Question: How do you wrap a list inside an object type?

Answer: Good question! It should be: Object("(1 2 3)"). The Object class uses a vector to store the elements of a list. If there are nested lists, it is essentially a vector of Objects that use vectors.

Question: I was playing around on a normal cpp file that uses objects. After having looked at the listop function in Object.h, I realized how I would need to write from pl460 to c++, so I was trying to wrap it into an object. That clarified how to do it!

Answer: Yes! C++ programs that use the Object class do not need to be generated by your P3.out executable. You can write them yourself to familiarize yourself with the class. Very good idea!

Question: This PL460 code creates a new scope:
(let ((num 2))
  (display num) (newline)
  (+ 2 num)
)
Should there be a scope in C++ as well?

Answer: Yes. You can create a new scope with a set of curly braces.

Question: While (display (round 2.6)) displays 3.0, cout << round (Object (“2.6”)); displays 3.

Answer: Fixed.

Question: While (display (cons '(1 2 3) 4)) displays ((1 2 3) . 4), cout << listop (“cons”, Object (“(1 2 3)”), Object (“3”)); throws the exception: ERROR: Wrong type for list operation function: cons (list or integer)

Answer: Scheme actually creates a string containing the dotted pair. Exception handling is another way of indicating a problem. It is fine.

Question: The generated code:
   cout << Object("0") == Object("0");
produces a large number of errors when compile with g++.

Answer: Wrapping the equality test in parenthesis:
   cout << (Object("0") == Object("0"));
eliminates these errors. (Why would this be true?)
Question: While (display (not(null? '(5)))) displays #t, cout << ! (nullp (Object("(5)"))); displays 1. Why would this be true?

Answer: I'm looking into it….

Question: In class when talking about project 3 you said we have a stack we can access to determine the last few tokens. I'm confused as to what stack outside of the call stack you might be talking about. I would appreciate some help with this.

Answer: You should not need another stack (in addition to the call stack). But, if you find one to be beneficial, that is fine. I would recommend using vectors to store additional stacks.

Question: Q

Answer: A