OHJ-2056 Principles of Programming Languages 5 cr 22–May–2009

Write onto each paper your name, student number, and email address. Calculators, notes, books, etc. are not allowed in the examination. The examination was written by Antti Valmari.

1 (a) List the three programming languages that you know best.

(b) Let $\oplus$ and $\otimes$ represent addition and multiplication. Evaluate $1 \oplus 2 \otimes 3 \oplus 4$ assuming that each operator is right associative, and $\oplus$ is on a higher precedence level than $\otimes$, that is, binds stronger than $\otimes$.

(c) Like (b), except that $\oplus$ is on the same precedence level with $\otimes$.

(d) Like (b), except that $\oplus$ on a lower precedence level than $\otimes$.

(e) Write an expression that is as simple as possible and contains a unary minus.

(f) Below is a C statement that looks unusual. Draw vertical lines that divide it to lexemes.

```
x=!x!=x;
```

2 Explain briefly ($\leq$ 15 words) the following concepts:

(a) The scope of a variable
(b) The lifetime of a variable
(c) Static typing
(d) Dynamic typing
(e) Coercion
(f) Primitive type, i.e., scalar type

3 What do the following mean: static array, stack-dynamic array, and explicit heap-dynamic array? Mention one disadvantage of static arrays compared to the other two groups. To which of these groups the C++ `vector` best fits, and how can one reason that?

4 (a) Write a subroutine (i.e., procedure) and its call so that if the parameters are passed with the value–result method, an output statement within the subroutine prints a different value from what it would print if the parameters were passed by reference.

(b) What good and/or bad features the value–result method has compared to pass by reference? Altogether two features suffice.

(c) How are actual parameters bound to formal parameters when keyword parameters are used? What is the name of the alternative way of binding actual parameters to formal parameters?

5 (a) Draw pictures that represent the following LISP lists: `(1 2 3)`, `((1) (2 3))` and `(())`.

(b) What does the following LISP function do?

```
(defun f (x y)
  (cond
    ( (null x) '(y) )
    ( T (cons (car x) (f (cdr x) y) ) )
  )
)
```

(c) Write a LISP function `g` that reverses a list, e.g., `(g (1 2 3))` returns `(3 2 1)`.

the end