

OLLSCOIL NA hÉIREANN
THE NATIONAL UNIVERSITY OF IRELAND

NATIONAL UNIVERSITY OF IRELAND, GALWAY
WINTER EXAMINATIONS 2002

THIRD UNIVERSITY EXAMINATION IN INFORMATION TECHNOLOGY

**PROGRAMMING PARADIGMS
CT331**

Prof. P. Nixon
Prof. G. Lyons
Ms. J. Griffith

Time allowed: **Two** hours

Answer any **THREE** questions
All questions carry equal marks

- Q. 1.** (i) What is meant by a *programming paradigm*? Distinguish between the imperative and declarative programming paradigms. Use sample code from a language of each of the two paradigms to support your answer.
- (ii) With respect to program translation distinguish between:
- Error recovery and error repair.
 - Lexical analysis and syntactic analysis.
 - Ambiguous grammars and equivalent grammars.
- (iii) Explain what is meant by a Finite State Automaton (FSA). Illustrate, with the aid of a diagram, a FSA which recognises strings of the form $b^n a^m b^p$ where $n \geq 0$, $m > 0$ and $p > 0$.

To what category of Chomsky's hierarchy does an FSA belong?

- Q. 2. (i) Describe the main features of the event-driven paradigm. Describe the three main approaches to event processing that can be taken by event-driven languages.
- (ii) MS Visual Basic provides a large amount of flexibility with respect to the approaches which can be taken when declaring variables. Discuss the advantages and disadvantages of this flexibility, listing the different approaches which can be taken.
- (iii) Discuss the main features of the concurrent programming paradigm. Describe what is meant by the term *deadlock* in relation to concurrent implementations. Describe possible approaches to the problem of deadlock, discussing their merits (or otherwise). With the aid of a diagram, describe the shared memory and message passing execution models.

- Q. 3. (i) Describe the main data structure available in the functional programming language SCHEME, outlining a representation for the data structure. With the aid of examples, describe the operations of `car`, `cdr` and `cons` which can be performed on the data structure.
- (ii) With the aid of examples, describe the SCHEME primitives `list` and `append`. Write a function in SCHEME which when passed an element and a list removes all occurrences of the element from the list. Explain the approach taken.
- (iii) Describe a representation of binary trees in SCHEME. Write code in SCHEME to implement a depth first search using the binary tree representation you developed. Explain the approach taken.

- Q. 4.** (i) Discuss the desirability (or otherwise) of the following programming language features:
- orthogonality.
 - fast translation.
 - portability.
- (ii) What is meant by programming language abstraction? With respect to procedural abstraction, discuss the problems that can arise with *side-effects* and *aliasing*. Use examples to support your answer.
- (iii) With respect to data abstraction, what is meant by a *binding*? Describe the four distinct binding times that can exist. With the aid of an example distinguish between the binding times in the SCHEME special forms `let*` and `let`.
-
- Q. 5.** (i) Describe the main features of the logic programming paradigm. With the aid of examples, distinguish between *facts*, *relations*, *rules* and *queries* in PROLOG.
- (ii) Control in SCHEME and PROLOG is mainly achieved through recursion. Distinguish between tail and non-tail recursive functions. Write a tail and non-tail recursive version of a function in SCHEME which finds the sum of the first n numbers, e.g.,
(sum 4) returns 10
- (iii) Describe, with the aid of examples, the list data structure in PROLOG, outlining its representation and syntax. Write PROLOG code to reverse the items (top level only) in a list, writing both a tail recursive and non-tail recursive version of the code. Explain the steps taken.