CS 460 – Project 3 - FAQ

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: If the PL460 input contains a very simple main function, what should the generated C++ output contain?

Answer:

```
pl460 : (define (main
                0
        )

C++   : int main ()
        {
                Object __RetVal;
                __RetVal = 0;
                return 0;
        }
```

Question: If the PL460 input contains a very simple function that is not main, what should the generated C++ output contain?

Answer:

```
pl460 : (define (aFunction a)
                 a
        )

C++   : Object aFunction (Object a)
        {
                 Object __RetVal;
                 __RetVal = a;
                 return __Retval;
        }
```

Question: I know that I need to put calls to cg->WriteCode in my SyntaxAnalyzer. Where should the calls go for the functions shown in the 2 previous questions?

Answer:

```
In SyntacticalAnalyzer::define

        string functionName;

After seeing the first IDENT_T token

        if (functionName == "main")
                cg->WriteCode (0, "int ");
        else
                cg->WriteCode (0, "Object ");
        cg->WriteCode (0, functionName + " (");

After seeing all of the statements

        if (functionName == "main")
                cg->WriteCode (1, "return 0;\n");
        else
                cg->WriteCode (1, "return __RetVal;\n");
        cg->WriteCode (0, "}\n");
```

Question: When should the result of an arithmetic expression be stored in a return value? When should it not.

Answer: Return values should not be generated for display or newline statements. They also should not be used for if control structures or for embedded arithmetic expressions.

```
pl460 : (define (aFunction a)
                (display (+ a 5))
                (newline)
                (+ a 5)
        )
C++   : Object aFunction (Object a)
        {
                Object __RetVal;
                cout << (a + 5);
                cout << endl;
                __RetVal = a + 5;
                return __Retval;
        }
```

Question: When I'm converting I'm a little confused on how the object should be handled and how it would be related to the cout. As of right now i'm printing the cout << Object(5); in the action function inside the DISPLAY_T else if. but it's only like this:

```
PL460 : (define (main)
  (display 5)
```

```
Becomes
        C++ : int main ()
        {
                Object __RetVal;
                cout << Object("5");
                cout << endl;
```

Answer: This looks accurate – keep going!

Question: Q

Answer: A