Comp Sci 1MD3
Mid-Term I 2003

Dr. Jacques Carette

Name: ______________________________

Student No.: ____________________________

Duration : 50 minutes

- This midterm contains 18 questions on 3 double-sided pages (including this one).
- This midterm will be marked out of 50. There are 55 total marks available.
- Answer the question in the space provided.
- Do not separate the pages.
- Make sure that you do not get stuck on one question; use your time wisely.

---

1. Precisely define “Algorithm”. Give one consequence of the definition. *[3]*

2. What “grouping” do the following algorithms belong to? *[3]*

   (a) quicksort

   (b) shortest path in a graph (with positive weights)

   (c) multiplication of n matrices of uneven sizes

3. How is the problem of ambiguity of algorithm representation usually solved? *[2]*

4. Give an example (by name) of an algorithm in each of the following classes: $\Theta(\lg n)$, $\Theta(n)$, $\Theta(n^2)$. *[3]*

5. This question concerns the following pseudo-code:

```
z← 0;
x← 1;
while (x < 6) do
    (z ← z+x;
     x ← x+1)
```

   (a) convert the pseudo-code to a C code fragment which uses the same constructs. No need to declare your variables. *[2]*

(b) convert the pseudo-code to a C code fragment that uses a *for* loop. No need to declare your variables. *[2]*

6. Design an algorithm that, given two strings of characters, tests whether the first string appears somewhere in the second. You may assume that neither strings are empty. Write the algorithm in C, filling-in the code below:

```
bool contains(char *s1, char *s2) {









}
```

   You may use the C library function *strlen* in your implementation. *[5]*

7. What is the difference between an assembler and a compiler? *[2]*

8. What is the difference between a declarative statement and an imperative statement? *[2]*

9. Give 3 different, syntactically correct examples of control structures in C. *[3]*

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

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

12. Name the 4 programming paradigms. Pick 2 and describe them. *[4]*

13. Define *encapsulation* in the context of imperative programming languages. Define *encapsulation* in the context of object-oriented programming languages. Does the assigned meaning match the structuring methodology of each paradigm? Explain. *[6]*

14. 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: *[6]*

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

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

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

15. Given the following XML:

```
<courses>
      <course>
            <name>CS 1MD3</name>
      </course>
      <course>
            <attendance>84</attendance>
            <course>
                  <name>SE 3M04</name>
            </course>
      </course>
</courses>
```

   (a) Is it well-formed ? *[1].*

   (b) Is it valid ? *[1].*

16. Where would you find ASTs and parse trees? *[1]*

17. SVG and postscript use the same technique to represent, respectively, 2D graphics and printer instructions. What technique do they use? *[2]*

18. Rewrite the following C fragment to use a *case* statement. *[4]*

```
extern int f1(int);
extern int f2(int);
extern int f3(int);
extern int f4(int);

typedef int (*pifi)(int);

static pifi tab[7] = {f2, f3, f1, f1, f1, f4, f2};

int g(int i) {
    return tab[(i*i)%7](i);
}
```

END TEST