The Design Recipe

Directions: Write a function `split-tab` that takes in a cost and the number of people sharing the bill and splits the cost equally.

Contract and Purpose Statement

Every contract has three parts…

```
# split-tab:: Number, Number -> Number
```

function name | *domain* | *range*

```
# Takes in a cost and a number of people and divides the cost by the number of people, returning the value.
```

what does the function do?

Examples

Write some examples, then circle and label what changes…

examples:

```
split-tab ( 200, 10 ) is 200 / 10
split-tab ( 500, 5 ) is 500 / 5
```

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

end

Definition

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

```
fun split-tab( cost, people ):
  cost / people
end
```

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

Directions: Write a function `num-cube` that takes in a number and returns the cube of that number.

Contract and Purpose Statement

Every contract has three parts…

```
# num-cube:: Number -> Number
```

function name | *domain* | *range*

```
# Takes in a number, cubes it and returns that value.
```

what does the function do?

Examples

Write some examples, then circle and label what changes…

examples:

```
num-cube ( 1 ) is 1 * (1 * 1)
num-cube ( 3 ) is 3 * (3 * 3)
```

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

end

Definition

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

```
fun num-cube( n ):
  n * (n * n)
end
```

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