COMPUTER SCIENCE TRIPOS Part IB

Monday 1 June 2009 1.30 to 4.30

COMPUTER SCIENCE Paper 3

*Answer **five** questions.*

*Submit the answers in five **separate** bundles, each with its own cover sheet. On each cover sheet, write the numbers of **all** attempted questions, and circle the number of the question attached.*

You may not start to read the questions printed on the subsequent pages of this question paper until instructed that you may do so by the Invigilator

STATIONERY REQUIREMENTS
Script paper
Blue cover sheets
Tags

SPECIAL REQUIREMENTS
Approved calculator permitted

1 Programming in C and C++

Explain *five* of the following C or C++ features. You may use a short fragment of code to complement your explanation.

(*a*) The declaration of a C++ class illustrating constructor, variable, and method

(*b*) The use of a virtual destructor

(*c*) The difference between `malloc()` & `free()` and `new` & `delete`

(*d*) Overloading an operator

(*e*) Pointer arithmetic

(*f*) Catching and throwing exceptions including the passing of a user-defined structure

(*g*) The meaning of the keywords `static` and `const`

[4 marks each]

2 Concepts in Programming Languages

(*a*) List *three* differences between the programming languages FORTRAN and LISP. [3 marks]

(*b*) Give an example of a weakness in the type system of a programming language of your choice, explaining how it leads to a run-time error. [3 marks]

(*c*) In the context of parameter-passing mechanisms, give one reason why pass-by-value is preferable to pass-by-reference or *vice versa*, from the point of view of the programmer. [2 marks]

(*d*) Give an example of the following constructs

(*i*) signature [3 marks]

(*ii*) functor [3 marks]

in the SML Modules language.

(*e*) (*i*) In the programming language Scala, a generic class like the following one

```
abstract class Stack[A] {
  def push( x:A ): Stack[A]
  def top: A
  def pop: Stack[A]
  }
```

is *non-variant* by default. Why? [3 marks]

(*ii*) Modify the above abstract class to make it *co-variant*. [3 marks]

3 Concurrent Systems and Applications

(*a*) Define the ACID properties of transactions. [4 marks]

(*b*) In the context of transactions, what does it mean to say that an execution schedule is *serialisable*? [1 mark]

(*c*) Transaction systems can enforce either *strict* or *non-strict* isolation.

(*i*) What is the difference between the two? [1 mark]

(*ii*) Why do many systems enforce only non-strict isolation? [1 mark]

(*iii*) When using non-strict isolation, executing transactions may experience *lost updates*, *dirty reads* or *unrepeatable reads*. For *each* of these, describe with the aid of an example what it means. [1 mark each]

(*d*) Compare and contrast *two-phase locking* (2PL) and *optimistic concurrency control* (OCC) as means of providing isolation. You should discuss the level of isolation achieved, the degree of concurrent execution enabled, the behaviour in case of aborts, and the likely performance in the presence of contention. [10 marks]

4 ECAD

Below is a Verilog module that is supposed to produce a debounced version of a physical button input (`keyin`) on the output (`keyout`). Any bouncing behaviour of the input will have settled down within 100ms. Input `rst` is the global reset signal and `clk` is the global clock running at 50MHz. The module is syntactically correct, but is functionally slightly erroneous.

```
module debounce(input keyin, input clk, input rst, output reg keyout);
  reg prevkeyin;
  reg [7:0] ctr;

  always @(posedge clk or posedge rst)
    if(rst) begin
      prevkeyin <= 0;
      ctr <= 0;
      keyout <= 0;
    end else begin
      prevkeyin <= keyin;
      if(keyin!=prevkeyin) ctr <= -1; // set ctr to maximum value
      if(ctr>0) ctr <= ctr-1;
      if(ctr==0) keyout <= keyin;
    end
endmodule
```

(*a*) What is the difference between *synchronising* and *debouncing* an input? [3 marks]

(*b*) What is the high-level circuit diagram corresponding to the debounce module? You may assume that multi data-bit input blocks like adders, multiplexers and comparators can be used, that blocks of D-flip-flops (with clear, preset and enable inputs) are provided as well as Boolean gates. [5 marks]

(*c*) What are the functional errors in the debounce module and how can they be corrected? [8 marks]

(*d*) How does functional testing differ from production testing? [2 marks]

(*e*) How would scannable flip-flops speed up production test of the debounce module? [2 marks]

5 Economics and Law

In what ways should the information goods and services sector be less or more severely affected by the credit crunch than the rest of the economy? [20 marks]

6 Floating-Point Computation

(*a*) Briefly describe the 32-bit IEEE floating-point format, explaining what values (or other mathematical objects) are represented by bit-patterns in this format (you need not give the values corresponding to denormalised numbers). [4 marks]

(*b*) What value, if any, does the following Java method return, assuming `x` and `old` are held as 32-bit IEEE values?

```
float c() { float old=0, x=1;
            while (old != x) { old = x; x = x+1; }
            return x; }
```

Explain your reasoning. [3 marks]

(*c*) Consider the function computed by the Java method

```
float f(float x) { return x+1; }
```

Discuss how the use of 32-bit IEEE floating-point arithmetic causes it to differ from the mathematical function $f(x) = x + 1$. [4 marks]

(*d*) Given a problem of the form "find x such that $f(x) = y$", explain informally what it means for it to be *ill-conditioned*. [2 marks]

(*e*) The Newton–Raphson iteration for $\sqrt{a}$ uses $x_{n+1} = (x_n + a/x_n)/2$. Let $x_n = \sqrt{a} + \epsilon_n$, where the error ϵ_n is assumed to be small.

(*i*) Calculate how the error declines from one iteration to the next. [3 marks]

(*ii*) Given $1 \leq a < 4$ and $x_0 = 1.5$, how many iterations are necessary to achieve approximate 32-bit IEEE accuracy, and 64-bit IEEE accuracy? [2 marks]

(*iii*) Summarise a possible implementation of square-root on the whole 32-bit IEEE input range rather than just on $[1, 4)$. [2 marks]

7 Prolog

(*a*) Write a short implementation for each of the following predicates.

(*i*) `unify/2` is true if and only if its two arguments can be unified. [1 mark]

(*ii*) `fail/0` is never true. [1 mark]

(*iii*) `numequal/2` is true if and only if its arguments, interpreted as numerical expressions (assume integer values and no use of division), are numerically equal. For example, `numequal(1+3,2+2)` is true. [1 mark]

(*iv*) `member/2` is true if and only if its first argument is within the list that is its second argument. [1 mark]

(*b*) Given the following code, list all solutions, in order, for the query `c(X,Y,Z)`.

```
a(1).
a(2).
b(a).
c(A,B,C) :- a(A),d(B,C).
c(A,B,C) :- b(A),d(B,C).
d(B,C) :- a(B),!,a(C).
d(B,_) :- b(B).
```

[4 marks]

(*c*) The recursive clause of a bubblesort predicate is reproduced below (assume that `append/3` is defined already). Define *two* different base clauses for this predicate, one of which should use a green cut. When the first argument is unified with a list containing only integers, both of your answers should produce no additional solutions on backtracking. Explain how your complete bubblesort predicate works, including the purpose of the red cut.

```
bubblesort(X,Y) :-
  append(A,[H1,H2|B],X), H1 > H2, !,
  append(A,[H2,H1|B],X1), bubblesort(X1,Y).
```

[6 marks]

(*d*) The power set of a set S is the set of all subsets of S. We will represent sets using lists (ignore list order and assume no duplicates). Write a predicate `ps(+S,-PS)` that unifies `PS` with the power set of `S`, and explain how it works. Include the code for all predicates that you use. [6 marks]

8 Software Engineering

After 9/11, some people suggested that airliners should be fitted with autopilot remote control systems that would enable them to be taken over by air traffic control in the event of hijacking. The idea was revived in 2005 after a Boeing 737 from Larnaca crashed near Athens; apparently a failure of cabin pressure rendered the pilots unconscious. In 2006, Boeing obtained a patent for an "uninterruptible autopilot system".

You have been hired by the Civil Aviation Authority to develop a requirements specification and airworthiness certification programme for such products. Describe how you would organise and run this project.

[20 marks]

END OF PAPER