COMPUTER SCIENCE TRIPOS Part IB 75%, Part II 50% – 2019 – Paper 7

1 Concepts in Programming Languages (am21)

(*a*) Algol-60 provided two parameter-passing mechanisms: call-by-value and call-by-name.

(*i*) Explain these mechanisms. [2 marks]

(*ii*) Justify or criticise the statement that "the former is expensive for arrays and the latter interacts badly with side effects". [2 marks]

(*iii*) What parameter-passing mechanism(s) do C and Java use, and how do such languages deal with an array being passed as a parameter? [2 marks]

(*b*) A side-effect-free call-by-value language has its ML-like syntax of expressions e extended to be able to model call-by-name and (LISP-like) call-by-text:

$$e ::= \ldots \mid \texttt{suspend}\ e \mid \texttt{force}\ e \quad \text{(call-by-name)}$$
$$e ::= \ldots \mid \texttt{quote}\ e \mid \texttt{eval}\ e \quad \text{(call-by-text)}$$

Both `suspend` e and `quote` e yield an unevaluated representation of e as a value for later evaluation by `force` and `eval` respectively. Sketch two programs (differing only in whether they use `suspend` and `force` or `quote` and `eval`) which give different results. [*Note:* Answers using side-effecting operators can only gain partial marks.]

[4 marks]

(*c*) A library defines a generic class `Foo<T>` in a Java-like language. A user's program declares a class `C` and subclasses it as class `D`, creating variables `fc` and `fd` of types `Foo<C>` and `Foo<D>` respectively.

(*i*) Construct a declaration of `Foo<T>` along with a program of the above form containing the assignment `fc=fd` which, if this statement were legal, would be the cause of a later run-time error when executed. [5 marks]

(*ii*) How might the language syntax be changed to optionally express that the above assignment is to be allowed, indicating any compensating restrictions required for the declaration of `Foo<T>` or `fc` to avoid run-time errors. [3 marks]

(*iii*) How do Java arrays of type `T[]` fit in with your answer to Part (*c*)(*i*)? [2 marks]