NUI Galway
OÉ Gaillimh

Semester I Examinations 2013/2014

Exam Code(s)	3BCT1, 3IF1, 1EM1
Exam(s)	3rd B.Sc. Computer Science and Information Technology 3rd B.Sc. (Information Technology) Erasmus
Module Code(s)	CT331
Module(s)	Programming Paradigms
Paper No.	1
Repeat Paper	
External Examiner(s)	Prof. Liam Maguire
Internal Examiner(s)	Prof. Gerard Lyons Dr. Michael Madden Dr. Enda Howley* Dr. Colm O'Riordan*
Instructions:	Candidates should answer **Three** questions, with at least one from each section. Each section should be answered in a separate answer book. All questions carry equal marks.
Duration	3 hours
No. of Pages	4

Requirements:
MCQ
Handout
Statistical/ Log Tables
Cambridge Tables
Graph Paper
Log Graph Paper
Other Materials

Release to Library: Yes ☐ No ☐

SECTION A

Q.1.

a) Explain what is meant by a Finite State Automaton (FSA) by drawing an FSA to recognise strings that have no consecutive 1s with an alphabet of {1,0}. For example

(i) 01101 is not a valid string
(ii) 1001 is a valid string
(iii)1 is a valid string
(iv)0 is a valid string

Illustrate how your FSA works given the above sample strings.

(7)

b) Given the FSA developed in part (a) suggest you would implement it in SCHEME.

(4)

c) Explain what is meant by a push-down automaton (PDA) by drawing a PDA and any associated data structures to recognise strings of the form 1^n0^n, i.e., any sequence of 1's of length n, followed by a sequence of 0's of the same length n. Illustrate how the PDA works with suitable examples.

(7)

d) With respect to grammars, explain the terms: *terminals*, *non-terminals*, *production rules.* What is meant by an *ambiguous* grammar? Give an example of such a grammar and explain the problems that may arise in using such a grammar.

(7)

P.T.O.

Q.2.

a) Distinguish between the SCHEME primitives *car* and *cdr* by writing sequences of *car* and *cdr* to extract the item "passing" from the following lists:

(3)

i) (and one day he came passing by)
ii) ((and one day) he (came passing by))
iii) ((and one day) (he came ((passing)) by))

b) Write a recursive function in SCHEME which performs a linear search of a list of numbers when passed a list and an item. You can assume that the data in the list is in ascending sorted order. The function should return #f or #t.
The function should stop searching when the item is found or when some number greater than the item is found or when the end of the list is reached without finding the item. For example, if the function is called linsearch:

(linsearch '10 '(2 4 6 8)) returns #f
(linsearch '4 '(2 4 6 8)) returns #t

Explain the approach taken, highlighting the base cases and the reduction stage.

(7)

c) Assume that a list represents a binary search tree. Write SCHEME code that performs a binary search on a tree and returns true (#t) if the value is found and returns false (#f) otherwise. Highlight both the base and reduction stages.

(7)

d) Write both a non-tail recursive and a tail-recursive function in SCHEME which counts the number of occurrences of an item in a list, returning this number. You can assume valid input and that there are no sub-lists. For example, if the function is named *countoccurs*:

(countoccurs 'a '(a b b a)) returns *2*

Explain the approach taken highlighting the base case and reduction stages for each function.

(8)

P.T.O.

SECTION B

Q.3.

(a) Outline the main benefits of using Object Oriented Languages?

(3)

(b) With the aid of suitable examples (including code), outline the main features of the Object Oriented paradigm. Your answer should make specific reference to each of the following:

- Objects,
- Methods,
- Inheritance,
- Polymorphism,

(12)

(c) Outline the core concept behind delegates in C#, and with the aid of a suitable code implementation show how they can be beneficial?

(10)

Q.4.

(a) What is meant by the Event Driven Paradigm?

(3)

(b) Design patterns are a significant aspect of C# Development. Outline the main features of Design Patterns and make specific reference to the following:

- Behavioral Design Patterns,
- Creational Design Patterns,
- Structural Design Patters,

(10)

(c) With the aid of suitable code examples, outline the main steps in managing and storing data in a C# environment. Make specific reference to:

- Managing a Database resource
- Serializable Objects.

(12)