COMP 251: 2000 Fall Semester Final

Date:	Monday, Dec 18, 2000
Time Allowed:	3 hours
Instructions:	(a) There are 7 questions in this paper.
	(b) Attempt ALL questions.
	(c) Write ALL answers in the blue book provided.
	(d) The total mark is 100.

Problem 1 [15%] Concepts.

Give short answer to each of the following questions.

(a) [3%] Consider the following codes:

```
class Complex { ... }
Complex x;
```

The above codes can be used in both C++ and Java to define an object variable "x", but the variable actually represents different things in the two languages. What are they?

(b) [3%] Explain the use of "cut" operator, "!", in PROLOG in words and with a simple example of not more than 5 lines.

(c) [3%] What is side effect? How does SML avoid side effect?

(d) [3%] What is a structured program?

(e) [3%] What is an array?

Problem 2 [10%] Context Free Grammar.

Consider the following CFG in BNF notation:

```
<Expr> ::= <Expr> + <Term> | <Expr> - <Term> | <Term>
<Term> ::= (<Expr>) | <Id>
<Id>   ::= a | b | c
```

Add the following **two** constructs to the grammar. The result should still be an unambiguous CFG.

(a) [7%] the exponentiation operator "^". The operator has higher precedence than "+" and "−", and is right-associative. e.g. the following is a valid expression with the exponentiation operator:

```
a+b^c^(a-b) .
```

(b) [3%] the function call. A function call consists of a function identifier and an expression in bracket. e.g. the following is a valid expression with a function call: `a+ b(a-b)^c`.

Problem 3 [10%] Parameter Passing.

Consider the following Pascal-like program where lexical scope rule is used for all variable bindings:

```
program Main();
    var Y: integer;

    procedure P(X: integer)
        begin X := X+1; write(X); write(Y) end;

    Y := 1; P(Y); write(Y)
end.
```

Give the numbers printed for each of the following cases when parameter Y is transmitted to procedure P by (a) value (b) reference (c) value-result (d) name. You may assume that a space is inserted after each printed number.

Problem 4 [10%] SML Programming.

You are given the following SML value and functions:

```
val X = [[1,2], [3,4], [5,6], [7,8], [9,10]];

fun append [] x = x
  | append (x::xs) y = x::(append xs y);

fun foldL g a [] = a
  | foldL g a (head::tail) = foldL g (g a head) tail;

fun foldR g a [] = a
  | foldR g a (head::tail) = g head (foldR g a tail);
```

(a) [4%] What are the types of the <u>two</u> functions: foldL and foldR?

(b) [1%] How would you use functions “append” and “foldL” to collapse the list of lists, X, into one list [1,2,...,10]?

(c) [1%] How would you use functions “append” and “foldR” to collapse the list of lists, X, into one list [1,2,...,10]?

(d) [4%] From your answers to (b) and (c), determine which of "foldL" or "foldR" is more efficient in collapsing a list of lists in general? You **must** give reason(s) and no marks will be given for a simple "yes/no" answer.

Problem 5 [10%] PROLOG Programming.

Give an example PROLOG program and a query that will never return, i.e. run into an infinite loop. Please explain why it will run into an infinite loop by drawing a search tree for the query.

Problem 6 [10%] Java Programming.

Two arrays a and b of objects are equal if they have the same length, and for each i such that $0 \leq i < a.length$, $a[i] == b[i]$.

(a) [7%] Write a Java class called `ArrayTest` such that:

- It has a method called `arrayEqual` that can be used to check whether any two arrays of objects are equal or not.
- It has a `main()` method that creates an array of A objects of length 3, and an array of B objects of length 3, and call `arrayEqual` to compare whether these two arrays are equal. The two classes A and B are as follows:

```
class A {
    private int number;
    private String name;
    A(int x, String y) { number = x; name = y; }
}

class B {
    private int number;
    private String name;
    B(int x, String y) { number = x; name = y; }
}
```

(b) [3%] Can you create two arrays, one an array of A objects, and the other an array of B objects so that these two arrays are equal according to the definition given above? If your answer is yes, then give an example. If your answer is no, explain why.

Problem 7 [35%] Comparative Programming.

An integer tree is a tree whose leaves are labelled by integers. For instance, the following is an integer tree:

```
       N0
      /  \
     N1   N2
   / | \  |
  1  N3 3 5
     |
     6
```

where N0, N1, N2, and N3 are labels for internal nodes. Notice that a node in such a tree can has zero, one, or more children. Given an integer tree, the value of this tree is the sum of its leaves. For instance, for the above tree, its value is $1 + 6 + 3 + 5$, which is 15. For **<u>EACH of the 3 languages</u>** that you have learnt: ML, PROLOG, and Java, decide on a data structure to represent integer trees and write a program to compute the value of an integer tree, and show how to call your program to compute the value of the above integer tree. More specifically

ML: In the case of ML, you need to define a new data type called `IntTree`, and a function called `evalIntTree` of the type: `fn: IntTree -> Int`. You can define any helper functions, global or local, that you need. Here are some examples of how to create some new datatypes in ML:

```
datatype IntList = mylist of int list;
```

This will create a new type called IntList. Values of this new type have the form: `intlist(x)`, where x is an int list such as `[1,2,3]`.

```
datatype Expr = sum of Expr * Expr
              | label of int;
```

This creates a new datatype called `Expr`. Notice that `Expr` is recursively defined, and each value in this new datatype is either of the form `label(x)`, where x is an integer, or of the form `sum(e1,e2)`, where `e1` and `e2` are values of type `Expr`.

PROLOG: In the case of PROLOG, you need to tell us in English how the trees are represented as PROLOG terms, and give as an example, a term that represents the above example integer tree. You also need to write a program that specifies the relation: `evalIntTree(Tree,V)` such that V is the value of `Tree` if $Tree$ represents a valid integer tree; otherwise, V is the atom 'Wrong Input'. Again, you are free to define any other relations. Remember that to compute the sum of X and Y, use: `Z is X + Y`.

Java: In the case of Java, you need to define a class called `IntTree` that has a method called `evalIntTree` that takes an `IntTree` object and returns an integer that is the value of the `IntTree` object. You also need to define a class called `IntTreeTest` that has a `main()` function that will construct the above example integer tree and print out its value.