 10 15           (- 15 (* 2 10))))</pre>   | <b>b</b> ; mystery : number number -> number                        |
| <b>3</b><br><pre>(EXAMPLE (mystery "New York"           (text "New York" 20 "red")) (EXAMPLE (mystery 20)           (text "New York" 20 "red")))</pre>  | <b>c</b> no match (the domain types are different)                  |
| <b>4</b><br><pre>(EXAMPLE (mystery "green" 32           (circle 32 "outline" "green")) (EXAMPLE (mystery 18 "green")           (circle 18 "outline" "green")))</pre>  | <b>d</b> no match (the domain types are in different orders)        |
| <b>5</b><br><pre>(EXAMPLE (mystery 6 9 10           (/ 6 (+ 9 10))) (EXAMPLE (mystery 3 7)           (/ 3 (+ 7 10))))</pre>   | <b>e</b> no match (the domains have different numbers of arguments) |
| <b>6</b><br><pre>(EXAMPLE (mystery "red" "blue"           (text "blue" 25 "red")) (EXAMPLE (mystery "purple" "Go Team!")           (text "Go Team!" 25 "purple")))</pre>  | <b>f</b> ; mystery : string string -> image                         |
| <b>7</b><br><pre>(EXAMPLE (mystery (circle 55 "outline" "black") 12)           (scale 12 (circle 55 "outline" "black"))) (EXAMPLE (mystery (rectangle 24 32 "outline" "purple") 5)           (scale 5 (rectangle 24 32 "outline" "purple"))))</pre> | <b>g</b> ; mystery : image number -> image                          |



Bootstrap:Algebra by Emmanuel Schanzer, Emma Youndtsmith, Kathi Fisler, and Shriram Krishnamurthi is licensed under a Creative Commons 4.0 Unported License. Based on a work at [www.BootstrapWorld.org](http://www.BootstrapWorld.org). Permissions beyond the scope of this license may be available by contacting [schanzer@BootstrapWorld.org](mailto:schanzer@BootstrapWorld.org).

Name:

Date:

# Exercise: Do Examples Have the Same Contracts?

**Directions:** For each pairs of Examples below, decide whether the two examples have the same contract. If they do, fill in the contract in the space provided. If not, write a few words explaining why.

|   |   |
|---|---|
| <b>1</b><br><pre>(EXAMPLE (mystery "+" 4 5) (+ 4 5)) (EXAMPLE (mystery "sqrt" 25) (sqrt 25))</pre>  | <b>a</b> no match (the domains have different numbers of arguments) |
| <b>2</b><br><pre>(EXAMPLE (mystery (triangle 70 "solid" "green")) (triangle 140 "solid" "green")) (EXAMPLE (mystery (circle 100 "solid" "blue")) (circle 200 "solid" "blue"))</pre> | <b>b</b> ; mystery : image -> image                                 |
| <b>3</b><br><pre>(EXAMPLE (mystery "red") (triangle 140 "solid" "red")) (EXAMPLE (mystery "blue" "circle") (circle 140 "solid" "blue"))</pre>                                       | <b>c</b> no match (the domains have different numbers of arguments) |
| <b>4</b><br><pre>(EXAMPLE (mystery "circle" 4) (* pi (sqr 4))) (EXAMPLE (mystery "square" 5) (sqr 5))</pre>   | <b>d</b> ; mystery : string number -> number                        |
| <b>5</b><br><pre>(EXAMPLE (mystery "dog") 3) (EXAMPLE (mystery "cat") "kitten")</pre>   | <b>e</b> no match (the range types are different)                   |
| <b>6</b><br><pre>(EXAMPLE (mystery "dog") 3) (EXAMPLE (mystery "kitten") 6)</pre>   | <b>f</b> ; mystery : string -> number                               |
| <b>7</b><br><pre>(EXAMPLE (mystery 4 5 "big") (scale 2 (rectangle 4 5 "solid" "blue"))) (EXAMPLE (mystery 10 "small" 4) (scale .5 (rectangle 10 4 "solid" "blue")))</pre>           | <b>g</b> no match (the domains have different orders of arguments)  |



Bootstrap:Algebra by Emmanuel Schanzer, Emma Youndtsmith, Kathi Fisler, and Shriram Krishnamurthi is licensed under a Creative Commons 4.0 Unported License. Based on a work at [www.BootstrapWorld.org](http://www.BootstrapWorld.org). Permissions beyond the scope of this license may be available by contacting [schanzer@BootstrapWorld.org](mailto:schanzer@BootstrapWorld.org).

Name:

Date:

# Exercise: Matching Contracts and Examples

**Directions:** Match each Example on the left with the Contract on the right. NOTE: Multiple examples may match to the same contract!

|  |  |
|--|--|
| 1<br>(EXAMPLE (match 30 "red")<br>(+ 30 (string-length "red")))  | b<br>; match : number string -> number |
| 2<br>(EXAMPLE (match 10 "blue")<br>(+ 10 (string-length "blue")))  | b<br>; match : number string -> number |
| 3<br>(EXAMPLE (match (abs -4) "45")<br>4)  | b<br>; match : number string -> number |
| 4<br>(EXAMPLE (match (triangle 20 "solid" "blue")<br>3)<br>(scale 3 (triangle 20 "solid"<br>"blue"))))   | d<br>; match : image number -> image   |
| 5<br>(EXAMPLE (match (circle 10 "solid" "orange")<br>22)<br>(scale 22<br>(circle 10 "solid" "orange")))) | d<br>; match : image number -> image   |
| 6<br>(EXAMPLE (match 5 (star 20 "solid" "red")<br>(rotate (- 90 5)<br>(star 20 "solid" "red"))))         | c<br>; match : number image -> image   |
| 7<br>(EXAMPLE (match (circle 20 "outline" "gold")<br>(rotate 37 (circle 20 "outline"<br>"gold"))))       | a<br>; match : image -> image          |
| 8<br>(EXAMPLE (match (circle 10 "solid" "green")<br>(rotate 37<br>(circle 10 "solid" "green"))))         | a<br>; match : image -> image          |



Bootstrap:Algebra by Emmanuel Schanzer, Emma Youndtsmith, Kathi Fisler, and Shriram Krishnamurthi is licensed under a Creative Commons 4.0 Unported License. Based on a work at [www.BootstrapWorld.org](http://www.BootstrapWorld.org). Permissions beyond the scope of this license may be available by contacting [schanzer@BootstrapWorld.org](mailto:schanzer@BootstrapWorld.org).

Name:

Date:

# Exercise: Matching Contracts and Examples

**Directions:** Match each Example on the left with the Contract on the right. NOTE: Multiple examples may match to the same contract!

|  |  |
|--|--|
| 1<br>(EXAMPLE (match 30)<br>(star (* 30 2) "outline" "purple"))  | d<br>; match : number -> image         |
| 2<br>(EXAMPLE (match 24)<br>(star (* 24 2) "outline" "purple"))  | d<br>; match : number -> image         |
| 3<br>(EXAMPLE (match (triangle 20 "outline" "brown") 3)<br>(* 3<br>(image-height<br>(triangle 20 "outline" "brown")))) | e<br>; match : image number -> number  |
| 4<br>(EXAMPLE (match (circle 10 "solid" "silver")<br>16)<br>(* 16<br>(image-height<br>(circle 10 "solid" "silver"))))  | e<br>; match : image number -> number  |
| 5<br>(EXAMPLE (match "triangle" "blue")<br>(triangle 40 "outline" "blue"))   | c<br>; match : string string -> image  |
| 6<br>(EXAMPLE (match (string-length "coffee")<br>(string-length "tea"))<br>(+ 6 3)))                                   | a<br>; match : number number -> number |
| 7<br>(EXAMPLE (match (string-length "tabletop")<br>"8"))   | b<br>; match : number -> string        |
| 8<br>(EXAMPLE (match 1.50)<br>"greater than 1")  | b<br>; match : number -> string        |



Bootstrap:Algebra by Emmanuel Schanzer, Emma Youndtsmith, Kathi Fisler, and Shriram Krishnamurthi is licensed under a Creative Commons 4.0 Unported License. Based on a work at [www.BootstrapWorld.org](http://www.BootstrapWorld.org). Permissions beyond the scope of this license may be available by contacting [schanzer@BootstrapWorld.org](mailto:schanzer@BootstrapWorld.org).

## Check for Mistakes in this Word Problem: calc-pencils

**Directions:** All students are given five (5) pencils at the beginning of the school year. Write a function called calc-pencils that takes 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, given variable names to all your input values...

(define( calculate-pencils p )  
function name variables  
(\* p 5))  
*what the function does with those variables*

## Check for Mistakes in this Word Problem: check-total

**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) 20))  
function name input(s) what the function produces

(EXAMPLE( (total 56.67) ) (56.67 (+ (0.2 \* 56.67) 56.67))  
function name input(s) what the function produces

### Definition

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

(define( check-total food-total )  
function name variables  
(\* (+ 0.2 food-total) food-total)  
what the function does with those variables)

## Check for Mistakes in this Word Problem: circle-area

**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

; Takes the diameter of a circle and calculates the area by multiplying the square of the radius

*what does the function do?*

### Examples

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

(EXAMPLE( circle-area                         10                     )    (\* (sqr (/ 10 2)) pi) )  
  function name                         input(s)                                     what the function produces

(EXAMPLE( circle-area                         50                     )    (\* (sqr (/ 50 2)) pi) )  
  function name                         input(s)                                     what the function produces

### Definition

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

(define( area                         diameter                     )  
  function name                         variables  
(\* (sqr diameter) pi) )  
*what the function does with those variables*

**target**



**danger**



## 05 Game Animation

# Word Problem: update-danger

**Directions:** Use the Design Recipe to write a function 'update-danger', which takes in the danger's x-coordinate and produces the next x-coordinate, which is 50 pixels to the left.

## Contract and Purpose Statement

Every contract has three parts...

|  |               |   |        |   |        |
|--|---------------|---|--------|---|--------|
| ;  | update-danger | : | number | → | number |
|  |               |   | domain |   | range  |
| ; Produce new coordinate by subtracting 50 from the given number |               |   |        |   |        |
| what does the function do?                                       |               |   |        |   |        |

## Examples

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

|           |               |          |   |    |     |    |                            |
|-----------|---------------|----------|---|----|-----|----|----------------------------|
| (EXAMPLE( | update-danger | 100      | ) | (- | 100 | 50 | )                          |
|           | function name | input(s) |   |    |     |    | what the function produces |
| (EXAMPLE( | update-danger | 35       | ) | (- | 35  | 50 | )                          |
|           | function name | input(s) |   |    |     |    | what the function produces |

## Definition

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

|   |               |           |     |
|---|---------------|-----------|-----|
| (define(                                    | update-danger | x         | )   |
|   | function name | variables |     |
|   | (-            | x         | 50) |
| what the function does with those variables |               |           | )   |

# Word Problem: update-target

**Directions:** Write a function 'update-target', which takes in the target's x-coordinate and produces the next x-coordinate, which is 50 pixels to the right.

## Contract and Purpose Statement

Every contract has three parts...

; update-target : number → number  
function name domain range  
; Produce new coordinate by adding 50 to the given number  
what does the function do?

## Examples

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

(EXAMPLE( update-target 0 ) (+ 0 50) )  
function name input(s) what the function produces  
(EXAMPLE( update-target 20 ) (+ 20 50) )  
function name input(s) what the function produces

## Definition

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

(define( update-target x )  
function name variables  
(+ x 50) )  
what the function does with those variables

## Check for Mistakes in this Word Problem: target-leap

**Directions:** Write a function which takes in 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 |
|---|---|--------|---|--------|
|   |   | domain |   | range  |
| ; Takes the x-coordinate and returns a new one, multiplied by 2 |   |        |   |        |
| what does the function do?                                      |   |        |   |        |

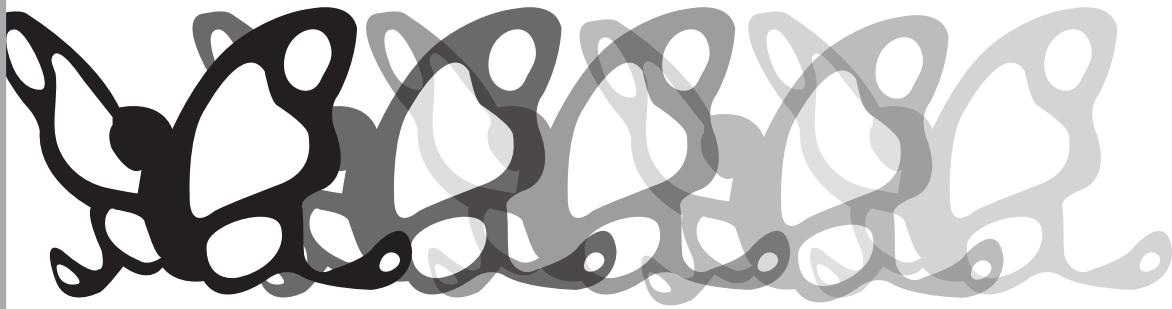
## Examples

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

|           |                      |                 |   |                                   |   |
|-----------|----------------------|-----------------|---|-----------------------------------|---|
| (EXAMPLE( | target-leap          | 100             | ) | (200)                             | ) |
|           | <i>function name</i> | <i>input(s)</i> |   | <i>what the function produces</i> |   |
| (EXAMPLE( | target-leap          | 40              | ) | (200)                             | ) |
|           | <i>function name</i> | <i>input(s)</i> |   | <i>what the function produces</i> |   |

## Definition

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



**“safe-left?”**

## 06 Comparing Functions

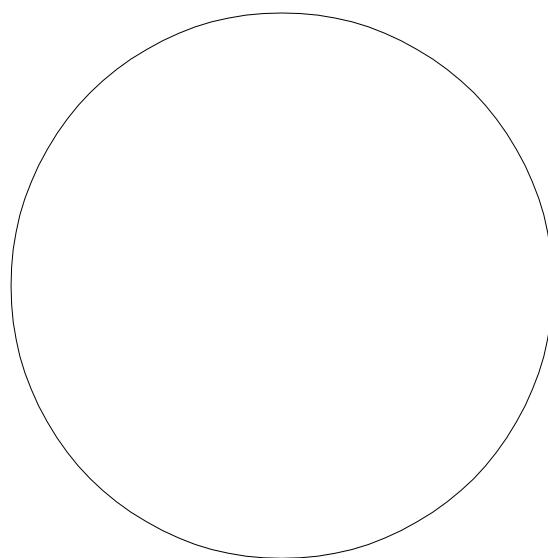
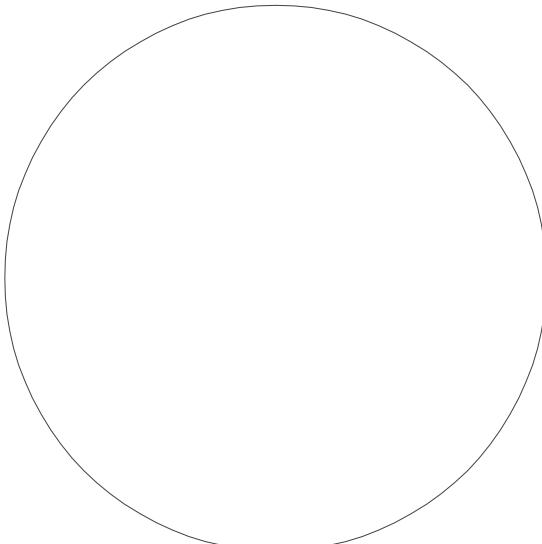
## Sam the Butterfly

Sam is in a  $640 \times 480$  yard. How far he can go to the left and right before he's out of sight?

1. A piece of Sam is still visible on the left as long as...  $\underline{(> \quad x \quad -50)}$

2. A piece of Sam is still visible on the right as long as...  $\underline{\hspace{2cm}}$

3. Draw the Circle of Evaluation for these two expressions in the circles below:



## Word Problem: safe-left?

**Directions:** Use the Design Recipe to write a function 'safe-left?', which takes in an x-coordinate and checks to see if it is greater than -50

### Contract and Purpose Statement

Every contract has three parts...

|  |            |   |        |   |                                   |
|--|------------|---|--------|---|-----------------------------------|
| ;  | safe-left? | : | number | → | boolean                           |
|  |            |   | domain |   | range                             |
| ; Determine whether given number is greater than -50 |            |   |        |   | <i>what does the function do?</i> |

### Examples

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

|           |            |     |                 |             |                                   |
|-----------|------------|-----|-----------------|-------------|-----------------------------------|
| (EXAMPLE( | safe-left? | 25  | )               | (> 25 -50)  | )                                 |
|           |            |     | <i>input(s)</i> |             | <i>what the function produces</i> |
| (EXAMPLE( | safe-left? | -51 | )               | (> -51 -50) | )                                 |
|           |            |     | <i>input(s)</i> |             | <i>what the function produces</i> |

### Definition

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

|             |            |                      |  |
|-------------|------------|----------------------|--|
| (define(    | safe-left? | x                    | )  |
|             |            | <i>function name</i> | <i>variables</i>                                   |
| ( > x -50 ) |            |                      | )  |
|             |            |                      | <i>what the function does with those variables</i> |

## Word Problem: safe-right?

**Directions:** Use the Design Recipe to write a function 'safe-right?', which takes in an x-coordinate and checks to see if it is less than 690.

### Contract and Purpose Statement

Every contract has three parts...

|   |   |        |   |                                   |
|---|---|--------|---|-----------------------------------|
| :      safe-right?                                | : | number | → | boolean                           |
| function name                                     |   | domain |   | range                             |
| ; Determine whether given number is less than 690 |   |        |   | <i>what does the function do?</i> |

### Examples

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

|               |             |          |   |                            |   |
|---------------|-------------|----------|---|----------------------------|---|
| (EXAMPLE(     | safe-right? | 100      | ) | (< 100 690)                | ) |
| function name |             | input(s) |   | what the function produces |   |
| (EXAMPLE(     | safe-right? | 820      | ) | (< 820 690)                | ) |

### Definition

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

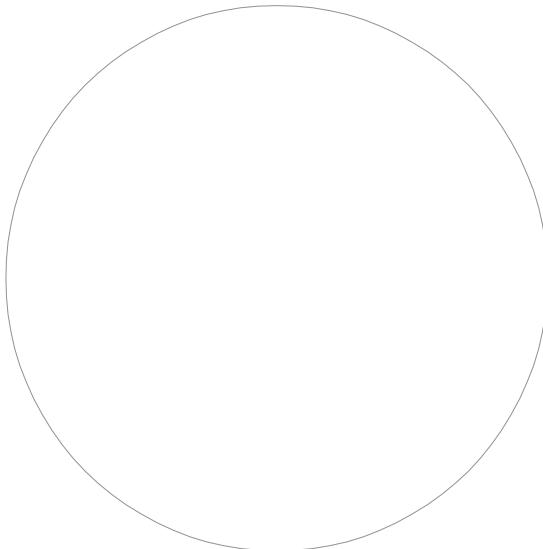
|               |             |           |   |
|---------------|-------------|-----------|---|
| (define(      | safe-right? | x         | ) |
| function name |             | variables |   |
| ( < x 690 )   |             |           | ) |

*what the function does with those variables*

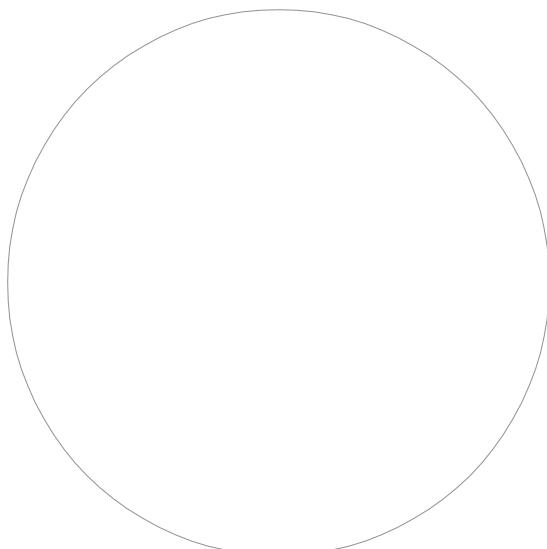
**and / or**

**Write the Circles of Evaluation for these statements, and then convert them to Racket**

1. Two is less than five, and zero is equal to six.



2. Two is less than four or four is equal to six.



## Word Problem: onscreen?

**Directions:** Use the Design Recipe to write a function 'onscreen?', which takes in the x-coordinate and checks to see if Sam is safe on the left AND safe on the right.

### Contract and Purpose Statement

Every contract has three parts...

|   |           |   |        |   |                                   |
|---|-----------|---|--------|---|-----------------------------------|
| ;   | onscreen? | : | number | → | boolean                           |
|   |           |   | domain |   | range                             |
| ; Sam is on the screen if his x-coordinate is safe on the left and safe on the right. |           |   |        |   | <i>what does the function do?</i> |

### Examples

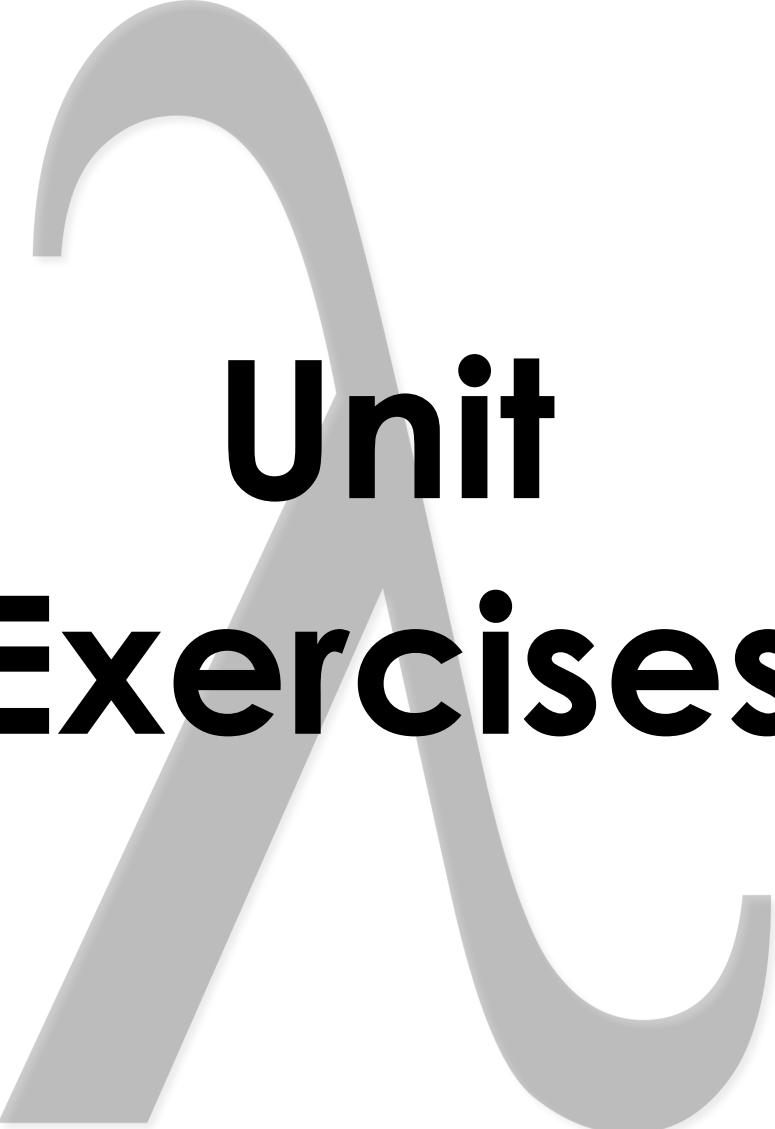
Write some examples, then circle and label what changes...

|           |           |                      |                 |  |   |
|-----------|-----------|----------------------|-----------------|--|---|
| (EXAMPLE( | onscreen? | 10                   | )               | (and (safe-left? 10) (safe-right? 10))   | ) |
|           |           | <i>function name</i> | <i>input(s)</i> | <i>what the function produces</i>        |   |
| (EXAMPLE( | onscreen? | -15                  | )               | (and (safe-left? -15) (safe-right? -15)) | ) |
|           |           | <i>function name</i> | <i>input(s)</i> | <i>what the function produces</i>        |   |

### Definition

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

|  |                      |                  |   |
|--|----------------------|------------------|---|
| (define(   | onscreen?            | x                | ) |
|  | <i>function name</i> | <i>variables</i> |   |
| (and (safe-left? x) (safe-right? x))               |                      |                  | ) |
| <i>what the function does with those variables</i> |                      |                  |   |



# **Unit Exercises**

## Check for Mistakes in this Word Problem: enough-cash?

**Directions:** You go the store with \$1.50 in your pocket. Write a function which 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? "gum" ) (< gum 1.5) )  
function name input(s) what the function produces  
(EXAMPLE( enough-cash? "soda" ) (< gum 150) )  
function name input(s) what the function produces

### Definition

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

(define( enough-cash? item )  
function name variables  
(< item 1.5) )  
what the function does with those variables

## Check for Mistakes in this Word Problem: long-name?

**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) )  
\_\_\_\_\_  
what the function produces

(EXAMPLE( long-name? "Joe Jones" ) (> (string=? "Joe Jones") 10) )  
\_\_\_\_\_  
function name input(s) what the function produces

### Definition

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

(define( long? name )  
\_\_\_\_\_  
function name variables  
(< name 20) )  
\_\_\_\_\_  
what the function does with those variables

## Check for Mistakes in this Word Problem: equal-length?

**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") )  
function name input(s)  
(= (string-length "dog") (string-length "cat"))  
what the function produces

### Definition

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

(define( equal-length? string1 string2 )  
function name variables  
(= string1 string2 )  
what the function does with those variables

## Check for Mistakes in this Word Problem: enough-carpet?

**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 less than 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, given variable names to all your input values...

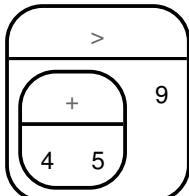
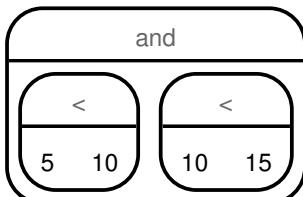
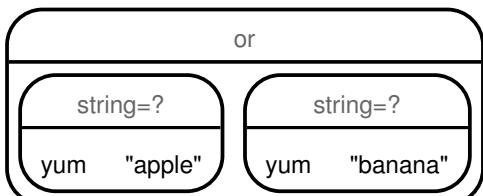
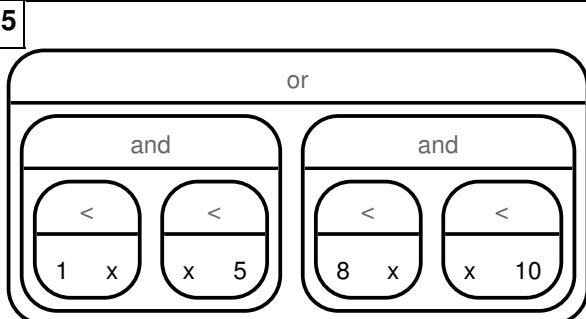
(define( enough-carpet? length width )  
function name variables  
(< (\* length width) 100)  
what the function does with those variables

Name:

Date:

# Exercise: Converting Circles of Evaluation with Booleans to Code

**Directions:** For each Circle of Evaluation in the left column, write the code that corresponds to the Circle in the right column:

|   |   |   |  |
|---|---|---|--|
| 1 |    | a | (> (+ 4 5) 9)  |
| 2 |    | b | (and (< 5 10) (< 10 15))                               |
| 3 |   | c | (or (string=? yum "apple")<br>(string=? yum "banana")) |
| 4 |  | d | (>= (string-length "My Game") 6)                       |
| 5 |  | e | (or (and (< 1 x) (< x 5))<br>(and (< 8 x) (< x 10)))   |



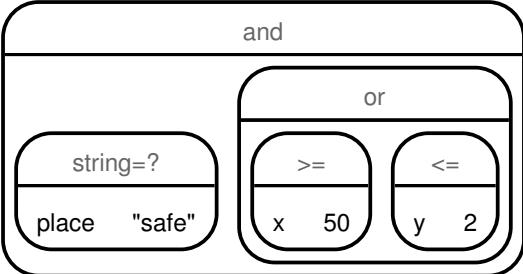
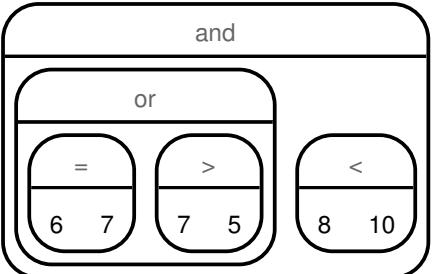
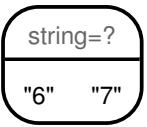
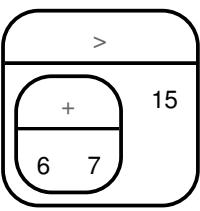
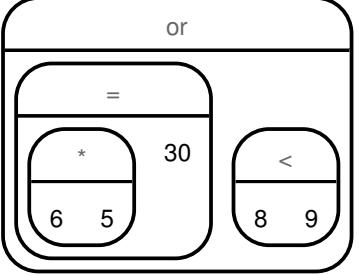
Bootstrap:Algebra by Emmanuel Schanzer, Emma Youndtsmith, Kathi Fisler, and Shriram Krishnamurthi is licensed under a Creative Commons 4.0 Unported License. Based on a work at [www.BootstrapWorld.org](http://www.BootstrapWorld.org). Permissions beyond the scope of this license may be available by contacting [schanzer@BootstrapWorld.org](mailto:schanzer@BootstrapWorld.org).

Name:

Date:

# Exercise: Converting Circles of Evaluation with Booleans to Code

**Directions:** For each Circle of Evaluation in the left column, write the code that corresponds to the Circle in the right column:

|   |  |
|---|--|
| <b>1</b><br>   | <b>a</b><br>$(\text{and} (\text{string=?} \text{ place } \text{"safe"}) (\text{or} (\geq x 50) (\leq y 2)))$ |
| <b>2</b><br>   | <b>b</b><br>$(\text{and} (\text{or} (= 6 7) (> 7 5)) (< 8 10)))$   |
| <b>3</b><br> | <b>c</b><br>$(\text{string=?} \text{ "6" } \text{ "7"})$   |
| <b>4</b><br> | <b>d</b><br>$(> (+ 6 7) 15)$   |
| <b>5</b><br> | <b>e</b><br>$(\text{or} (= (* 6 5) 30) (< 8 9)))$  |



Bootstrap:Algebra by Emmanuel Schanzer, Emma Youndtsmith, Kathi Fisler, and Shriram Krishnamurthi is licensed under a Creative Commons 4.0 Unported License. Based on a work at [www.BootstrapWorld.org](http://www.BootstrapWorld.org). Permissions beyond the scope of this license may be available by contacting [schanzer@BootstrapWorld.org](mailto:schanzer@BootstrapWorld.org).

## Check for Mistakes in this Word Problem: offscreen?

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

### Contract and Purpose Statement

Every contract has three parts...

|   |  |   |                      |               |                |
|---|--|---|----------------------|---------------|----------------|
| ; | <u>offscreen?</u>  | : | <u>number</u>        | →             | <u>boolean</u> |
|   |  |   | <i>function name</i> | <i>domain</i> | <i>range</i>   |
| ; | Given an x-coordinate, returns true if the coordinate is less than -50 or greater than 690 |   |                      |               |                |

*what does the function do?*

### Examples

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

|           |                      |                 |   |                                   |   |
|-----------|----------------------|-----------------|---|-----------------------------------|---|
| (EXAMPLE( | <u>offscreen?</u>    | <u>60</u>       | ) | (true)                            | ) |
|           | <i>function name</i> | <i>input(s)</i> |   | <i>what the function produces</i> |   |
| (EXAMPLE( | <u>offscreen?</u>    | <u>800</u>      | ) | (false)                           | ) |

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

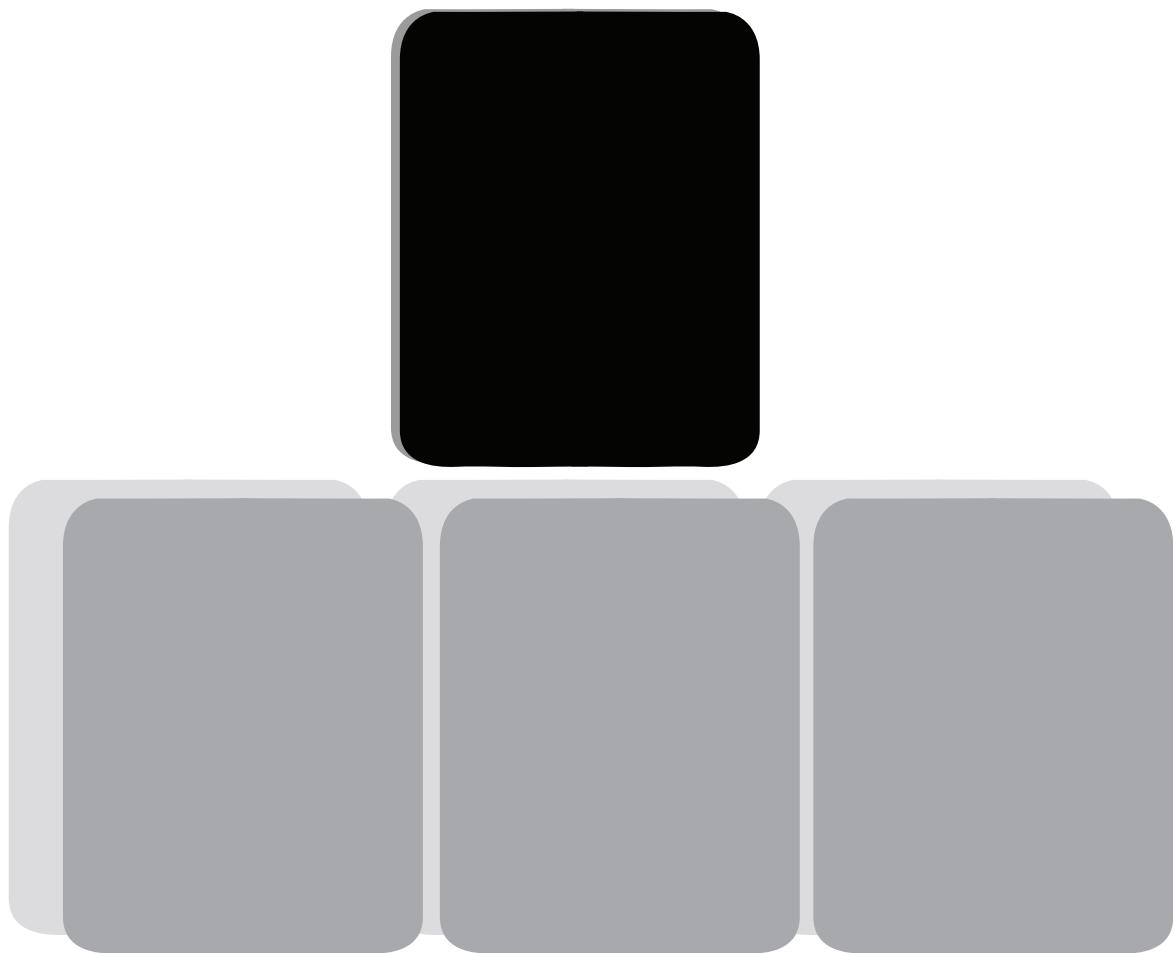
### Definition

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

|          |   |                  |   |
|----------|---|------------------|---|
| (define( | <u>off-screen</u>                                     | <u>x-coord</u>   | ) |
|          | <i>function name</i>                                  | <i>variables</i> |   |
| (and     | <u>(&lt; x-coord -50)</u> <u>(&gt; x-coord 690)</u> ) |                  |   |

*what the function does with those variables*

# 07 Conditional Branching

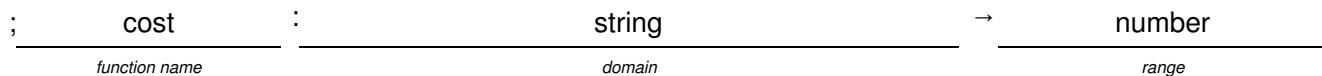


# Word Problem: cost

**Directions:** Luigi's Pizza has hired you as a programmer. They offer Cheese (\$9.00), Pepperoni (\$10.50), Chicken (\$11.25) and Broccoli (\$10.25). Write a function called `cost` which takes in the name of a topping and outputs the price of a pizza with that topping.

## Contract and Purpose Statement

Every contract has three parts...



; Produce cost of a pizza with given topping

*what does the function do?*

## Examples

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

|           |               |             |   |                            |   |
|-----------|---------------|-------------|---|----------------------------|---|
| (EXAMPLE( | <u>cost</u>   | "cheese"    | ) | 9                          | ) |
|           | function name | input(s)    |   | what the function produces |   |
| (EXAMPLE( | <u>cost</u>   | "pepperoni" | ) | 10.5                       | ) |
|           | function name | input(s)    |   | what the function produces |   |
| (EXAMPLE( | <u>cost</u>   | "chicken"   | ) | 11.25                      | ) |
|           | function name | input(s)    |   | what the function produces |   |
| (EXAMPLE( | <u>cost</u>   | "broccoli"  | ) | 10.25                      | ) |
|           | function name | input(s)    |   | what the function produces |   |

## Definition

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

(define(

|               |   |           |   |
|---------------|---|-----------|---|
| <u>cost</u>   | : | topping   | ) |
| function name |   | variables |   |

(cond

|                   |  |
|-------------------|--|
| <u>          </u> |  |
|-------------------|--|

[(string=? "cheese" topping)      9      ]

[(string=? "pepperoni" topping)      10.5      ]

[(string=? "chicken" topping)      11.25      ]

[(string=? "broccoli" topping)      10.25      ]

[else:            1000000      ]))

## Word Problem: update-player

**Directions:** Write a function called `update-player`, which takes in the player's y-coordinate and the name of the key pressed, and returns the new y-coordinate.

# **Contract and Purpose Statement**

*Every contract has three parts....*

|   |   |               |   |                                   |
|---|---|---------------|---|-----------------------------------|
| ; update-player                                   | : | number string | → | number                            |
| <i>function name</i>                              |   | <i>domain</i> |   | <i>range</i>                      |
| ; Produce new y-coordinate depending on key press |   |               |   | <i>what does the function do?</i> |
|   |   |               |   |                                   |

## Examples

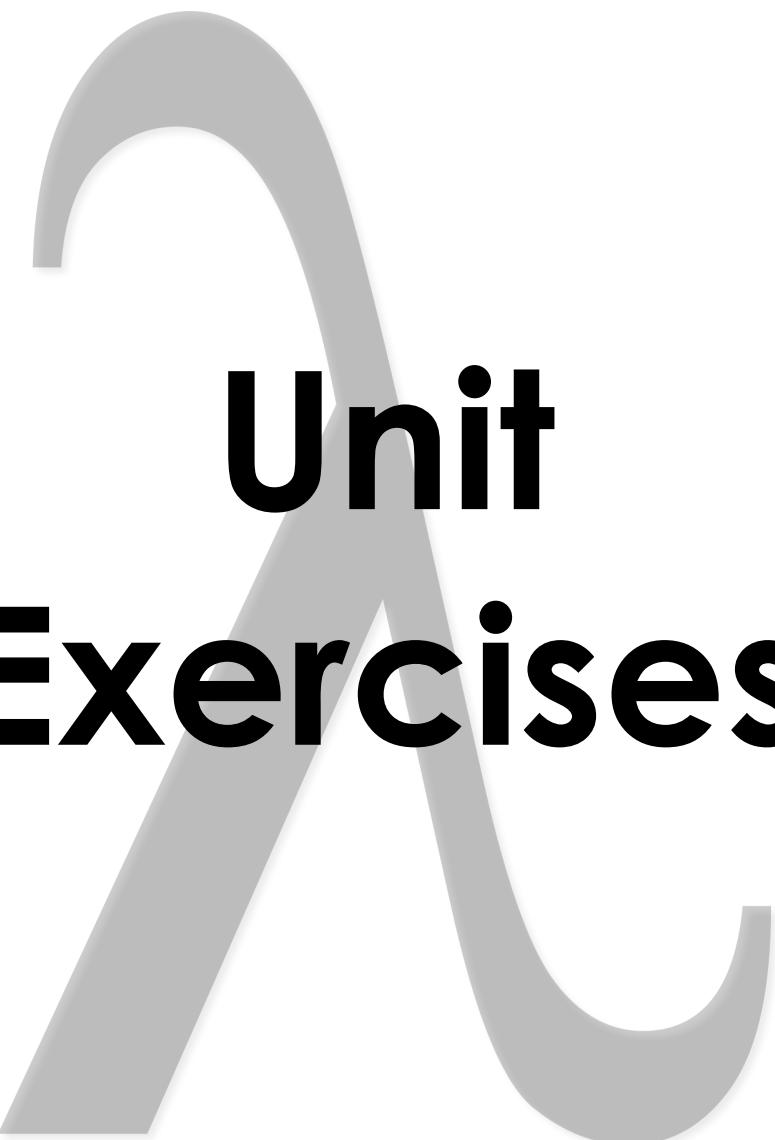
*Write some examples, then circle and label what changes...*

|           |                      |                 |   |                                   |   |
|-----------|----------------------|-----------------|---|-----------------------------------|---|
| (EXAMPLE( | <u>update-player</u> | 320 "up"        | ) | (+ 320 20)                        | ) |
|           | <i>function name</i> | <i>input(s)</i> |   | <i>what the function produces</i> |   |
| (EXAMPLE( | <u>update-player</u> | 100 "up"        | ) | (+ 100 20)                        | ) |
|           | <i>function name</i> | <i>input(s)</i> |   | <i>what the function produces</i> |   |
| (EXAMPLE( | <u>update-player</u> | 320 "down"      | ) | (- 320 20)                        | ) |
|           | <i>function name</i> | <i>input(s)</i> |   | <i>what the function produces</i> |   |
| (EXAMPLE( | <u>update-player</u> | 100 "down"      | ) | (- 100 20)                        | ) |
|           | <i>function name</i> | <i>input(s)</i> |   | <i>what the function produces</i> |   |

## Definition

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

```
(define( update-player y key )  
       function name           variables  
  
(cond _____  
  
[(string=? "up" key) (+ y 20)]  
  
[(string=? "down" key) (- y 20)]  
  
[else: y]))
```



# **Unit Exercises**

## Check for Mistakes in this Word Problem: scale-image

**Directions:** Write a function which 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") )  
function name input(s) what the function produces

(EXAMPLE( scale-triangle (triangle 20 "solid" "blue") "smaller" )  
function name input(s)  
(triangle 10 "solid" "blue")  
what the function produces )

### Definition

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

(define( scale-image original-image scale-factor )  
function name variables

(cond  
\_\_\_\_\_

[(string=? scale-factor "bigger") (scale 2 original-image) ]

[(string=? scale-factor "smaller") (scale 0.5 original-image) ]

[else: original-image \_\_\_\_\_ ]))

## Check for Mistakes in this Word Problem: state-tax

**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, returns the 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, given variable names to all your input values...

(define( state-tax state price )  
function name variables

(cond  
[ (string=? state "Pennsylvania") (\* 0.03 price) ]

[ (string=? state "New York") (\* 0.08 price) ]

[ (string=? state "Delaware") (\* 0.0 price) ]

[ else: (\* 0.05 price) ]))

## Check for Mistakes in this Word Problem: flower-name

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

(EXAMPLE( flower-name "tulip" ) "purple")  
function name input(s) what the function produces

(EXAMPLE( flower-name "yellow" ) "daisy")  
function name input(s) what the function produces

### Definition

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

(define( flower-name color ))  
function name variables

(cond \_\_\_\_\_)

[(string=? color "red") "rose"]

[(string=? color "purple") "tulip"]

[(string=? color "yellow") "daisy"]

[else:"That flower isn't in the garden!"]))

## Check for Mistakes in this Word Problem: turkey-price

**Directions:** The grocery store charges \$2/pound for a turkey for the first 5 pounds, and then \$1/pound for every pound after that. Write a function that takes in the weight of a turkey and returns how much it will cost.

### Contract and Purpose Statement

Every contract has three parts...

; turkey-price : weight → number  
function name domain range  
; Return the price of the turkey.  
what does the function do?

### Examples

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

(EXAMPLE( turkey-price      3      )    (\* 2 3) )  
function name      input(s)      what the function produces  
(EXAMPLE( turkey-price      6      )    (+ 10 (\* 1 (- 6 5))) )  
function name      input(s)      what the function produces

### Definition

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

(define( turkey-price      weight      )  
function name      variables  
(cond  
\_\_\_\_\_  
[(< weight 5)      (\* 2 weight) ]  
[else:(+ 10 (\* 1 (- weight 5))) ]))

# 08 Collision Detection

# collision



distance

# Word Problem: line-length

**Directions:** Write a function called 'line-length', which takes in two numbers and returns the \*positive difference\* between them. It should always subtract the smaller number from the bigger one, and if they are equal it should return zero.

## Contract and Purpose Statement

Every contract has three parts...

; line-length : number number → number  
function name domain range

; Produce positive difference between two given numbers

*what does the function do?*

## Examples

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

(EXAMPLE( line-length 10 5 ) (- 10 5) )  
function name input(s) what the function produces

(EXAMPLE( line-length 2 8 ) (- 8 2) )  
function name input(s) what the function produces

## Definition

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

(define( line-length a b )  
function name variables

(cond  
\_\_\_\_\_

[(> a b) (- a b) ]  
\_\_\_\_\_]

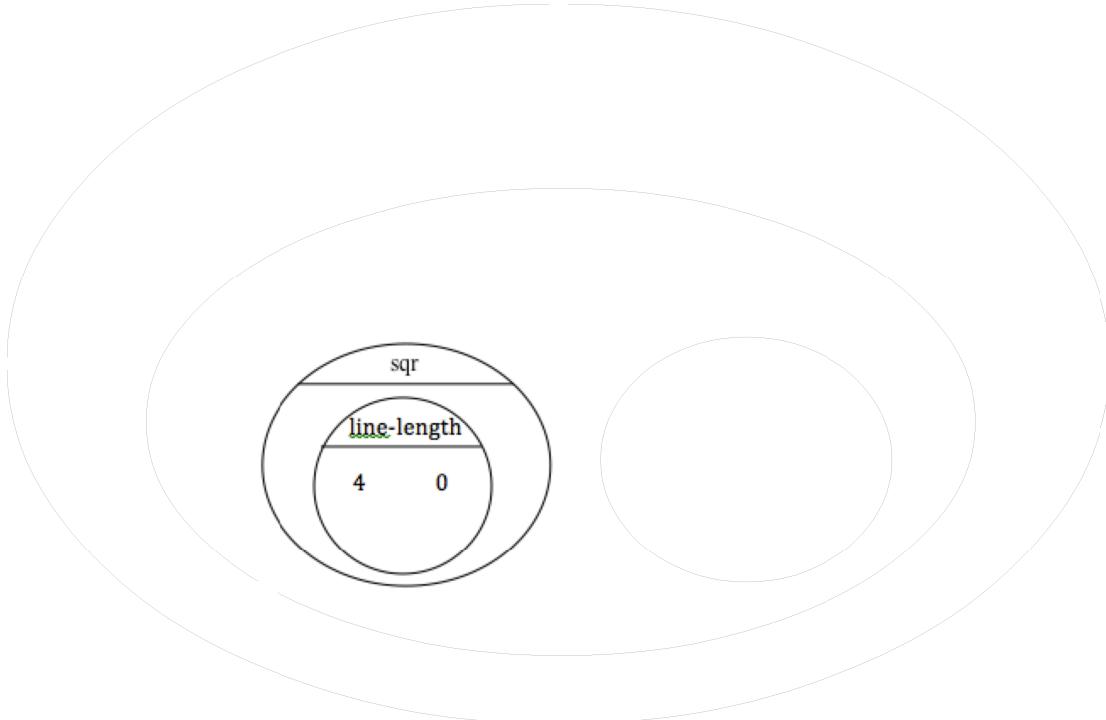
[(< a b) (- b a) ]))  
\_\_\_\_\_

## The Distance Formula (an example)

The distance between the points  $(0, 0)$  and  $(4, 3)$  is given by:

$$\sqrt{(line-length\ 4\ 0)^2 + (line-length\ 3\ 0)^2}$$

Convert the formula above into a Circle of Evaluation. (We've already gotten you started!)



Convert the Circle of Evaluation to code, then label the numbers with  $(x1,y1)$  &  $(y1,y2)$ :

## Word Problem: distance

**Directions:** Write a function *distance*, which takes *FOUR* inputs:

- *px*: The *x*-coordinate of the player
- *py*: The *y*-coordinate of the player
- *cx*: the *x*-coordinate of another game character
- *cy*: the *y*-coordinate of another game character

*It should return the distance between the two, using the Distance formula. (HINT: look at what you did on the previous page!)*

### Contract and Purpose Statement

*Every contract has three parts...*

; distance : number number number number → number  
function name domain range  
; Produce distance between two points with given coordinates  
what does the function do?

### Examples

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

(EXAMPLE( distance 0 0 3 4 )  
function name input(s)  
( $\sqrt{(\text{line-length } 3 \ 0) + (\text{line-length } 4 \ 0)}$ )  
what the function produces  
(EXAMPLE( distance 10 20 13 24 )  
function name input(s)  
( $\sqrt{(\text{line-length } 13 \ 10) + (\text{line-length } 24 \ 20)}$ )  
what the function produces

### Definition

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

(define( distance px py cx cy )  
function name variables  
( $\sqrt{(\text{sqr } (\text{line-length } px \ cx)) + (\text{sqr } (\text{line-length } py \ cy)))}$ )  
what the function does with those variables

## Word Problem: collide?

**Directions:** Write a function *collide?*, which takes FOUR inputs:

- *px*: The x-coordinate of the player
- *py*: The y-coordinate of the player
- *cx*: the x-coordinate of another game character
- *cy*: the y-coordinate of another game character

Are the coordinates of the player within 50 pixels of the coordinates of the other character?

### Contract and Purpose Statement

Every contract has three parts...

; collide? : number number number number → boolean  
function name domain range

; Is the Player (px, py) within 50 pixels of another Character (cx, cy)?

*what does the function do?*

### Examples

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

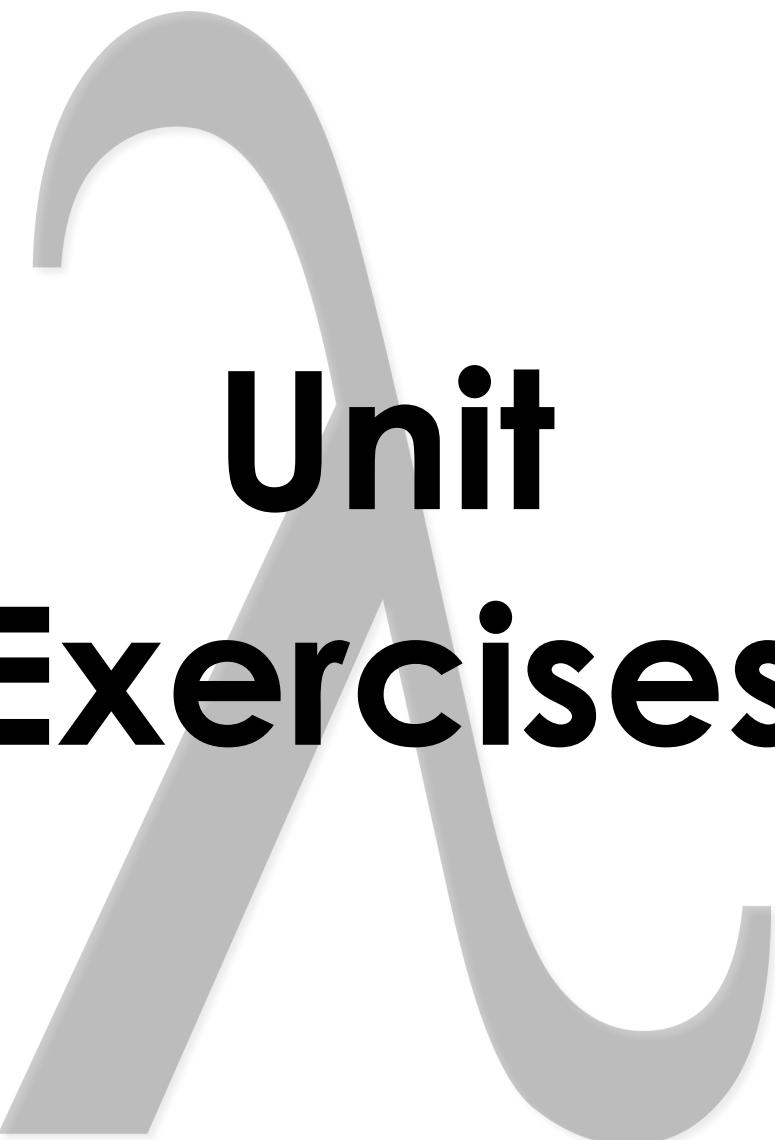
(EXAMPLE( collide? 25 75 30 270 ) (< (distance 25 75 30 270) 50)  
function name input(s) what the function produces

(EXAMPLE( collide? 25 75 100 75 ) (< (distance 25 75 100 75) 50)  
function name input(s) what the function produces

### Definition

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

(define( collide? px py cx cy )  
function name variables  
(< (distance px py cx cy) 50)  
what the function does with those variables )



# **Unit Exercises**

## Check for Mistakes in this Word Problem: late-to-class?

**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 coordinates of my location and a classroom and returns true if the distance is more

*what does the function do?*

### Examples

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

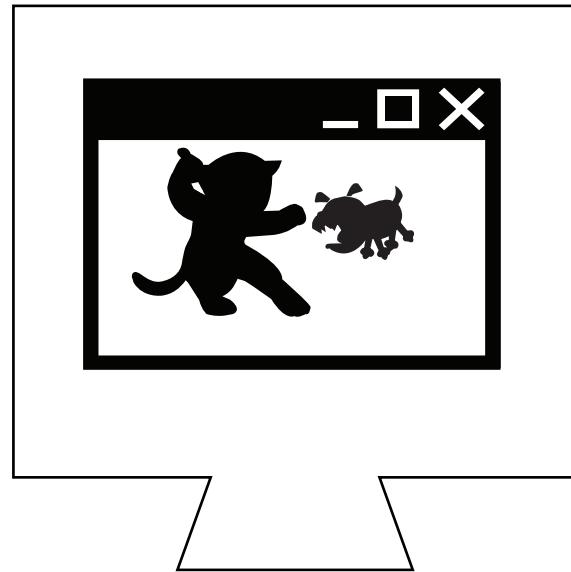
(EXAMPLE((late-to-class? 40 5) ) ( > 25 (distance 40 55 65 80)) )  
function name input(s) what the function produces

(EXAMPLE((late-to-class? 40 5) ) ( < 25 (distance 40 55 65 80)) )  
function name input(s) what the function produces

### Definition

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

(define( late-to-class? student-x student-y school-x school-y )  
function name variables  
((< 25 (distance student-x student-y school-x school-y)))  
*what the function does with those variables*)



## 09 Presentation Preparation



# Lesson 9

Catchy Intro:

---

---

---

Name, Age, Grade:

---

Game Title:

---

---

---

---

Back Story:

---

---

---

---

Characters:

---

---

---

Explain a piece of your code:

---

---

---

---



## **Presentation Feedback**

*For each question, circle the answer that fits best.*

Was the introduction catchy?      No way!      A little.      Definitely!

Did they talk about their characters?    No way!      A little.      Definitely!

Did they explain the code well?    No way!      A little.      Definitely!

Did they speak slowly enough?    No way!      A little.      Definitely!

Did they speak loudly enough?    No way!      A little.      Definitely!

Were they standing confidently?    No way!      A little.      Definitely!

Did they make eye contact?    No way!      A little.      Definitely!

## **Presentation Feedback**

*For each question, circle the answer that fits best.*

Was the introduction catchy?      No way!      A little.      Definitely!

Did they talk about their characters?    No way!      A little.      Definitely!

Did they explain the code well?    No way!      A little.      Definitely!

Did they speak slowly enough?    No way!      A little.      Definitely!

Did they speak loudly enough?    No way!      A little.      Definitely!

Were they standing confidently?    No way!      A little.      Definitely!

Did they make eye contact?    No way!      A little.      Definitely!

## Word Problem: red-shape

**Directions:** Write a function called `red-shape`, which takes in the name of a shape and draws that shape (solid and red). Add an `else` clause that produces a sensible output.

### Contract and Purpose Statement

Every contract has three parts...

|   |  |   |        |   |       |
|---|--|---|--------|---|-------|
| ; | red-shape                                  | : | string | → | image |
|   |  |   | domain |   | range |
| ; | Create a solid red shape of the given kind |   |        |   |       |
|   | what does the function do?                 |   |        |   |       |

### Examples

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

|           |               |             |   |                                |   |
|-----------|---------------|-------------|---|--------------------------------|---|
| (EXAMPLE( | red-shape     | "circle"    | ) | (circle 50 "solid" "red")      | ) |
|           | function name | input(s)    |   | what the function produces     |   |
| (EXAMPLE( | red-shape     | "triangle"  | ) | (triangle 50 "solid" "red")    | ) |
|           | function name | input(s)    |   | what the function produces     |   |
| (EXAMPLE( | red-shape     | "rectangle" | ) | (rectangle 99 9 "solid" "red") | ) |
|           | function name | input(s)    |   | what the function produces     |   |
| (EXAMPLE( | red-shape     | "star"      | ) | (triangle 50 "solid" "red")    | ) |
|           | function name | input(s)    |   | what the function produces     |   |

### Definition

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

|                               |                                 |           |   |
|-------------------------------|---------------------------------|-----------|---|
| (define(                      | red-shape                       | shape     | ) |
|                               | function name                   | variables |   |
| (cond                         |                                 |           |   |
| [(string=? "circle" shape)    | (circle 50 "solid" "red")]      |           |   |
| [(string=? "triangle" shape)  | (triangle 50 "solid" "red")]    |           |   |
| [(string=? "rectangle" shape) | (rectangle 99 9 "solid" "red")] |           |   |
| [(string=? "star" shape)      | (star 50 "solid" "red")]        |           |   |
| [else:(text 20 "????" "red")  | ]))                             |           |   |

## Translating into Algebra

### Value Definitions

| Racket Code                   | Algebra   |
|-------------------------------|-----------|
| (define x 10)                 | $x = 10$  |
| (define y (* x 2))            | $y = x^2$ |
| (define z (+ x y))            |           |
| (define age 14)               |           |
| (define months (* age 12))    |           |
| (define days (* months 30))   |           |
| (define hours (* days 24))    |           |
| (define minutes (* hours 60)) |           |

### Function Definitions

| Racket Code  | Algebra   |
|--|---|
| (define (area length width)<br>(* length width))                                 | $\text{area}(\text{length}, \text{width}) = \text{length} * \text{width}$ |
| (define (circle-area radius)<br>(* pi (sqr radius)))                             |   |
| (define (distance x1 y1 x2 y2)<br>(sqrt (+ (sqr (- x1 x2))<br>(sqr (- y1 y2))))) |   |

# Design Recipe

A rocket is flying from Earth to Mars at 80 miles per second. Write a function that describes the distance  $D$  that the rocket has traveled, as a function of time  $t$ .

## I. Contract+Purpose Statement

Every contract has three parts:

; D : \_\_\_\_\_ -> \_\_\_\_\_  
name Domain Range  
;  
What does the function do?

## II. Give Examples

Write an example of your function for some sample inputs

D( 1 ) = \_\_\_\_\_  
Use the function here What should the function produce?  
  
D( 2 ) = \_\_\_\_\_  
Use the function here What should the function produce?  
  
D(    ) = \_\_\_\_\_  
Use the function here What should the function produce?  
  
= \_\_\_\_\_

## III. Definition

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

D(    ) = \_\_\_\_\_

# Design Recipe

A rocket is traveling from Earth to Mars at 80 miles per second. Write a function that describes the time the rocket has been traveling, as a function of distance.

## I. Contract+Purpose Statement

Every contract has three parts:

; \_\_\_\_\_ : \_\_\_\_\_ -> \_\_\_\_\_  
; name Domain Range  
;  
; \_\_\_\_\_  
What does the function do?

### II. Give Examples

Write an example of your function for some sample inputs.

=  
Use the function here      What should the function produce?

=  
Use the function here      What should the function produce?

=  
Use the function here      What should the function produce?

### **III. Definition**

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

# Design Recipe

A rocket leaves Earth, headed for Mars at 80 miles per second. **At the exact same time**, an asteroid leaves Mars traveling towards Earth, moving at 70 miles per second. If the distance from the Earth to Mars is 50,000,000 miles, how long will it take for them to meet?

## I. Contract+Purpose Statement

Every contract has three parts:

; \_\_\_\_\_ : \_\_\_\_\_  $\rightarrow$  \_\_\_\_\_  
; name Domain Range  
;  
*What does the function do?*

### **III. Give Examples**

Write an example of your function for some sample inputs

|                       |                                   |
|-----------------------|-----------------------------------|
| =                     |                                   |
| Use the function here | What should the function produce? |
| =                     |                                   |
| Use the function here | What should the function produce? |
| =                     |                                   |
| Use the function here | What should the function produce? |
| =                     |                                   |
| Use the function here | What should the function produce? |

### **III. Definition**

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

# Design Recipe

## I. Contract+Purpose Statement

Every contract has three parts:

; \_\_\_\_\_ : \_\_\_\_\_ -> \_\_\_\_\_  
            name                        Domain                        Range  
;  
What does the function do?

## **II. Give Examples**

Write an example of your function for some sample inputs

|                       |                                   |
|-----------------------|-----------------------------------|
| =                     |                                   |
| Use the function here | What should the function produce? |
| =                     |                                   |
| Use the function here | What should the function produce? |
| =                     |                                   |
| Use the function here | What should the function produce? |
| =                     |                                   |
| Use the function here | What should the function produce? |

### **III. Definition**

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

# Design Recipe

## I. Contract+Purpose Statement

Every contract has three parts:

; \_\_\_\_\_ : \_\_\_\_\_ -> \_\_\_\_\_  
                name                            Domain                            Range  
;  
*What does the function do?*

## **II. Give Examples**

Write an example of your function for some sample inputs

|                       |                                   |
|-----------------------|-----------------------------------|
| =                     |                                   |
| Use the function here | What should the function produce? |
| =                     |                                   |
| Use the function here | What should the function produce? |
| =                     |                                   |
| Use the function here | What should the function produce? |
| =                     |                                   |
| Use the function here | What should the function produce? |

### **III. Definition**

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

## Contracts

## Contracts