Problem 1

a)

x = 45

const g = (x) => 12

g(10)

print x

In the case of dynamic scoping, this would return 12, as g maps x to 12. In the case of static scoping, this would return 45 because all instances of x would be substituted with 45.

Problem 3

The evaluation is deterministic as the Search rules evaluate e1 before e2 in all cases. Values are not returned until the Do rules.

Problem 4

The expression e1 + e2 will first hit the SearchBinary rule to evaluate e1 to a value before evaluating e2. In order to change the evaluation order, we would need to change the SearchBinary rule to evaluate e2 first. After both expressions have been evaluated the doArith rule is executed.

Problem 5

a)

Consider the expression e1 && e2. If we evaluate e1 to value v1 and if toBoolean(v1) == false, we know the expression should return false. This short-circuit evaluation means that we never have to evaluate e2.

b)

According to the figure's small step operational semantics, e1 && e2 will short circuit by the SearchBinary and DoAndFalse rules. According to SearchBinary, e1 is evaluated first and by DoAndFalse, if e1 is evaluated to v1 and if toBoolean(v1) == false, the evaluation will short circuit and return v1.