Sami Meharzi
Lab 3
CSCI 3155

# Write up!

2(a)

One example of a test case of a test case where dynamic scoping will offer a different answer than static scoping would be:

```
const x = 10;
const plus = function(x){ return function(y){ return x + y; } };
jsy.print(plus(5)(5));
```

The static result would be: 10
The dynamic result would be: 15

The static case doesn't take into account the earlier assignment of v1, while the dynamic version will immediately account for v1's value, then use it for the function.

3(c)

The system is deterministic because it always goes from left to right. The eval function is written to interpret expressions in such a manner. If the same expression were to run multiple times, the results will be the same and consistent.

4

For e1 + e2, e1 would come first, then e2, and then +. If it were reversed this we could simply evaluate to e2 being first, then e1, and finally +. This can be done by reversing the eval function. There are three different case methods for adding, and ambiguity is avoided by taking dealing with all possible situations. If the addition is e1 + e2, the first step would be check if e1 or e2 are strings. If e1 is a string, and e2 isn't, use the DoPlusString1 case in figure 7. If e2 is a string and e1 isn't, use the DoPlusString2 case. Otherwise, neither of them are strings, then transform both into numbers and return the sum.

5(a)

The “|| =” symbol in Ruby is a fine example of useful short-circuit evaluation. For example “x || = y” will check if x is set to a value, and if so, the evaluation is short-circuited and x is returned; if not, then x is set to the value of y.

5(b)

e1 && e2 short-circuits because the program first checks if e1 is valid, if not, false is returned since there isn’t a way to proper use the And statement. The And statement cannot be true with one false element.