CSCI 3155: Principles of Programming Languages
Spring 2014: February 25, 2014 **Quiz 3**

Name: ______________________

There are 2 questions for 20 points. 15 minutes individually and 10 minutes of group time.

1. **Small-Step Substitution Semantics: Arithmetic Expressions with Binding**. Consider an arithmetic expression language with binding:

$$e \quad ::= \quad n \mid x \mid e_1 + e_2 \mid e_1 * e_2 \mid \mathbf{const}\ x = e_1;\ e_2$$

where n and x correspond to numbers (e.g., 0, 1, 42) and variable identifiers (e.g., x, y), respectively. The values of this language are simply numbers n. The operators $+$ and $*$ are the usual arithmetic operators. The **const** $x = e_1;\ e_2$ expression binds a variable x to the value e_1, which is then in scope in expression e_2.

A one-step evaluation judgment for this language has the following form: $e \longrightarrow e'$. Informally, this judgment says "Closed expression e takes one step of evaluation to expression e'." This judgment gives a small-step operational semantics for this language, which we define in this question.

(a) $\boxed{\text{3 points}}$ Suppose that rules to evaluate expressions of the form $e_1 + e_2$ are as follows:

$$\frac{n' = n_1 \mathbf{+} n_2}{n_1 + n_2 \longrightarrow n'} \qquad \frac{e_1 \longrightarrow e_1'}{e_1 + e_2 \longrightarrow e_1' + e_2} \qquad \frac{e_2 \longrightarrow e_2'}{e_1 + e_2 \longrightarrow e_1 + e_2'}$$

where the **+** in the first rule is bolded to emphasize that it is the mathematical plus.

What is stated about the order of evaluation for $+$ in the above rules? Explain in 1–2 sentences.

> **Solution:** These rule say that the evaluation of $e_1 + e_2$ is non-deterministic. The evaluator can interleave steps of evaluating e_1 or e_2.

(b) $\boxed{\text{5 points}}$ Give rules for evaluating $e_1 * e_2$ such that the evaluation short-circuits if e_1 is 0.

> **Solution:**
>
> $$\textsc{DoTimes}\ \frac{n' = n_1 \mathbf{*} n_2 \qquad n_1 \neq 0}{n_1 * n_2 \longrightarrow n'} \qquad \textsc{DoTimesZero}\ \frac{}{0 * e_2 \longrightarrow 0} \qquad \textsc{StepTimesLeft}\ \frac{e_1 \longrightarrow e_1'}{e_1 * e_2 \longrightarrow e_1' * e_2}$$
>
> $$\textsc{StepTimesRight}\ \frac{e_2 \longrightarrow e_2' \qquad n_1 \neq 0}{n_1 * e_2 \longrightarrow n_1 * e_2'}$$
>
> The highlighted premise ensures that the DoTimesZero rule is the only one that applies when the left expression evaluates to 0.

(c) $\boxed{\text{4 points}}$ Give rules for evaluating **const** $x = e_1;\, e_2$ expressions where first e_1 must be evaluated to a value and then e_2 is evaluated. Recall that x is in scope in e_2 and refers to the value of e_1. You may use the notation $e[e'/x]$ for the capture-avoiding substitution in e of e' for x.

Solution:

$$\frac{}{\textbf{const}\ x = n_1;\ e_2 \longrightarrow e_2[n_1/x]} \qquad \frac{e_1 \longrightarrow e_1'}{\textbf{const}\ x = e_1;\ e_2 \longrightarrow \textbf{const}\ x = e_1';\ e_2}$$

2. **Big-Step Environment Semantics: Dynamic Scoping**. Consider the following JAVASCRIPT expression:

$$\textbf{const}\ \mathtt{x} = 1;\ \textbf{const}\ \mathtt{g} = (\textbf{function}\ (\mathtt{y})\ \mathtt{x});\ \big(\textbf{function}\ (\mathtt{x})\ \mathtt{g}(2)\big)(3)$$

Recall that **function** $p(x)\ e$ is an abbreviation for **function** $p(x)\ \{\ \textbf{return}\ e\ \}$.

(a) $\boxed{\text{4 points}}$ What is the value of this expression under static scoping? Under dynamic scoping? Hint: this example expression, should look very familiar.

Solution: Static scoping: 1. Dynamic scoping: 3.

(b) $\boxed{\text{4 points}}$ Recall that a derivation of a judgment is an application of inference rules arranged in a tree. Using the big-step operational semantics with dynamic scoping as in Lab 3 (see Figure 1), let us consider a derivation that specifies the value of the above expression. After the evaluation of the two top-level **const** bindings, there is one sub-derivation of the following form:

$$\frac{\mathscr{D}_1 \quad \mathscr{D}_2 \quad \mathscr{D}_3}{E_1 \vdash \big(\textbf{function}\ (\mathtt{x})\ \mathtt{g}(2)\big)(3) \Downarrow \boxed{}}\ \textsc{Call}$$

where $E_1 \stackrel{\text{def}}{=} \cdot[\mathtt{x} \mapsto 1][g \mapsto \textbf{function}\ (\mathtt{y})\ \mathtt{x}]$ and with three sub-derivations $\mathscr{D}_1$, $\mathscr{D}_2$, and $\mathscr{D}_3$. Derivations $\mathscr{D}_1$ and $\mathscr{D}_2$ are given as follows:

$$\mathscr{D}_1 \stackrel{\text{def}}{=} \frac{}{E_1 \vdash \textbf{function}\ (\mathtt{x})\ \mathtt{g}(2) \Downarrow \textbf{function}\ (\mathtt{x})\ \mathtt{g}(2)}\ \textsc{Val} \qquad \mathscr{D}_2 \stackrel{\text{def}}{=} \frac{}{E_1 \vdash 3 \Downarrow 3}\ \textsc{Val}$$

Complete the derivation above by filling in the box above for the value of the expression and give the sub-derivation $\mathscr{D}_3$ below. Hint: sub-derivation $\mathscr{D}_3$ consists of applications of 4 inference rules.

Solution: The box gets filled in with the value 3. The sub-derivation $\mathscr{D}_3$ is as follows:

$$\frac{\dfrac{v_1 = \textbf{function}\ (\mathtt{y})\ \mathtt{x}}{E_1[\mathtt{x} \mapsto 3] \vdash \mathtt{g} \Downarrow v_1}\ \textsc{Var} \quad \dfrac{}{E_1[\mathtt{x} \mapsto 3] \vdash 2 \Downarrow 2}\ \textsc{Val} \quad \dfrac{}{E_1[\mathtt{x} \mapsto 3][\mathtt{y} \mapsto 2] \vdash \mathtt{x} \Downarrow 3}\ \textsc{Var}}{E_1[\mathtt{x} \mapsto 3] \vdash \mathtt{g}(2) \Downarrow 3}\ \textsc{Call}$$

Name: ____________________

$$\boxed{E \vdash e \Downarrow v}$$

$$\textsc{Var}\ \frac{}{E \vdash x \Downarrow E(x)} \qquad \textsc{Val}\ \frac{}{E \vdash v \Downarrow v} \qquad \textsc{Call}\ \frac{E \vdash e_1 \Downarrow \textbf{function}\ (x)\ e' \qquad E \vdash e_2 \Downarrow v_2 \qquad E[x \mapsto v_2] \vdash e' \Downarrow v'}{E \vdash e_1(e_2) \Downarrow v'}$$

Figure 1: Big-step operational semantics with dynamic scoping for JAVASCRIPTY (select rules).