Scheme Conditionals

Returning a Value - Part 1

1. Create the file SchemeFlipped1.scm. Inside that file, create a method named "is-negative". The function should return true (`#t`) if a negative number (0 is considered positive) is passed in, and false otherwise. The function needs to contain either a "cond" or an "if" branch structure.

   Call your function three times in main with the hard-coded values -100, 0, and 100.

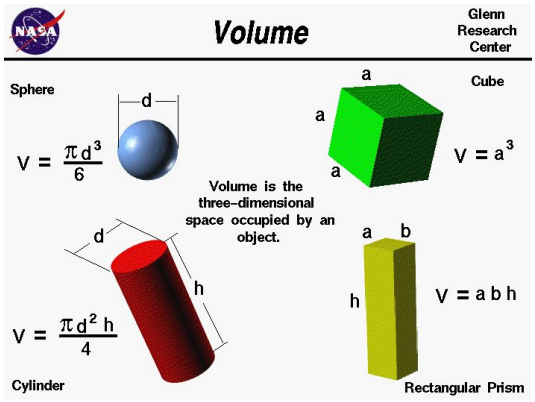
Returning a Value - Part 2

2. Create the file SchemeFlipped2.scm using Emacs. Inside that file, create a method named "letterGrade", which returns a letter value depending on the parameter entered. The function needs to contain either a "condo" or an "if" branch structure. (Hint: cond may be helpful). Use the following grade scale: 90-100 A, 89-80 B, 79-70 C, 69-60 D, and 59-0 F as your setup.

   Run your program with 0, 75, and 95 in main.

Returning a Value - Part 3

3. Create the file SchemeFlipped3.scm. Inside that file, create the methods named:
   a. "sphereVol"
      i. Takes one parameter: diameter
      ii. Calculates and returns the volume of a sphere (equation below)
   b. "rectVol"
      i. Takes three parameters: height, width, and length
      ii. Calculates and returns the volume of a Rectangular Prism (equation below)
   c. "isContained"
      i. Takes four parameters: diameter, height, width, and length
      ii. Will determine IF the sphereVol is LESS THAN OR EQUAL TO the rectVol
      iii. Must use an "if" or "cond"
      iv. Returns `#t` if the sphere has less volume than the rectangular prism



	Sphere			Rectangular Prisms			
Diameter		Volume		Height	Width	Length	Volume
15		1766.25		10	8	45	3600
15.5		1948.828		10	8	48	3840
16		2143.573		10	8	53	4240
16.5		2350.879					
17		2571.137					
17.5		2804.74					
18		3052.08					
18.5		3313.55					
19		3589.543					
19.5		3880.451					
20		4186.667					
20.5		4508.582					
21		4846.59					
21.5		5201.083					
22		5572.453					

Here is the scenario. A manufacturer of “product A”, stores their product in a Sphere. While the distributor uses a Rectangular Prism 18 wheel (big rig) trucks seen along the highway. The trouble is that to save money, the distributor contracts out many of the trucks, and they have container cars of various sizes. The operator to fill the truck needs to check if the truck is big enough for the Volume of material about to be transferred when ONLY given the diameter of the Sphere and the height, length, and width of the Rectangular Prism.

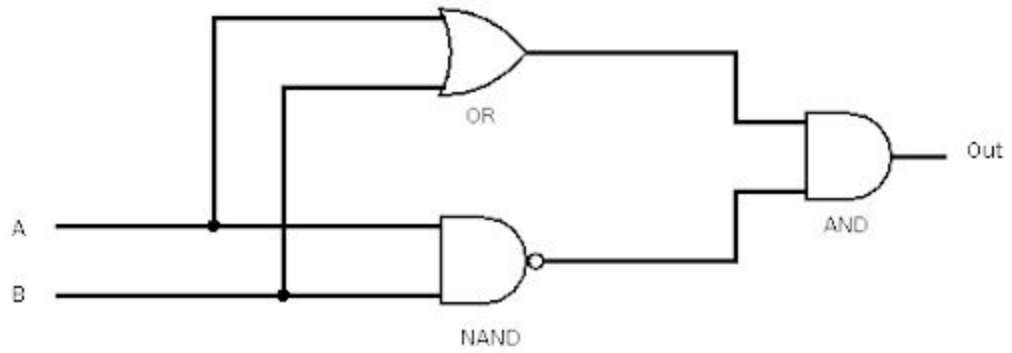
I have created a few scenarios and values above. Your program needs to reflect the same values in return.
Hard code **the first three rows** into your program. The Main function should display if the truck can fit the sphere for each call.

Returning a Value - Part 4

4. Create a method named “xorGate”, which takes in two boolean (`#t` or `#f`) values and outputs the appropriate truth value. You must do this by using only not, or, and and operators. Below is the truth table for the XOR gate for reference.

INPUT A	**INPUT B**	**OUTPUT**
F	F	F
F	T	T
T	F	T
T	T	F

Below is the XOR gate schematic for reference. You may create separate methods for each gate (i.e. orGate, nandGate) and use them in the xorGate method.



After this, create a method called “stateValue” that takes in a boolean value (`#t` or `#f`). If the input is `#t`, then the method will print “`The value is #t`”. If the input is `#f`, then the method will print “`The value is #f`”. In your code, the xorGate must take in any combination of two `#t`/`#f` values. The the output of that must be used as an input for stateValue so that it can print whether the output value is `#t` or `#f`.

Returning a Type

5. Create the file SchemeFlipped5.scm using Emacs. Write a function that takes four arguments. The first two arguments are functions; f1(x) and f2(y), where f1(x) = x+1 and f2(y) = y+2. The third and fourth arguments are the values x and y. If both values are equal, and both are equal to 1, then return f1(x) + f2(y). If either value is the letter “k”, return “`Keishla`”. If x is greater than y, return f1(x) * f2(y). If none of these conditions are true, return x * y. YOU SHOULD ONLY USE **THREE** “IF” keywords, and you should not use any COND statements. You should also use lambda functions to define f1(x) and f2(y).

   Run your program with the following values:
   Input: x = 4, y = 1. Expected Output = 15