INF3110/4110

Problems week 42 (16.10 - 20.10, 2006)

Problem 1
Sketch the code needed for updating the context vector when entering a block/calling a function and when exiting a block/function call.

Problem 2

```
begin
  integer procedure Jensen(x, i, n);
    name x, i, n; integer x, i, n;
  begin
    integer index, sum;
    for index := 1 step 1 until n do
      begin
        i := index;
        sum := sum + x;
     end;
    Jensen := sum;
  end Jensen;
  integer ix, res1, res2, res3; integer array a(1:5);
  a(1):= 7; a(2):= -1; a(3):= 11; a(4):= 8; a(5):= 4;

  res1:= Jensen(ix*ix, ix, 10);
  res2:= Jensen(a(ix), ix, 5);
  res3:= Jensen(if Rem(a(ix),2)<>0 then 1 else 0, ix, 5);
end
```

What are evaluated by the three calls of Jensen? Rem(x,y) is the remainder of x divided by y.

Problem 3
In the following program sketch the procedure b has a function f as formal parameter. In main it is called with the procedure a as actual parameter.

```
int i, j;
a(){
  int k; …
};
b(function f){
  int i, j, k;
  c(){
    … k = …
   };
  f();
  b(c);
  …
}
main(){
  b(a);
};
```

Describe each stage in the life of the run-time stack as the result of calling b(a).

Problem 4
Exercise 7.12 in Mitchell

Problem 5
Can the L-value of a variable be accessed only when its name is visible (i.e. within scope)? Why or why not.

Problem 6
Some language designer like the so-called Correspondence Principle: For each form of declaration there exists a corresponding parameter mechanism.

a) Give examples showing that the converse is not always the case.
b) Give examples where this principle has been followed, e.g. in C, C++ Java (we will take ML later)
c) Would it make sense to apply the principle to declaration of literals? To types?

Problem 7
Slide 21 of the 16.10-‘Lysark’ indicates a way to have functions as parameters in Java. Is it possible to pass static methods in this way, and if not, in another way?

Problem 8
Parameters to procedures are often used in order to parameterize the computation, so that procedures called with different actual values perform different computations.

In which cases will a parameterless procedure not perform the same computation every time it is called?

Problem 9
By-reference and by-value-result have almost the same effect. Make a small example where the same call behaves differently.

Problem 10
Give one good reason (based on how we would like memory to be organized as a stack) for classes not having function parameters. Illustrate the problem by sketching an example.