

Comp Sci 1MD3
Mid-Term I 2005
Dr. Jacques Carette

Name: _____

Student No.: _____

Duration : 50 minutes

-
- This midterm contains 17 questions on 7 pages.
 - This midterm will be marked out of 50. There are 54 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. First, translate the following Pascal code fragment to C, keeping the structure as similar as possible. Then write the smallest C program you can which does the same thing. [6]

```
program funny;

const len = 10;

var i,j: Integer;
    L: Array[1..len] of Integer;

begin
  for i := 1 to len do
    begin
      L[i] := len-i;
    end;

    j := len;
  repeat
    begin
      writeln(L[j]);
      j := j - 1;
    end;
  until j = 0;
end.
```

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

3. What is the difference between machine language and assembly language? Illustrate your answer with an example. *[3]*

4. Explain the difference between immediate and direct mode addressing. *[2]*

5. Using an analogy with concepts in imperative programming languages, what is the relation between an object and a class? *[2]*

6. Answer each of the following with true/false: *[5]*
 - (a) Java bytecode is machine specific
 - (b) A Virtual Machine (VM) is like an abstract CPU
 - (c) Unicode representations take more space than ASCII representations for characters
 - (d) Real numbers and floating point numbers are equivalent
 - (e) A *value* parameter is passed via a pointer

7. Give the syntax for **sequence** and **selection** in C. [2]

8. Is HTML a programming language? Justify your answer. [2]

9. Give an example of a semantic tag and a presentation tag in HTML. [2]

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

(a) LISP (or Scheme)

(b) Smalltalk

(c) Fortran

(d) Java

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

12. Explain 2 examples of metaphors used in GUIs, giving examples of their use. *[4]*

13. Write down a valid Python list which could not be stored (directly) in a C array. Explain the difference in terms of the typing mechanisms of each language. *[4]*

14. Write down the value of the variable `a` after executing the following piece of Python code:

[2]

```
l = 'abc'
l3 = 3*l
a = l3[0:-2]
```

15. What is the difference between a *data type* and a *data structure*? Give an example of each. [3]

16. Implement (in C) a function which takes as input an array of integers and the size of the array (as an integer), and prints the contents of the array in reverse order (i.e. last element first). Indicate, in words, how you would implement the same function if your input was instead a linked-list of integers (hint: recursion). [4]

17. BONUS question. Consider the following 3 implementations of Quicksort in different programming languages. The first 2 have a bug, while the 3rd is correct (though it may not necessarily help...). The bugs in the first 2 are “semantic” – in other words the code given is syntactically correct, it runs, but it gives the wrong answer. Each piece of buggy code only contains one (1) bug. Find and fix the bugs. [4]

```
qs := proc(l) local a,b;
  if l=[] then []
  else
    (a,b) := selectremove( x->(x <= l[1]), l[2..-1]);
    [ qs(a)[], qs(b)[] ];
  end if;
end proc;
```

```
qs [] = []
qs x:xs = qs b ++ [x] ++ qs a
  where (a,b) = partition (<= x) xs
```

```
qs([], []).
qs([X | W], Sorted) :-
  partition(X, W, Less, Greater),
  qs(Less, SortedLess), qs(Greater, SortedGreater),
  join(SortedLess, [X | SortedGreater], Sorted).
partition(_, [], [], []).
partition(X, [Y | V], [Y | Less], Greater) :-
  X > Y, partition(X, V, Less, Greater).
partition(X, [Y | V], Less, [Y | Greater]) :-
  partition(X, V, Less, Greater).
```