OHJ-2050 Ohjelmointikielten periaatteet 27.5.2010

Tentissä ei saa olla laskimia, muistiinpanoja, kirjoja yms. Tentin laati Antti Valmari.

1 (a) Kirjoita jokaiseen vastauspaperiin nimesi, opiskelijanumerosi ja sähköpostiosoitteesi niin, että niistä saa selvän.

Selitä lyhyesti seuraavat käsitteet:

(b) Pinodynaaminen muuttuja.

(c) Koottu lause.

(d) (Eksplisiittinen) tyyppimuunnos.

(e) Osavälityyppi.

(f) Assosiatiivinen taulukko.

2 Mitä alla olevat C++- ja Pascal-koodit tulostavat?

(a)
```
k = 2; for( i = 0; i < k; ++i ){ if( i < 2 ){ ++k; }} std::cout << k;
```

(b)
```
k := 2; for i := 0 to k-1 do begin if i < 2 then k := k+1 end write(k);
```

(c) Pascal-ohjelma muunnettiin C++-kielelle. C++-ohjelma oli paljon hitaampi kuin alkuperäinen Pascal-ohjelma. Syy löytyi seuraavalta riviltä. Selitä syy.
```
for( i = 0; i < foo(A); ++i ){ ... }
```

(def) Esittele kaksi erilaista Pascalin case- tai C:n switch-lauseen toteutusta ja kerro, missä tilanteessa kumpikin toteutus on toista parempi.

3 Uusi ohjelmointikieli näyttää C:ltä, mutta sillä on tuntematon parametrinvälitysmekanismi `%`. Se on jokin kurssillä käsitellyistä. Kirjoita `int main(){ ... }` ja `void test( int % n ){ ... }` siten, että ne tulostavat mekanismin `%` nimen. Globaaleja muuttujia saa käyttää. Kerro, kuinka ohjelma toimii. Kolme mekanismia riittää maksimipisteisiin.

4 Luokat *B* ja *C* perivät luokan *A*. Luokka *D* perii *B*:n ja *C*:n. *A*:lla on instanssimuuttuja (eli tietojäsen) *ia*. Samalla lailla *B*:llä on *bi*, *C*:llä on *ci* ja *D*:llä on *di*.

(a) Osoittimet *pa*, *pb*, *pc* ja *pd* osoittavat samaan *D*-luokan olioon. *pa*:n tyyppi on osoitin *A*:han, *pb* on osoitin *B*:hen, jne. Piirrä olion muisti sisältäen *ia*, *ib*, *ic* ja *id*, ja piirrä osoittimet. Käytä monistettua moniperintää.

(b) Toista (a) käyttäen jaettua moniperintää.

(c) *A*:lla on instanssimetodi (eli funktio) *f*, joka määritellään *b*:ssä uudelleen. Mikä kielensuunnitteluongelma nousee kun *f*:ää kutsutaan *pd*:n kautta? Esittele kaksi mahdollista ratkaisua.

5 (a) Mitä seuraavat LISP-funktiokutsut tekevät?

(a) `(a nil)`

(b) `(a '3)`

(c) `(a '(3,5,2))`

(d) `(a '(3,(5,2),nil,7,1))`

```
(defun a (x)
  (cond
    ( (null x) '0 )
    ( (null (cdr x) ) nil )
    ( t (cons (car x) (a (cdr x) )) )
) )
```

(ef) Kirjoita LISP-funktio `b` siten, että `(b x y)` palauttaa listan `x`, josta on poistettu kaikki `y`:n esiintymät. Esim. `(b '(5,3,4,3,3,2) '3)` palauttaa `(5,4,2)`.

Loppu (Samat kysymykset ovat paperin toisella puolella englanniksi.)

OHJ-2056 Principles of Programming Languages 27–May–2010

Calculators, notes, books, etc. are not allowed in the exam. The exam was written by A. Valmari.

1 (a) Write clearly readably onto each paper your name, student number, and email address.

Explain briefly the following concepts:

(b) Stack-dynamic variable.

(c) Compound statement.

(d) (Explicit) type cast.

(e) Subrange type.

(f) Associative array.

2 What do the C++ and Pascal codes below print?

(a)
```
k = 2; for( i = 0; i < k; ++i ){ if( i < 2 ){ ++k; }} std::cout << k;
```

(b)
```
k := 2; for i := 0 to k-1 do begin if i < 2 then k := k+1 end write(k);
```

(c) A Pascal program was converted to C++. The C++ program was much slower than the original Pascal program. The reason was found on the following line. Explain the reason.
```
for( i = 0; i < foo(A); ++i ){ ... }
```

(def) Describe two different implementations of Pascal case or C switch statement and tell in which situation each implementation is better than the other.

3 A new programming language looks a lot like C, but has an unknown parameter passing mechanism `%`. It is one of those discussed in the course. Write `int main(){ ... }` and `void test( int % n ){ ... }` so that they print the name of the mechanism `%`. Global variables can be used. Tell how the program works. Three mechanisms suffice for maximum points.

4 Classes *B* and *C* inherit class *A*. Class *D* inherits *B* and *C*. *A* has an instance variable (i.e., data member) *ia*. Similarly *B* has *bi*, *C* has *ci*, and *D* has *di*.

(a) An object of class *D* is pointed to by the pointers *pa*, *pb*, *pc*, and *pd*. *pa* is of type pointer to *A*, *pb* is a pointer to *B*, and so on. Draw the memory of the object showing *ia*, *ib*, *ic*, and *id*, and draw the pointers. Use replicated multiple inheritance.

(b) Repeat (a) using shared multiple inheritance.

(c) *A* has an instance method (i.e., function) *f* that is redefined in *b*. What language design problem arises when *f* is called via *pd*? Describe two possible solutions.

5 (a) What do the following LISP function calls do?

(a) `(a nil)`

(b) `(a '3)`

(c) `(a '(3,5,2))`

(d) `(a '(3,(5,2),nil,7,1))`

```
(defun a (x)
  (cond
    ( (null x) '0 )
    ( (null (cdr x) ) nil )
    ( t (cons (car x) (a (cdr x) )) )
) )
```

(ef) Write a LISP function `b` such that `(b x y)` returns list `x` with all instances of `y` removed. For instance, `(b '(5,3,4,3,3,2) '3)` returns `(5,4,2)`.

The end (The same questions are in Finnish on the other side of the paper.)