Comp Sci 1MD3
Mid-Term I 2004

Dr. Jacques Carette

Name: ______________________________

Student No.: ___________________________
Duration : 50 minutes

- This midterm contains 18 questions on 8 pages.
- This midterm will be marked out of 50. There are 55 total marks available.
- Answer the questions in the space provided.
- Do not separate the pages.
- Make sure that you do not get stuck on one question; use your time wisely.
- You should spend roughly 1 minute per mark for each question.

1. Translate the following Pascal code fragment to C. *[5]*

```
const Len = 10;

var J,N: Integer;
    L: Array[1..Len] of Integer;

begin
    for J := 1 to Len do
        L[J] := J*J;

    N := 1;
    repeat
        begin
            writeln(L[N]);
            N := N + 1;
        end;
    until N > Len;
end.
```

2. What is the main innovation of 3rd generation programming languages? *[1]*

3. What is the difference between a declarative statement and an imperative statement (in languages like C and Pascal)? *[2]*

4. A for loop can always be rewritten as a semantically equivalent while loop. Demonstrate this. Extra credit *[2]*: show that it can be done with a do-while loop as well. *[2]*

5. What is the difference between an object and a class? *[2]*

6. What is a constructor? *[1]*

7. What is the run-time of these code fragments, as a function of n? Assume the proper declarations have been made and that the code is correct. Use $\Theta()$ notation for your answers: *[7]*

   (a)
   ```
   for (i=0; i<n*n; i++) {
       for (j=0; j<n; j++) {
           a[i][j] = i+j;
       }
   }
   ```

   (b)
   ```
   for (i=0; i<10; i++) {
       for (j=0; j<=n; ++j) {
           a[i][j] = i-j;
       }
   }
   ```

   (c)
   ```
   i=0;
   while (i*i < n) {
       for (j=0; j<n; j++) {
           a[i][j] = i+j;
       }
       j=0;
   while (j*j < n*n*n) {
           a[i][j] += a[j][i];
       }
   i++;
   }
   ```

8. Of the above 3 pieces of code, which will (asymptotically) be the fastest? The slowest? *[2]*

9. Name the programming paradigm of each the following languages (1 point each): *[4]*

   (a) C$^{\#}$

   (b) Pascal

   (c) Haskell

   (d) Prolog

10. Which of the following roughly corresponds to *data*, *type* and *function* definitions. *[3]*

    (a) DTD

    (b) XSL

    (c) XML

11. Assume that the following Maple statements have already been executed

```
L := [7,6,5,4,3,2,1];
g := y -> (x->x+y);
h := [[3], [3, 9], [3, 9, 27], [3, 9, 27, 81], [3, 9, 27, 81, 243]];
```

What is the result of executing the following Maple statements (1 point each): *[3]*

(a) map($x \rightarrow 10 - x$,L);

(b) map(g(z), L);

(c) map($c \rightarrow \text{add}(i, i = c)$,h);

12. (Haskell) Assume that the following has already been executed:

```
let l = [12, 1, -5, 3]
let apply f x = f x
```

Give the result of executing the following Haskell expressions: (1 point each) *[3]*

(a) "Madam" == reverse "Madam"

(b) map (5*) l

(c) (flip apply) "hello" reverse

13. Give 3 (of the 8) rules of good design for a GUI. *[3]*

14. Consider the following C$^{\#}$ code.

```
using System;
class Squares1 {
   static void Main() {
      for (int i=1; i<=10; i++) {
         Console.WriteLine("{0} ", i*i);
      }
   }
}
```

It prints the squares of the integers 1-10, all on the same line, separated by one space. Change it to print the cube of the integers 5-12, all on the same line, separated by a comma, in reverse order. Ending with a comma is fine. *[3]*

15. Consider the following Perl program.

```
for (1..10) { print $_**2," "};
```

It prints the squares of the integers 1-10, all on the same line, separated by one space. Change it to print the cube of the integers 5-21, all on the same line, separated by commas. Ending with a comma is fine. *[3]*

16. Consider the following embedded Javascript program.

```
<html>
  <head>
    <title>Javascript Squares</title>
  </head>
  <body>
    <script>
      for (var i = 1; i <= 10; ++i) {
        document.write( Math.pow(i, 2) + "<br>" );
      }
    </script>
  </body>
</html>
```

The resulting document contains the squares of the integers 1-10, on different lines. Change the script to print the cube of the integers 5-12, all on the same line, separated by a comma and a space, in reverse order. Ending wiht a comma is fine. *[3]*

17. Consider the following Common Lisp program.

```
(dotimes (i 10)
  (format t "~D " (* i i)))
```

It prints the squares of the integers 1-10, all on the same line, separated by one space. Change it to print the cube of the integers 1-21, all on the same line, separated by one space. *[3]*

18. BONUS question. What does the following (valid, if not very nice) C code do:

```
main(a){printf(a="main(a){printf(a=%c%s%c,34,a,34);}",34,a,34);}
```

Be very precise! *[3]*

END TEST