CS383 Quiz 1

Solution

1.

Which one is not a basic property of programming languages?

a. Functions

b. Syntax

c. Type

d. Semantics

2.
For C, which one does not constitute a scope?

a. Compilation unit

b. Function

c. For loop

d. Block

3.

Which one of the following is NOT a part of doing an inductive proof?

a. Clearly state the induction hypothesis.

b. Make a proper inductive definition.

c. Clearly state what you are doing induction on.

d. Show one case for each rule in the inductive definition.

4.

If the structure of your induction hypothesis is "If X and Y then A", which of the following things is proper for you to assume and prove?

a. Assume X or Y, prove A

b. Assume X and Y, prove A

c. Assume X prove A, or Assume Y prove A

d. Assume X prove A, and Assume Y prove A

5.
If the structure of your induction hypothesis is "If X or Y then A", which of the following things is proper for you to assume and prove?

a. Assume X or Y, prove A

b. Assume X and Y, prove A

c. Assume X prove A, or Assume Y prove A

d. Assume X prove A, and Assume Y prove A

6.

Which one of the following is not essential in lambda calculus?

a. Variable: x

b. Conditional: if e1 then e2 else e3

c. Abstraction: λ x . e

d. Application: e1 e2

7.

Which one of the following is incorrect about the semantics of if-then-else?

a. $$\frac{e_1 \rightarrow e_1' \quad e_2 \rightarrow e_2' \quad e_3 \rightarrow e_3'}{\text{if } e_1 \text{ then } e_2 \text{ else } e_3 \rightarrow \text{if } e_1' \text{ then } e_2' \text{ else } e_3'}$$

b. $$\frac{e_1 \rightarrow e_1'}{\text{if } e_1 \text{ then } e_2 \text{ else } e_3 \rightarrow \text{if } e_1' \text{ then } e_2 \text{ else } e_3}$$

c. $$\frac{}{\text{if } \mathit{true} \text{ then } e_2 \text{ else } e_3 \rightarrow e_2}$$

d. $$\frac{}{\text{if } \mathit{false} \text{ then } e_2 \text{ else } e_3 \rightarrow e_3}$$

8.

Which one of the following is different from the other three after evaluation?

a. $\lambda x.\, x\ x$

b. $(\lambda x.\, x\ x)\ (\lambda y.\, y\ y)$

c. $(\lambda y.\, \lambda x.\, y\ x)\ (\lambda x.\, x\ x)\ (\lambda y.\, y\ y)$

d. $(\lambda x.\, (\lambda x.\, x\ x)\ x)\ (\lambda x.\, x\ x)$

9.

Which one of the following is incorrect in simply typed lambda calculus?

a. $(\lambda f{:}int \to int.\ \lambda g{:}int.\ f\ g)\ (\lambda x{:}int.\ x+5)$

b. $(\lambda f{:}bool.\ \lambda g{:}bool \to int.\ g\ f)$

c. $(\lambda f{:}int \to int.\ \lambda g{:}int.\ f\ g)\ (\lambda x{:}int \to int.\ x)$

d. $(\lambda f{:}int.\ f)\ ((\lambda x{:}int.\ x)\ 5)$

10.
Which one of the following is incorrect?

a. Type checking is the process of verifying type safety of a program (or a term).

b. If a well-typed term has type t, its evaluation result (if any) also has type t.

c. Well-typed programs do not get stuck.

d. Typed lambda calculus is more expressive than untyped lambda calculus.

#	A	B	C	D	A%	B%	C%	D%
1	22	1	0	3	**85**	**4**	**0**	**12**
2	13	1	11	1	**50**	**4**	**42**	**4**
3	1	20	4	1	**4**	**77**	**15**	**4**
4	0	25	1	0	**0**	**96**	**4**	**0**
5	0	0	10	16	**0**	**0**	**38**	**62**
6	0	24	1	1	**0**	**92**	**4**	**4**
7	19	4	3	0	**73**	**15**	**12**	**0**
8	17	0	6	3	**65**	**0**	**23**	**12**
9	3	12	10	1	**12**	**46**	**38**	**4**
10	2	6	13	5	**8**	**23**	**50**	**19**