Semantics of Programming Languages

(a) The integer expressions $e$ of a C-like language take the form $e ::= n \mid x \mid x++ \mid ++x \mid e + e$, where $n$ ranges over integer constants and $x$ over integer storage variables. The expression $x++$ returns the value stored in the integer variable $x$ and then increments the stored value by one; whereas $;++x$ first increments the stored value by one and then returns it. Assuming a left-to-right evaluation order, give an operational semantics for all these expressions, in the form of an evaluation relation $\langle s, e \rangle \downarrow \langle s', n \rangle$, where $s, s'$ range over states which are finite functions from integer storage variables to integers. [5 marks]

(b) The commands (statements) $c$ of this same language take the form $c ::= x = e \mid x += e \mid c; c$. The first form is assignment and the last is sequencing; the command $x += e$ evaluates $e$, adds the result to the value stored in $x$ and stores the result there. Give an operational semantics for these commands in the form of an evaluation relation $\langle s, c \rangle \downarrow s'$ (where $s, s'$ are as above). [4 marks]

(c) Define the notion of semantic equivalence for these expressions and commands. [3 marks]

(d) For each of the following pairs of expressions or commands, state, with justification, whether or not they are semantically equivalent.

(i) $x++ + 1$ and $;++x$ [2 marks]
(ii) $x = ++x$ and $x = x++$ [2 marks]
(iii) $x = ++x$ and $x += 1$ [2 marks]
(iv) $x += e$ and $x = x + e$, for any $e$ [2 marks]