Exercise 1: Javascript

Given the following Javascript code snippet:

```javascript
s = "some random string";
s.x = 42;
s.x;
```

1. Use the JavaScript shell from [http://www.squarefree.com/shell/shell.html](http://www.squarefree.com/shell/shell.html) to execute the above Javascript code. Which results do you get?
2. Change the first or second line of the example, such that executing the third line (s.x;) prints 42.
3. Explain the behavior you observe. What would you suggest to prevent such mysterious bugs from happening?

Exercise 2: Types and Evaluation for JAUS

1. Which of the following JAUS expressions are type correct? Give a type derivation for all type correct expressions. Assume that variable `x` is of type `int` and variable `y` is of type `boolean`.
   - 1 + true
   - 23 + (47 - 11)
   - !(false)
   - y + x
   - !y

2. Evaluate the following JAUS expressions as far as possible. Which of the resulting expressions are values?
   - 23 + (47 - 11)
   - (1 + 1) + true
Exercise 3: Conditional Expression

Consider the language JAUS of Java expressions.

\[
\begin{align*}
\text{Variables} & \ni x ::= \ldots \\
\text{Numbers} & \ni n ::= 0 \mid 1 \mid \ldots \\
\text{Truth Values} & \ni b ::= \text{true} \mid \text{false} \\
\text{Expressions} & \ni e ::= x \mid n \mid b \mid e + e \mid e
\end{align*}
\]

Extend the language with a conditional expression as known from Java, C or C++.

\[
\begin{align*}
\text{Types} & \ni t ::= \text{int} \mid \text{boolean} \\
\text{Expressions} & \ni e ::= \ldots \mid (e ? e : e)
\end{align*}
\]

\textit{Judgment} \vdash e : t

1. Define a typing rule (COND) for the conditional.

\[
\begin{array}{c}
\text{(COND)} \\
\vdash e_0 : \\
\vdash e_1 : \\
\vdash e_2 : \\
\hline
\vdash (e_0 ? e_1 : e_2) :
\end{array}
\]

2. Define evaluation rules for the conditional.

\[
\begin{align*}
\text{Values} & \ni v ::= n \mid b \\
\text{Evaluation} & e \rightarrow e' \\
\text{(E-COND1)} & (e_0 ? e_1 : e_2) \rightarrow \\
\text{(E-COND2)} & (v_0 ? e_1 : e_2) \rightarrow \\
\text{(E-COND3)} & (v_0 ? e_1 : e_2) \rightarrow
\end{align*}
\]

3. Extend the preservation proof from the lecture by treating one case for the conditional.

Exercise 4: Subtyping

Suppose we extend JAUS to distinguish the integer types \texttt{byte}, \texttt{short}, and \texttt{int}, where \texttt{byte} <: \texttt{short} and \texttt{short} <: \texttt{int} (subtypes as in Java) and the respective typing rules for constants:

\[
\begin{align*}
\text{(BYTE)} & \quad -128 \leq n \leq 127 \quad \vdash n : \texttt{byte} \\
\text{(SHORT)} & \quad -32768 \leq n \leq 32767 \quad \vdash n : \texttt{short}
\end{align*}
\]
Argue why each of the following typing rules is sound. Provide a counterexample if a rule is unsound.

1. 
   \[ \text{(MUL)} \]
   \[ \vdash e_1 : \text{int} \quad \vdash e_2 : \text{int} \]
   \[ \vdash e_1 \cdot e_2 : \text{int} \]

2. 
   \[ \text{(MUL2)} \]
   \[ \vdash e_1 : \text{byte} \quad \vdash e_2 : \text{byte} \]
   \[ \vdash e_1 \cdot e_2 : \text{short} \]

3. 
   \[ \text{(DIV)} \]
   \[ \vdash e_1 : \text{byte} \quad \vdash e_2 : \text{byte} \]
   \[ \vdash e_1 / e_2 : \text{byte} \]